2D IMAGE ENCRYPTION AND DECRYPTION USING AES ALGORITHM IN SPARTAN - 3E FPGA

Chandrasekhar V[1], Ravi Babu T[2], PSN Bhaskar[3], Nihar Ranjan Panda[4]
M. Tech Student, Assistant professor, Assistant professor, Associate professor
Dept. of Electronics & Communication Engineering,
Sanketika Vidya Parishad Engineering College, Visakhapatnam, India

Abstract: *AES* is one of the standard algorithm and widely used to encrypt and decrypt the data. In this paper, image encryption and decryption algorithm is implemented by using AES 128-bit core. Here, image information is converted into a hexadecimal format using MATLAB code and this plain hexadecimal data are transmitted to the FPGA via UART for encryption. Thus, the same process is also used for Decryption. The entire AES 128-bit core is simulated and synthesized for Spartan-3E-1600E FPGA using Xilinx ISE 14.3 and implementing on FPGA. By this design we calculate Throughput and observe Memory Allocation, Timing Report, Power Distribution and Device Utilization. To achieve high security to data, less power consumption and gives low complexity architecture. It's applicable in Defense and Research Organizations for send secure data from third party and make it easy for other Software and Hardware applications.. Results prove area utilization is less when compared to other results.

Keywords: AES, Image Encryption and Decryption, Block Cipher, Cipher text, Decipher text.

I. INTRODUCTION

In day to day life, a huge number of sectors exchange the large amount of database in various fields such as banking sectors, financial sectors, and medical sectors and so on. So, in these sectors security is essential. Every sector needed to keep secure the database then data cannot be hacked by unknown people. Most of the sectors, database usually secured by using cryptography techniques. There are several cryptography techniques that are available such as DES, triple DES, Blowfish, two fish, RSA, and AES. Among all cryptography techniques, AES is one of the standard algorithms.
The primary target of this study paper is to show a review of AES calculation measured with regarded to throughput. AES is more dependable design and it can be effortlessly worked at low inertness with high throughput.

The primary target of this paper is to execute a picture encryption and decoding utilizing AES-128-piece centre on simple FPGA through UART. Here, not just this proposed approach would measure and contrasted with past works with regarded with region, power, and dormancy.

RELATED WORK

Jignesh et al [2] proposed a hybrid based 128-bit key AES-DES encryption algorithm. In this method, the input image is initially converting into 128-bit plain text. This converted 128-bit text is again divided into two sets of 64-bit plain text data. This 64-bit plain text considered as input to the DES algorithm. Two such encrypted 64 but texts are again merged as single 128-bit encrypted data, which is applied to the AES algorithm for further encryption. This type of hybrid model gives better non - linearity when compared to plain AES. This design has better diffusion by merging with DES algorithm.

Y. Ou et al [3] proposed a concept of the region based selective encryption scheme which is used to achieve secure access for medical images. It employs AES algorithm to encrypt a certain region of data in the code stream.

S.H Kamali et al [4] proposed a modification of the AES algorithm[MAES], which has high level security and this design can be used for better image encryption. The main modification of this design is done in Shifting rows and columns. Suppose, If the bit positions are even in the first row and columns then the first and fourth rows are unchanged, and each byte in the second and third rows is shifted to the right cyclically. On the other hand, if the first and the second rows are unchanged, each byte of the second and fourth rows is shifted to the left. However security is same when entropy reaches maximum value.Here in this design approach results are proved that it provides better encryption than that of the original AES algorithm.

Ju-Young Ho et al [5] proposed the concept of expansion of the AES –Rijndael, named as Selective Encryption Algorithm considering five criteria namely compression of the plain data, second is size of the block, third is selectable round, fourth is optimization of software implementation and fifth is the selective function of the whole routine. The compressed image as input data not only gets high security but also reduces more than 35% of average execution time than that of the original AES algorithm.

M.zeghid et.al [6] proposed modified AES based algorithm for image encryption by adding a key steam gen-orator(A5/1,W7) to AES by ensuring improvement in the encryption performance that can be mainly useful for images characterized by reduced entropy., costs on toll operation and in view of the vehicle classification, cash toll is received by the collector.

