International Journal of Engineering Technology and Management Sciences
Website: ijetms.in Issue: 4 Volume No.6 July – 2022
DOI:10.46647/ijetms.2022.v06i04.0050 ISSN: 2581-4621

A Low Power Nano AES Security Algorithm For Image Cryptography

Tabassum Siddiqua T[1], Antony Joseph Arputha Raj A[2]

[1]*PG -VLSI Design, M.I.E.T Engineering College, Trichy, Tamilnadu*

[2]*Assistant Professor, Electronics &Communication Engineering, M.I.E.T Engineering College, Trichy, Tamilnadu*

ABSTRACT

Advanced Encryption Standard (AES) is a specification for electronic data encryption. This standard has become one of the most widely used encryption method and has been implemented in both software and hardware. A high-secure symmetric cryptography algorithm, implementation on field-programmable gate array (FPGA). The proposed architecture includes 8-bit data path and five main blocks. We design two specified register banks, Key-Register and State-Register, for storing the plain text, keys, and intermediate data. This project is simulated by Modelsim 6.4 c and synthesized by Xilinx tool.

Keywords— Advanced Encryption Standard,Cryptography

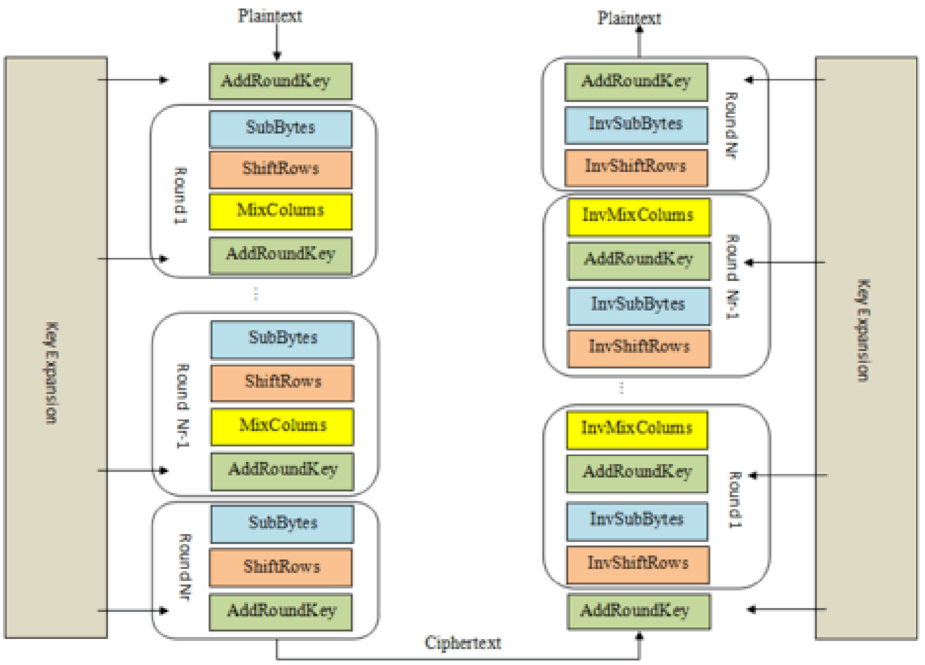
1. Introduction

An Advanced encryption standard (AES) is one of the secure symmetric cryptography algorithms that is widely used indifferent networks and is the main security part in various applications, platforms, and networks, such as IoT and in other Internet standards. According to the length of the key, AES provides different security levels. Based on the AES algorithm with a 256-bit key is secure enough in the quantum era, which means that this algorithm can provide the security requirement of different levels of IoT applications and protocols.

With AES both the encryption and the decryption are performed using the same key. This is called a symmetric encryption algorithm. Encryption algorithms that use two different keys, a public and a private key, are called asymmetric encryption algorithms. An encryption key is simply a binary string of data used in the encryption process. Because the same encryption key is used to encrypt and decrypt data, it is important to keep the encryption key a secret and to use keys that are hard to guess. Some keys are generated by software used for this specific task. Another method is to derive a key from a pass phrase. Good encryption systems never use a pass phrase alone as an encryption key. Some of the disadvantages of the software implementation of AES are the high latency for processing the data and transmission and consumption of more power.