The authority, who additionally apportions change, may acknowledge and offer scrip, tickets, coupons, making a passage of the vehicle in the framework and issuing receipt to the supporter. Because of manual mediation, the processing time is highest, change probl em is there.

IMPLEMENTATION OF AES-128 BIT CORE

The objective of this paper is to encrypt and decrypt of an image data. An image data cannot pass directly to the AES, So, there is necessary to implement any one of communica-tion interface to transferring an image to AES Core. In this implementation UART is opted as communication interface for AES 128-bit core. It has several hardware blocks such as UART, AES encryption and decryption block.

A. AES Encryption

AES is supported for both encryption and decryption for any kind of data. There are several standard type of bit lengths and along with symmetric Keys are used in AES such as 128,192,256-bits [7]–[9]. Here, in this paper 128-bit standard AES core is used. AES has several steps such as adding plain text, adding symmetric Key, shifting rows and columns, mixed columns, substitution box contains sub-bytes and shift rows. All these blocks are used for encryption process. A brief overview of AES Encryption process as shown in figure-1.

Initially, two inputs are given to the AES such as 128-bit data and 128-bit symmetric key (cipher key). At first 128-bit data is divided in to 16 bytes, and each byte consist of 8-bit length as illustrated in the figure-1

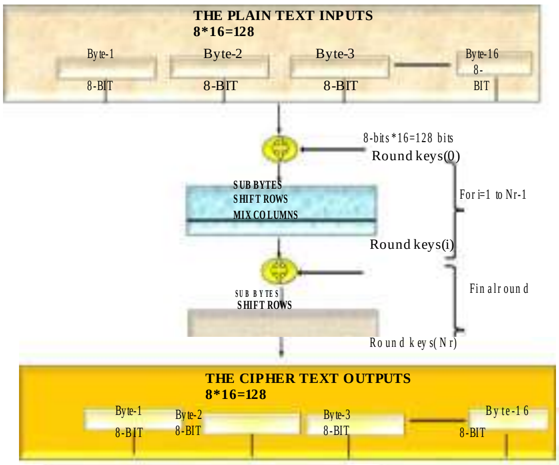


Figure 1: Encryption process

tion in that each byte is inverted independently and it can be performed by using look-up table. The total substitution box contains 4x4 matrix and all these substitution transformation is done in one clock cycle and it can be called as S-box transformation.

Second step shift row operation is performed in that first row is unchanged, second row rotate shift left by one byte, third row rotate shift left by two bytes, fourth row rotate shift left by three bytes. This process can be called as shift row transforma tion.

Third step is Mix-columns transformation is performed. In this mix-column transformation contains 4x4 matrix in that, states are considered as polynomial over Galois Field GF (28) and multiplied modulo X4 + 1 with fixed polyno-mial. This process is known as Mix-column transformation.

Fourth step is add round key, in this step, add round key is added to the output of Mix column transformation by making simpl e XOR operation. Each round has one distinctive key which is generated from maim key. So, tota lly 10 rounds are taken place with 11 distinctive keys which is generated from cipher key. Then this total number of 10 rounds are essential to encrypt the data.

similarly, the same steps are inversely happen in decryp-tion process. The decryption steps are like inv-sub bytes, inv-shift rows, inv-Mixcolumns, add round key. The decryption process is as illustrated in fig-2.

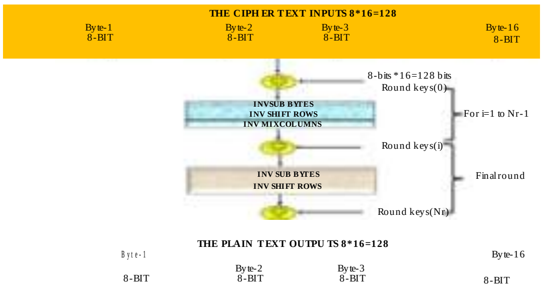


Figure 2: Encryption process

FPGA IMPLEMENTATION PROCESS OF AES-128 BIT CORE INTERFACE WITH UART

FPGA implementation of image encryption and decryption process has many blocks such as UART receiver & transmitter, Buffer/RAM, clock generator, counter, FSM control, 16x1 Mux, AES-128-bit core as shown in figure-3. Here, an image is considered as an input with the size of 4096 bytes. An image is converted into hexadecimal values and each byte is transferred through UART. UART receiver [10] can receive only 8-bits at a time. So, that the inputs are driven into the dual port memory which can be used for storing and retrieving purpose. The dual port memory size is 128 bits and each 128 bits is considered as one block of data. So, the target size is 4096 bytes then totally 256 blocks of data needed to receive. The number of blocks of data is measured by the counter. After receiving each block of data, it should be transferred to the AES 128-bit core. Here, the UART, Dual port memory, counter and AES-128 bit core are controlled by the finite state machine design. AES input key is fixed and another data input is getting from dual port buffer. The output size of AES is 128 bits and each 128-bit is divided in to 16 bytes. Each output byte is transferred to UART transmitter by 16x1 MUX. Here, the baud rate at the receiver and transmitter is set to 9600 bits per second. So, there is some essential calculations are required to transfer each bit.
According to the given baud rate is 9600, Given baud rate = 9600bits=second So,
each bit = 1=9600sec = 104usec,
Assume applied FPGA clock frequency is 50 Mhz and this clock frequency is converted into 20ns. Therefore, to transfer each bit is equal to given baud rate multiplied by applied clock frequency.

eachbit = 104usec 50M hz = 104000ns=20ns = 5200cyles

In this regard we cannot monitor those many clock cycles, so to increase the sample rate by 16 i.e,

1 bit = 5200=16 = 325cycles

So, by this we can determine the glitch problems by monitoring for every 8 clock cycles.

Therefore, FPGA implementation of AES decryption is as followed the same step implemented in AES encryption. Here, only difference is encrypted image is taken as an input and at the final output is decrypted image.

data considered as an input for AES. Initially, the input data is read by memory after storing into the memory then byte by byte has to be transferred for UART transmitter. For each byte of data delayed with 100ns. At the receiver side, UART receiver is receiving byte by byte and it is stored in file called crypto image.txt. The total amount of time taken for image encryption process is about 5358995ns. In the same way, decryption process is also done, but the only difference is cypto image.txt considered as an input and at the final output is decipher image.txt.

Assume applied FPGA clock frequency is 50 MHz and this clock frequency is converted into 20ns. Therefore, to transfer each bit is equal to given baud rate multiplied by applied clock frequency.

each bit = 104usec 50M Hz = 104000ns=20ns = 5200cyles

In this regard we cannot monitor those many clock cycles, so to increase the sample rate by 16 i.e.,
1bit = 5200=16 = 325cycles
So, by this we can determine the glitch problems by monitoring for every 8 clock cycles.
Therefore, FPGA implementation of AES decryption is as followed the same step implemented in AES encryption. Here, only difference is encrypted image is taken as an input and at the final output is decrypted image.

SIMULATION RESULTS WITH PERFORMANCE ANALYSIS

AES encryption and decoding reenactment is finished by utilizing Xilinx ISIM. Reproduction and test seat process as appeared in figure-4. The information picture is changed over in to Hexadecimal code utilizing MATLAB. The extent of the picture is 4096 bytes changed over in to 64x64 bytes. This hexadecimal
data considered as an input for AES. Initially, the input data is read by memory after storing into the memory then byte by byte has to be transferred for UART transmitter. For each byte of data delayed with 100ns. At the receiver side, UART receiver is receiving byte by byte and it is stored in file called crypto image.txt. The total amount of time taken for image encryption process is about 5358995ns. In the same way, decryption process is also done, but the only difference is crypto image.txt considered as an input and at the final output is decipher image.txt.