2.AES Algorithm

The Pipelined AES Design for Image Encryption & Decryptionare shown using the figure given below. The Pipelined algorithm is a symmetric iterated block cipher. The block and key lengths can be 128, 192, or 256 bits. The algorithm begins with an Add round key stage followed by 9 rounds



of four stages and a tenth round of three stages. This applies for both encryption and decryption with the exception that each stage of a round the decryption algorithm is the inverse of its counterpart in the encryption algorithm.

FIGURE 2.1 AES ALGORITHM

The four stages are as follows:

- Substitute bytes
- Shift rows
- Mix Columns
- Add Round Key

The tenth round simply leaves out the Mix Columns stage. The first nine rounds of the decryption algorithm consist of the following:

- Inverse Shift rows
- Inverse Substitute bytes
- Inverse Add Round Key
- Inverse Mix Columns

3. Methodology

The main contribution of our work is to design a lightweight AES architecture. To achieve this goal, we employ some of the best implementation techniques and design specified blocks according to our goals as follows.

1) In order to reduce the required logic, the Shift-Rows are embedded inside the State-Register.

2) We optimize Sub-Bytes block and share it with key expansion phase and encryption phase.

3) We design an optimized 8-bit block for Mix-Columns with 8-bit input and output that is based on the structure of 8-bit data path, which is followed by Add-Round-Key. In comparison to 32-bit Mix-columns, it is not necessary to store the results in the registers or increase the data path for Key-Register to 32-bit.

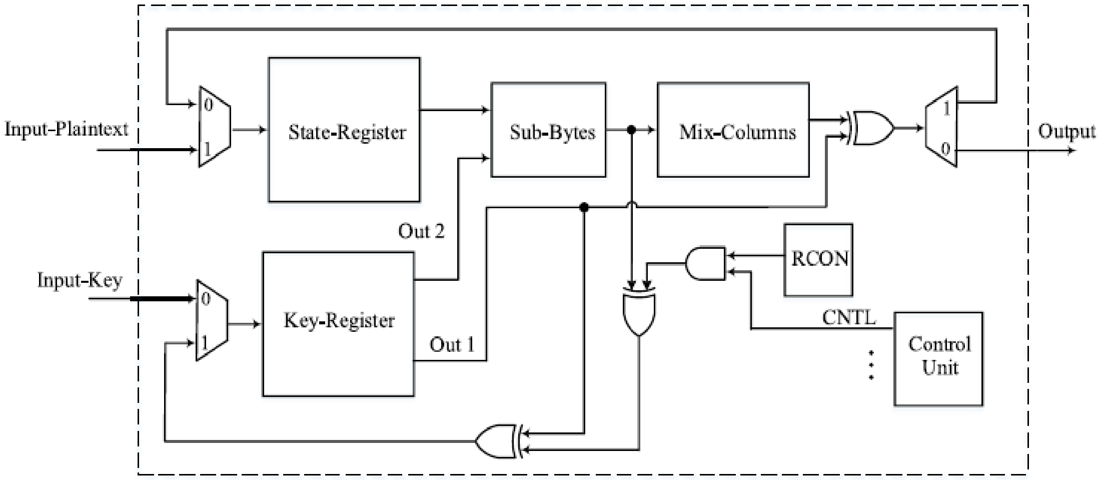


FIGURE 3.1 BLOCK DIAGRAM OF PROPOSED SYSTEM

4) To reduce the power consumption of the design, the clock gating technique is applied in different parts of the design, which leads to reduce the power consumption. In the different parts of the design, we apply the clock gating technique to reduce the dynamic power consumption. The clock gating is separately applied on State-Register, the internal registers of Mix-Columns, Key-Register & RCON. For instance, the most power consumption is saved during the key expansion phase; the clock of State-Register and Mix-Columns is disabled to save power because these two blocks are not used in the key expansion phase.

State Register

The State-Register consists of sixteen 8-bit registers, each register contains eight flip-flops. State-Register is based on a shift-register memory topology, in which, in each cycle, one 8-bit is fed to the design and one 8-bit is stored in the State- Register if needed. To reduce the area, we do not design a specific block for Shift-Rows. Since Shift-Rows and Sub-Bytes are executed on each byte of the input, these two functions can be exchanged without any effect on final results. We calculated the final results after the Shift-Rows function and applied it to the State-Register.One of the State-Register's duties is to execute Shift-Rows. Each register of State-Register contains one or two inputs that require a multiplexer to select from two inputs. One input receives the data from the previous register to execute the encryption phase and pass the data between internal registers The other one includes some interconnections between different units of the State- Register to execute Shift-Rows. Thus, Shift-Rows isdone by wiring, and this completely removes the logic for Shift-Rows step.

Key-Register

The architecture of the proposed Key-Register includes sixteen 8-bit registers, in which each register contains eight flip-flops with one or two input(s) and a 2-1 MUX, to store the keys. It also contains one 8-bit input and two 8-bit outputs. The reason that we design our Key-Register with two outputs is that to expand the keys, two columns of the previous keys are required at the same time. Key-Register includes five main operations: first, store the initial keys in the Key-Register; second, feed the design by one 8-bit for encryption phase. And store the same key again in the same cycle, thus after 16 clock cycles, the keys are stored in the same position before encryption phase.

Sub-Bytes Optimization

Sub-Bytes is one of the most critical parts of the AES design in terms of power, area, and latency. Our design contains one Sub-bytes, which is used for both the encryption and key expansion. There are different methods of implementations of this block. Although the most straight forward way of implementation id lookup table (LUT).

KeyExpansion

For each round of the algorithm, one 128-bit key is needed. The key in each round is generated from the previous round key. Storing all the keys in the design requires a huge memory, which is not a proper method for resource-constrained devices. Thus,On-the-fly key expansion is a wise way that only requires one 128-bit register. The key expansion phase contains the shift of the last column of Key-Register, Sub-Bytes, RCON, and XOR.

Control Unit

The control unit includes one 4-bit and 2-bit counters. During the data encryption, the Key-Register and State-Register put one element in data pathand store one element according to the shift-register topology.

Clock-Gating Technique

Clock gating is a popular technique used in many synchronous circuits for reducing dynamic power dissipation. Clock gating saves power by adding more logic to a circuit to prune the clock tree. The clock -gating is one of the effective logics in RTL and architectural power reduction. Clock gating is an effective technique to reduce dynamic power, because individual IP usage varies across applications, not all IP cores are used all the time, giving rise to opportunity for reducing the unused IP cores power.Bycombining (AND gate) the clock with a gate-control signal, clock gating essentially disables the clock to an IP core. When that IP is not used, avoiding power dissipation due to unnecessary charging and discharging of the unused circuits. The continuous decrease in the minimum feature size of transistors which increase of both device density and design complexity.

4.Proposed Application

An Image Cryptography (Encryption & Decryption Based on Images). We are going to merge the Matlab & VLSI for this process. We are using Matlab to convert an image into plaintext and then it is encrypted using a key. Then, decrypted using the same key. For the conversion of an image into text Matlab is used. And Modelsim is used for the VLSI process.

The Following process is given in steps:

S-1: A colour image of 256 x 256 uploaded.
S-2: Then, the colour image is converted into grey image.
S-3: Now, the input image is encrypted using the verilog coding.
S-4: Again, the input image is seen using decryption verilog coding.Thus, the image cryptography is enumerated.