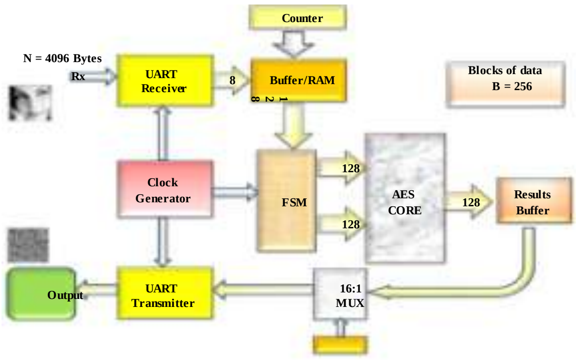


Figure 3: FPGA implementation process of AES core interface with UART

data considered as an input for AES. Initially, the input data is read by memory after storing into the memory then byte by byte has to be transferred for UART transmitter. For each byte of data delayed with 100ns. At the receiver side, UART receiver is receiving byte by byte and it is stored in file called crypto image.txt. The total amount of time taken for image encryption process is about 5358995ns. In the same way, decryption process is also done, but the only difference is cypto image.txt considered as an input and at the final output is decipher image.txt.

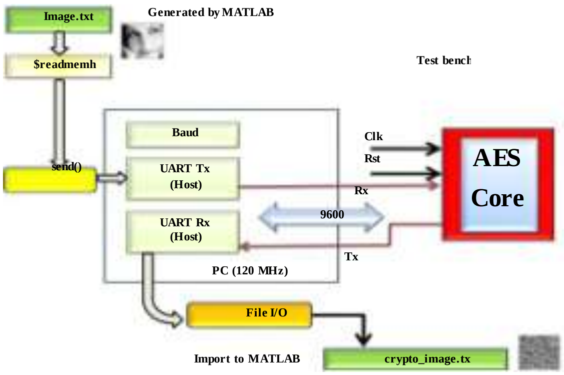


Figure 4: Simulation and Test bench process of AES

The performance analysis of AES encryption and decryption is described here. According to the synthesis report, the amount of latency is taken for 128-bits to encrypt and decrypt.

Therefore, the latency difference between encryption and decryption is 1.125 ns. Due to this difference, there is some noise presented in the decrypted image. In the decryption process, there is some packets are missing this is because of UART communication.

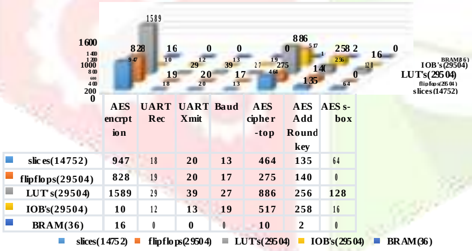
A performance analysis

The execution examination of AES encryption and decryp-tion is depicted here. As indicated by the combination report, the measure of idleness is taken for 128-bits to scramble around 6.645 ns and comparatively, for decoding is around 7.770 ns.

Therefore,the idleness contrast amongst encryption and decoding is 1.125 ns. Because of this distinction there is some clamor introduced in the unscrambled picture. In the decoding procedure there is a few parcels are feeling the loss of this is a direct result of UART correspondence.

The measure of region and power utilization is additionally ex-ceeded around 10% BRAMS and others are expanded about

AES Encryption Synthesis Report

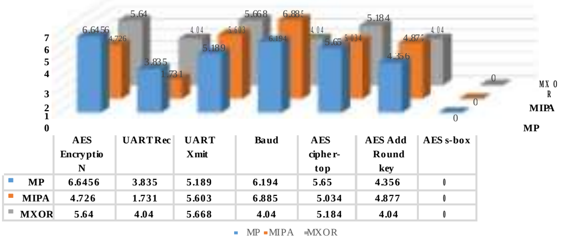


	AES encrpt ion	UART Rec	UART Xmit	Baud	AES cipher -top	AES Add Round key	AES s-box
slices(14752)	947	18	20	13	464	135	64
flipflops(29504)	828	19	20	17	275	140	0
LUT's(29504)	1589	29	39	27	886	256	128
IOB's(29504)	10	12	13	19	517	258	16
BRAM(36)	16	0	0	0	10	2	0

Figure 5: Synthesis Report of AES Encryption

In this synthesis report, total amount of area occupied for encryption is 6% of slices,2% of slice Flip flops, 5% of 4-input LUTs and 44% of BRAMs. A detailed report with graph representation as shown in fig-5.