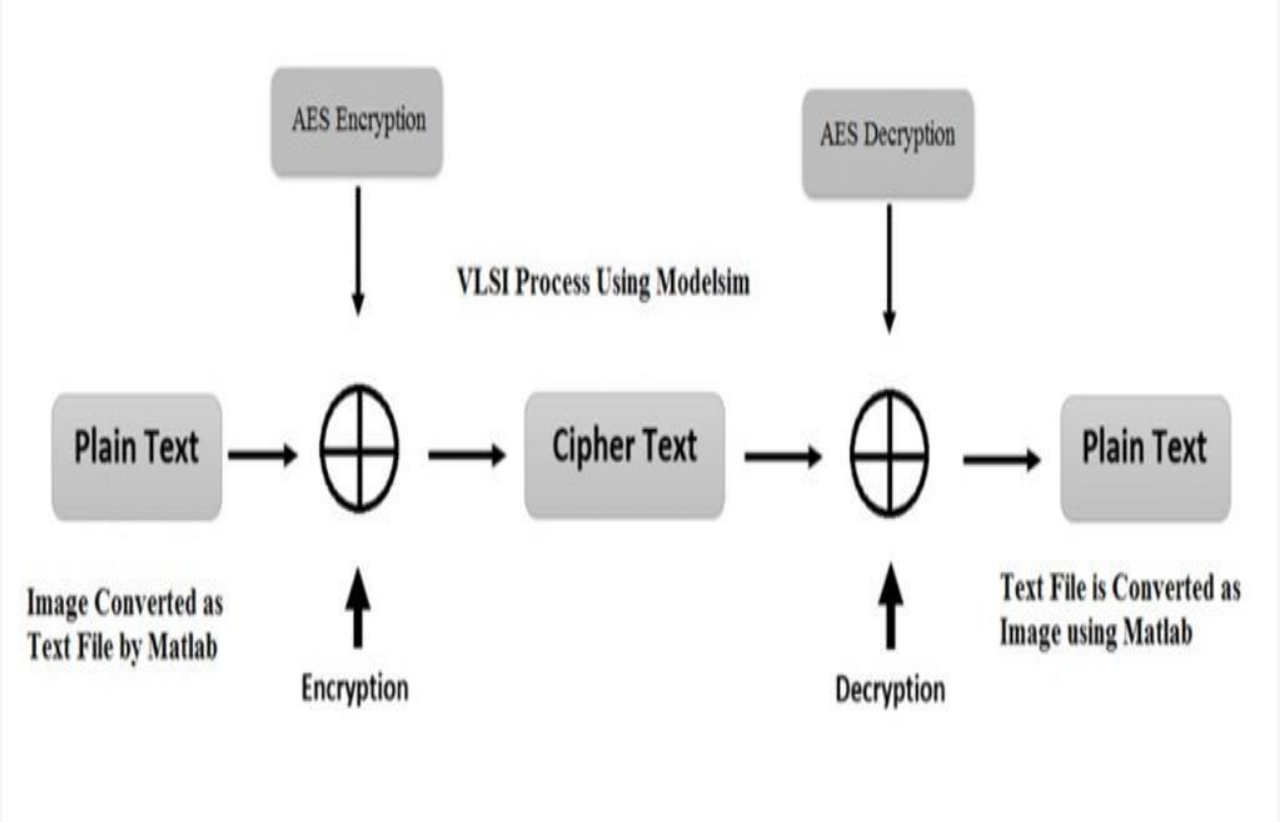
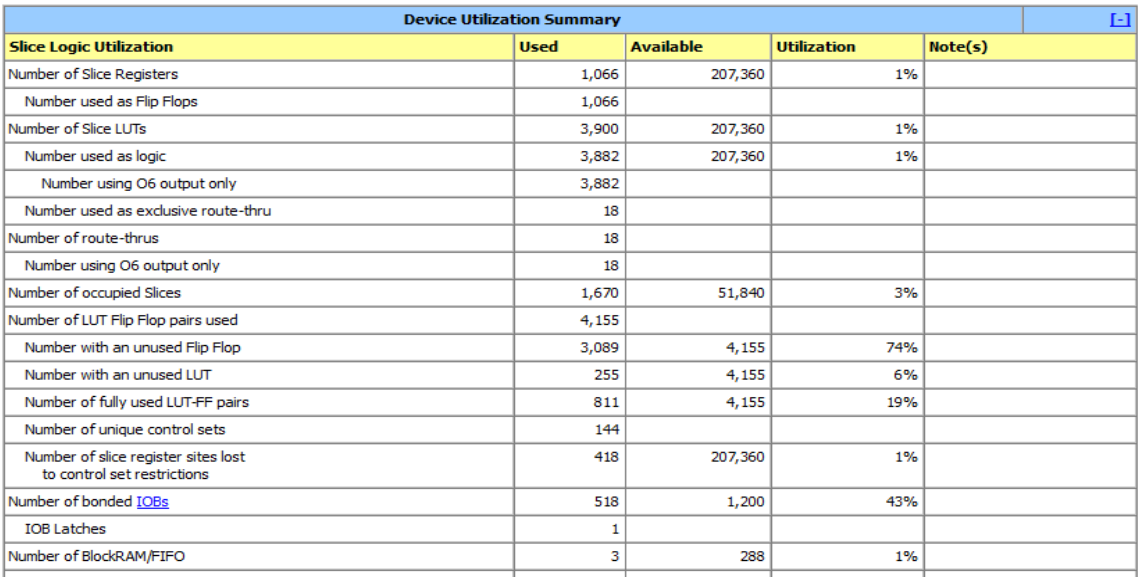

AES Encryption
AES Decryption
VLSI Process Using Modelsim
Plain Text
Cipher Text
Plain Text
Image Converted as Text File by Matlab
Encryption
Decryption
Text File is Converted as Image using Matlab

FIGURE 4.1 PROPOSED APPLICATION DIAGRAM

5. Results and Discussion

Synthesization Using Xilinx 9.1 Tool

In Comparison to the existing system and technologies, the verilog code for an encryption and decryption using nano AES(Advanced Encryption Standard) clock gating technique is computed which gives the output of device utilisation summary where the usage of Slice Registers, Flip-Flops, Look-up Table are reduced.



Device Utilization Summary				[-]
Slice Logic Utilization	**Used**	**Available**	**Utilization**	**Note(s)**
Number of Slice Registers	1,066	207,360	1%	
Number used as Flip Flops	1,066			
Number of Slice LUTs	3,900	207,360	1%	
Number used as logic	3,882	207,360	1%	
Number using O6 output only	3,882			
Number used as exclusive route-thru	18			
Number of route-thrus	18			
Number using O6 output only	18			
Number of occupied Slices	1,670	51,840	3%	
Number of LUT Flip Flop pairs used	4,155			
Number with an unused Flip Flop	3,089	4,155	74%	
Number with an unused LUT	255	4,155	6%	
Number of fully used LUT-FF pairs	811	4,155	19%	
Number of unique control sets	144			
Number of slice register sites lost to control set restrictions	418	207,360	1%	
Number of bonded IOBs	518	1,200	43%	
IOB Latches	1			
Number of BlockRAM/FIFO	3	288	1%	

FIGURE 5.1 SUMMARY REPORT

Schematic View

It shows the implementation logic of the circuit that how data flows in and out from the circuit. RTL is used in the logicdesign phase of the integrated circuit design cycle. An RTL schematic description is usually converted to a gate-level description of the circuit by a logic synthesis tool. The synthesis results are then used by placement and routing tools to create a physical layout. Logic simulation tools may use a design's RTL description to verify its correctness. The Schematic View of Encryption & Decryption Module with Clock Gating using Xilinx tool is shown below.