AES Encryption Timing Summary

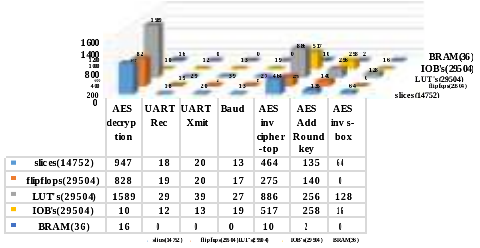


	AES EncryptioN	UART Rec	UART Xmit	Baud	AES cipher-top	AES Add Round key	AES s-box
MP	6.6456	3.835	5.189	6.194	5.65	4.356	0
MIPA	4.726	1.731	5.603	6.885	5.034	4.877	0
MXOR	5.64	4.04	5.668	4.04	5.184	4.04	0

Figure 6: Timing Report of AES Encryption MP:Minimum time period(ns) MIPA: Minimum input arrival time(ns) MXOR: Maximum output required time(ns)

In fig-6, overall timing summary for encryption is 6.6456ns, minimum input arrival time 4.726ns and maximum output required time 5.64ns.Here the encryption add round key alone minimum amount of time period is 4.356ns and along with some of the individual timing summary as represented in terms of table with graph.

AES Decryption Synthesis Report

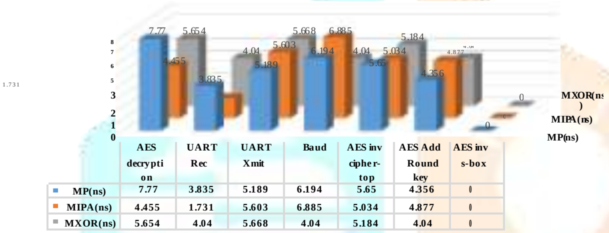


	AES decryption	UART Rec	UART Xmit	Baud	AES inv cipher-top	AES Add Round key	AES inv s-box
slices(14752)	947	18	20	13	464	135	64
flipflops(29504)	828	19	20	17	275	140	0
LUT's(29504)	1589	29	39	27	886	256	128
IOB's(29504)	10	12	13	19	517	258	16
BRAM(36)	16	0	0	0	10	2	0

Figure 7: Synthesis Report of AES Decryption

In this synthesis report, total amount of area occupied for decryption is 7% of slices,2% of slice Flip flops, 7% of 4-input LUTs and 55% of BRAMs. A detailed report with graph representation as shown in fig-7.

AES Decryption Timing Summary



	AES decryption	UART Rec	UART Xmit	Baud	AES inv cipher-top	AES Add Round key	AES inv s-box
MP(ns)	7.77	3.835	5.189	6.194	5.65	4.356	0
MIPA(ns)	4.455	1.731	5.603	6.885	5.034	4.877	0
MXOR(ns)	5.654	4.04	5.668	4.04	5.184	4.04	0

Figure 8: Timing Report of AES Decryption MP:Minimum time period(ns) MIPA: Minimum input arrival time(ns) MXOR: Maximum output required time(ns)

In fig-8, overall timing summary for encryption is 7.77ns, minimum input arrival time 4.455ns and maximum output required time is 5.654ns. Here the decryption add round key alone minimum amount of time period is 4.356ns and along with some of the individual timing summary as represented in terms of table with graph.

CONCLUSION

In this paper, picture encryption and unscrambling calculation actualized by utilizing AES 128-piece center. Here, the trial comes about are measured and contrasted with deference with region, power, and inertness. The measure of range and power utilization is surpassed under 1% in decoding. Here, not just that there is some clamor exhibited in unscrambling in light of the inertness variety. This issue is a result of UART correspondence. Because of this dormancy variety amongst encryption and decoding, UART correspondence isn't very much bolstered. Subsequently, rather than utilizing UART, UART is more speed and it can keep away from the loss of packets.