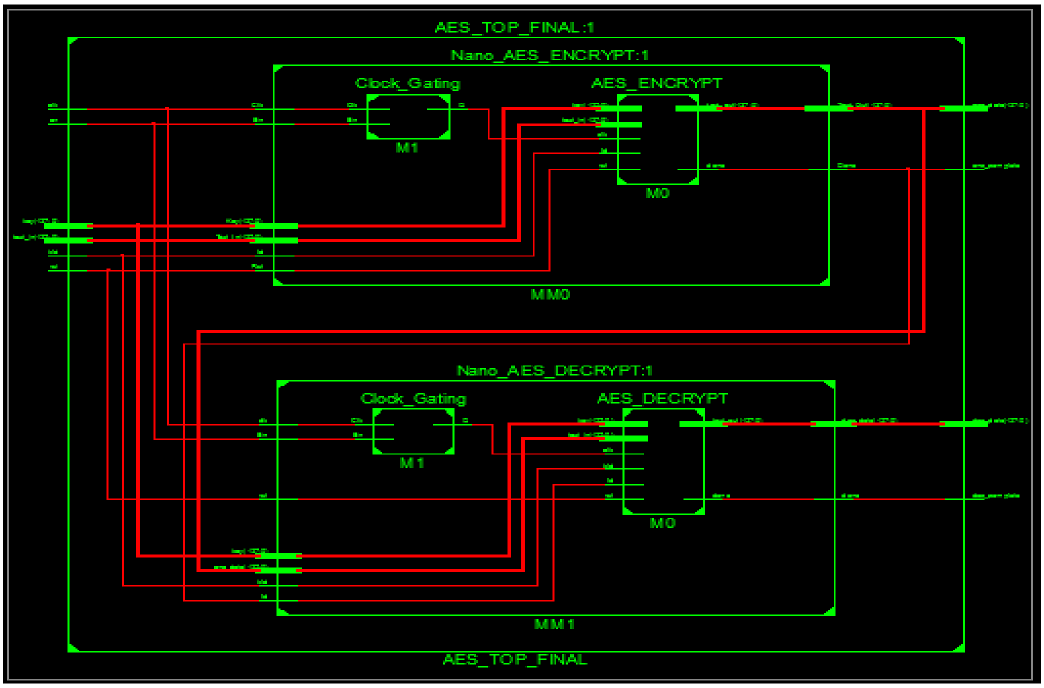
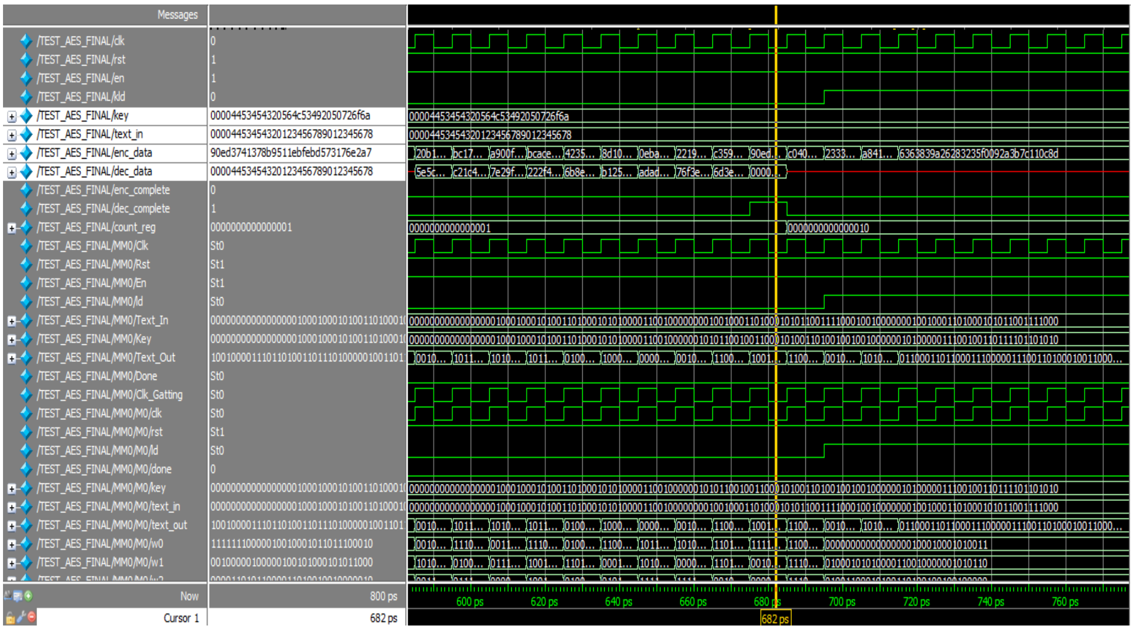


FIGURE 5.2 ENCRYPTION & DECRYPTION MODULE WITH CLOCK GATING

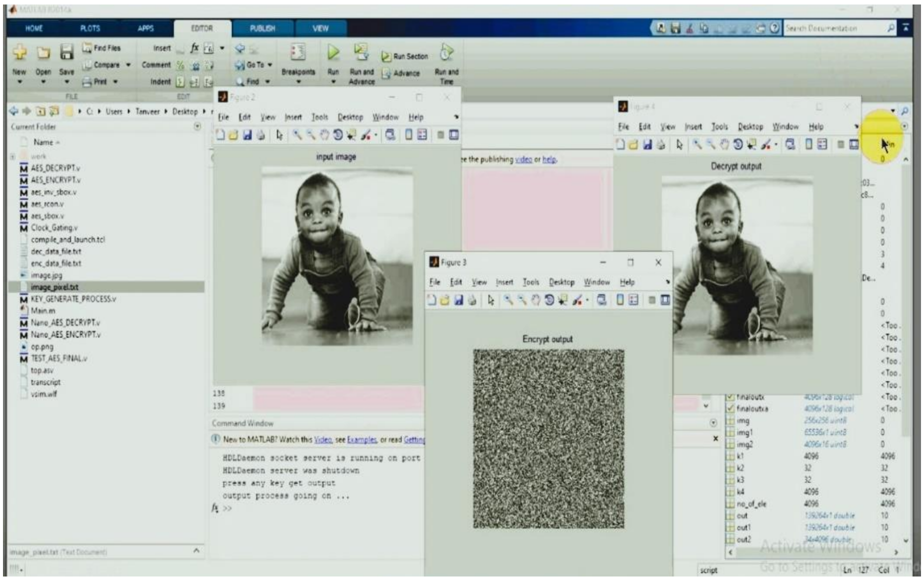
Simulation Using ModelSim 6.4a

The code is simulated to display forms. To display waveforms, select the design under test in the "sim" tab, right-click the mouse, and select "Add > To Wave >All items in region". Alternately, bring up a wave window by selecting "View > Wave" in the ModelSim menu. The waveform shows occurrence of encryption and decryption of the system.



```
Messages
/TEST_AES_FINAL/clk                 0
/TEST_AES_FINAL/rst                 1
/TEST_AES_FINAL/en                  1
/TEST_AES_FINAL/kld                 0
/TEST_AES_FINAL/key                 00004453454320564c53492050726f6a
/TEST_AES_FINAL/text_in             00004453454320123456789012345678
/TEST_AES_FINAL/enc_data            90ed3741378b9511ebfebd573176e2a7
/TEST_AES_FINAL/dec_data            00004453454320123456789012345678
/TEST_AES_FINAL/enc_complete        0
/TEST_AES_FINAL/dec_complete        1
/TEST_AES_FINAL/count_reg           0000000000000001
/TEST_AES_FINAL/MM0/Clk             St0
/TEST_AES_FINAL/MM0/Rst             St1
/TEST_AES_FINAL/MM0/En              St1
/TEST_AES_FINAL/MM0/ld              St0
/TEST_AES_FINAL/MM0/Text_In         00000000000000000100010001010011010001
/TEST_AES_FINAL/MM0/Key             00000000000000000100010001010011010001
/TEST_AES_FINAL/MM0/Text_Out        100100001110110100110111010000010011011
/TEST_AES_FINAL/MM0/Done            St0
/TEST_AES_FINAL/MM0/Clk_Gatting     St0
/TEST_AES_FINAL/MM0/M0/clk          St0
/TEST_AES_FINAL/MM0/M0/rst          St1
/TEST_AES_FINAL/MM0/M0/ld           St0
/TEST_AES_FINAL/MM0/M0/done         0
/TEST_AES_FINAL/MM0/M0/key          00000000000000000100010001010011010001
/TEST_AES_FINAL/MM0/M0/text_in      00000000000000000100010001010011010001
/TEST_AES_FINAL/MM0/M0/text_out     100100001110110100110111010000010011011
/TEST_AES_FINAL/MM0/M0/w0           11111110000010010001011011100010
/TEST_AES_FINAL/MM0/M0/w1           00100000100000100101000101011000
Now                                 800 ps
Cursor 1                            682 ps
```

FIGURE 5.3 NANO AES ENCRYPTION & DECRYPTION



```
New to MATLAB? Watch this Video, see Examples, or read Getting
HDLDaemon socket server is running on port
HDLDaemon server was shutdown
press any key get output
output process going on ...
>>
```

FIGURE 5.4 APPLICATION OUTPUT

Image Cryptography using MatLab and Modelsim (Application)

We are using Matlab to convert an image into plaintext and then it is encrypted using a key. Then, decrypted using the same key. For the conversion of an image into text MatLab is used. And Modelsim is used for the VLSI process.

The given image is converted into plain text. And then, encrypted using a key. It is again decrypted using same key. Thus, the image is decrypted.

Comparison Table

In this comparison table, Area and Delay of the Normal AES Design and Proposed Nano-AES with clock-gating is compared. As a result of this comparison, area and delay is reduced in our proposed system to a great extent.

S. No	Method Name	Area			Delay		
		Slice	Flip Flops	LUT	Max Delay	Gate Delay	Path Delay
1	Normal AES Design	7734	21207	21207	160.860ns	25.302ns	135.558ns
2	Proposed Nano AES with Clock Gating	1670	1066	3900	3.405ns	2.923ns	0.482ns