In this paper, picture encryption and unscrambling calculation actualized by utilizing AES 128-piece center. Here, the trial comes about are measured and contrasted with deference with region, power, and inertness. The measure of range and power utilization is surpassed under 1% in decoding. Here, not just that there is some clamor exhibited in unscrambling in light of the inertness variety. This issue is a result of UART correspondence. Because of this dormancy variety amongst encryption and decoding, UART correspondence isn't very much bolstered. Subsequently, rather than utilizing UART, UART is more speed and it can keep away from the loss of packets.

Result

The contributions of 4096 bytes information (ASCII Data) will be sent to Spartan 3E 1600E-FS320-5. In any case, it takes serial change of single substance (Byte) at once and after each of the 16-byte changed to FPGA then AES Encryption is improved the situation 128 bits and again same procedures for next 128 bits obstruct to 4096 bytes and the entire information/content is actualized on Spartan 3E 1600Efs320-5. In this plan the figure enter is modified in Verilog and added to this outline. In this way, its need not to give the figure key. The TMFT Terminal is utilized for changing information of 4096 bytes into FPGA by utilizing UART, which interface between Computer

also, Spartan 3E 1600E FPGA Kit. The information and yield can be seen in TMFT Terminal programming.

For decoding we utilize figure message as info and utilize a similar figure key for unscrambling calculation. The decoding is done on pc. At the point when decoding is done, we get unscrambling yield that is unique information. Unique information is gotten after ten rounds R10 of AES as appeared

ACKNOWLEDGMENT

The creators might want to thank the Principal and Management of Sankethika vidya parishad engineering college, Visakhapatnam for their kind help to complete the exploration work. Authors would like to thank all organizations for their sincere encouragement, support and assistance in the development. The authors would like to take this opportunity to thank them for their contributions towards refining improving the system towards a better product.

REFFERENCES

[1] Daemen J., and Rijmen V, "The Design of Rijndael: AES-the Advanced Encryption Standard", SpringerVerlag, 2002

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

[3] Ahmad, N.; Hasan, R.; Jubadi, W.M; “Design of AES S-Box using combinational logic optimization”, IEEE Symposium on Industrial Electronics & Applications (ISIEA), pp. 696-699, 2010.

[4] Alex Panato, Marcelo Barcelos, Ricardo Reis, “An IP of an Advanced Encryption Standard for Altera Devices”, SBCCI 2002, pp. 197-202, Porto Alegre, Brazil, 9 and 14 September 2002.

[5] Mr. Atul M. Borkar, Dr. R. V. Kshirsagar and Mrs. M. V. Vyawahare, “FPGA Implementation of AES Algorithm”, International Conference on Electronics Computer Technology (ICECT), pp. 401-405, 2011

[6] M.Zeghid , M.Machhout,L.Khriji,A.Baganne and R.Tourki,”A modified AES based algorithm for image encryption”,International journal of computer electrical, Automation, Control and information engineering2007.

[7] Mr. Atul M.Borkar, Dr. R. V. Kshirsagar, Mrs. M. V.Vyawahare” FPGA Implementation of AES Algorithm”, IEEE-2011, Volume : 3,pp 401-405.

[8] Monica Lib eratori, Fernando Otero, J. C. Bonadero, Jorge Castifieira”AES-128 cipher. high speed, low cost fpga implementation”, IEEE-2007.

[9] Chi-Wu Huang, Chi-Jeng Chang, Mao-Yuan Lin, Hung-Yun Tai, “Compact FPGA Implementationof 32-bits AES Algorithm Using Block RAM”, IEEE-2007.

[10] Hazim Kamal Ansari, Asad Suhail Farooqi,”Design Of High Speed Uart For Programming Fpga”,International Journal Of Engineering And Computer ScienceVolume1 Issue 1 Oct 2012 Page No. 28-36.