TABLE 5.1 COMPARISON BETWEEN NORMAL AND PROPOSED AES WITH CLOCK-GATING

CONCLUSION

Nano AES is a secure symmetric cryptography algorithm with a high level of security, which is widely used in many applications and networks. Thus, AES is a suitable algorithm for tiny IoT devices. In this project, we designed a lightweight AES architecture for resource-constrained IoT devices.The power consumption of the design was simulated in different timing constraints. To make a fair comparison with other similar works, we calculated the normalized power of previous works. The power consumption of the proposed design was better than most of the previous works. The result of the proposed lightweight AES design is suitable for resource-constrained devices and can be supplied by low-power devices. As it is said, AES is a high secure symmetric algorithm. The AES with key length 256-bit is secure in the quantum era. In future work, wewill focus on designing a postquantum resistance AES for resource constrained IoT devices. Increasing the number of rounds and keys has a high impact on area, power, and latency. To mitigate these issues, we will focus on an optimized architecture for the design to reduce the power.

REFERENCES

1. Satoh, T. Sugawara, N. Homma, and T. Aoki, “High-performance concurrent error detection scheme for AES hardware,” in Proc. 10th Int. Workshop CHES, Aug. 2008, pp. 100–112.

2. C. J. A. Jansen, T. Helleseth, and A. Kholosha, “Cascade jump controlled sequence generator and Pomaranch stream cipher (version 3),” Dept. Informat., Univ. Bergen, Bergen, Norway, Tech. Rep. 2006/006, 2006. [Online].

3. C. J. A. Jansen, T. Helleseth, and A. Kholosha, “Cascade jump controlled sequence generator (CJCSG),” in Proc. Workshop Symmetric Key Encryption, 2005, pp. 1–16.

4. D. Halperin, T. Kohno, T. S. Heydt-Benjamin, K. Fu, and W. H. Maisel, “Security and privacy for implantable medical devices,” IEEE Pervasive Comput., vol. 7, no. 1, pp. 30–39, Jan./Mar. 2008.

5. H. Khurana, M. Hadley, N. Lu, and D. A. Frincke, “Smart-grid security issues,” IEEE Security Privacy, vol. 8, no. 1, pp. 81–85, Jan./Feb. 2010.

6. K. Fu and J. Blum, “Controlling for cyber security risks of medical device software,” Commun. ACM, vol. 56, no. 10, pp. 35–37, Oct. 2013.

7. M. Mozaffari-Kermani and A. Reyhani-Masoleh, “A lightweight high performance fault detection scheme for the Advanced Encryption Standard using composite fields,” IEEE Trans. Very Large Scale Integr. (VLSI) Syst., vol. 19, no. 1, pp. 85–91, Jan. 2011.

8. M. Mozaffari-Kermani and A. Reyhani-Masoleh, “A low-power high performance concurrent fault detection approach for the composite field S-box and inverse S-box,” IEEE Trans. Comput., vol. 60, no. 9, pp. 1327–1340, Sep. 2011

9. M. Mozaffari-Kermani and A. Reyhani-Masoleh, “Concurrent structure independent fault detection schemes for the Advanced Encryption Standard,” IEEE Trans. Compute., vol. 59, no. 5, pp. 608–622, May 2010.

10. M. Mozaffari-Kermani and R. Azarderakhsh, “Efficient fault diagnosis schemes for reliable lightweight cryptographic ISO/IEC standard CLEFIA benchmarked on ASIC and FPGA,” IEEE Trans. Ind. Electron., vol. 60, no. 12, pp. 5925–5932, Dec. 2013.

11. M. Mozaffari-Kermani, M. Zhang, A. Raghunathan, and N. K. Jha, “Emerging frontiers in embedded security,” in Proc. 26th Int. Conf. VLSI Design, Jan. 2013, pp. 203–208.

12. M. Rostami, W. Burleson, A. Jules, and F. Koushanfar, “Balancing security and utility in medical devices?” in Proc. 50th ACM/EDAC/IEEE Int. Conf. Design Autom., May/Jun. 2013, pp. 1–6.

13. M. Zhang, A. Raghunathan, and N. K. Jha, “Trustworthiness of medical devices and body area networks,” Proc. IEEE, vol. 102, no. 8, pp. 1174–1188, Aug. 2014.

14. P. Maistri and R. Leveugle, “Double-data-rate computation as a countermeasure against fault analysis,” IEEE Trans. Comput., vol. 57, no. 11, pp. 1528–1539, Nov. 2008.

15. R. Roman, P. Najera, and J. Lopez, “Securing the Internet of things,” Computer, vol. 44,no. 9, pp. 51–58, Sep. 2011.