iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 10 **Issue:** IV **Month of publication:** April 2022

DOI: https://doi.org/10.22214/ijraset.2022.41376

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 10 Issue IV Apr 2022- Available at www.ijraset.com

Efficient PPA Nano AES for IoT Applications

Dr. Yogesh G S [1], Dr. Anitha R [2], Sagar S [3]
[1]*Professor and Head of the Dept of Electronics and Communication Engineering, East Point College of Engineering and Technology, Bengaluru, 560049*
[2]*Associate Professor, Dept of Electronics and Communication Engineering, East Point College of Engineering and Technology, Bengaluru, 560049*
[3]*Pg Scholar, Dept of Electronics and Communication Engineering, East Point College of Engineering and Technology, Bengaluru, 560049*

Abstract: ***Due to the fast-growing number of connected tiny devices to the Internet of Things (IoT), providing end-to-end security is vital. Therefore, it is essential to design the cryptosystem based on the requirement of resource constrained IoT devices. This article presents a lightweight advanced encryption standard (AES), a high-secure symmetric cryptography algorithm, implementation on field-programmable gate array (FPGA) and 65-nm technology for resource-constrained IoT devices. 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. To reduce the area, Shift-Rows is embedded inside the State-Register. To adapt the Mix-Column to 8-bit data path, we design an optimized 8-bit block for Mix-Columns with four internal registers, which accept 8-bit and send back 8-bit. Also, a shared optimized Sub-Bytes is employed for the key expansion phase and encryption phase. To optimize Sub-Bytes, we merge and simplify some parts of the Sub-Bytes. To reduce power consumption, we apply the clock gating technique to the design. Application-specific integrated circuit (ASIC) implementation results show a respective improvement in the area over the previous similar works from 35% to 2.4%. Based on the results, the proposed design is a suitable cryptosystem for tiny IoT devices.***
Keywords: IOT, Clock gating technique, ASIC

Abbreviations
1) ***IOT – Internet of Things***
2) ***AES – Advanced Encryption Standard***
3) ***FPGA – Field Programmable Gate Array***
4) ***ASIC – Application- Specific Integrated Circuit***

I. INTRODUCTION

The network-centric IoT architecture under consideration in this project is aimed at providing a high-performance platform for experimentation with various adaptive wireless network protocols ranging from simple etiquettes to more complex ad-hoc collaboration. Particular emphasis has been placed on high performance in a networked environment where each node may be required to carry out high throughput packet forwarding functions between multiple physical layers.

A. Objectives

Key design objectives for the IoT platform include:
1) Multi-band operation, fast frequency scanning and agility;
2) Software-defined modem including waveforms such as DSSS/QPSK and OFDM operating at speeds up to 50 Mbps;
3) Packet processor capable of ad-hoc packet routing with aggregate throughput ~100 Mbps;
4) Spectrum policy processor that implements etiquette protocols and algorithms for dynamic spectrum sharing.

B. AES Algorithm

The AES algorithm is a symmetric cipher. In symmetric ciphers, a single secret key is used for both the encryption and decryption, whereas in asymmetric ciphers, there are two sets of keys known as private and public keys. The plaintext is encrypted using the public key and can only be decrypted using the private key. In addition, the AES algorithm is a block cipher as it operates on fixed-length groups of bits (blocks), whereas in stream ciphers, the plaintext bits are encrypted one at a time, and the set of transformations applied to successive bits may vary during the encryption process. And the composite field S-box and Inverse S-box are divided into many blocks and fault detection scheme is introduced only in S-box and Inverse S-box and optimum solutions were found.

II. NANO AES FOR IOT

In this, a hardware model for implementing the AES128 algorithm is introduced. The model is implemented using the Verilog hardware description language. This chapter covers the design and implementation issues of the AES128 algorithm. This means that the model provides a cycle-by-cycle RTL description of the circuit that a logic synthesis tool can convert to an optimized gate-level netlist.

The modeling process utilized in this project is the bottom-up approach. This means that the leaf components in the design hierarchy were developed first and the higher-level modules were constructed by instantiating their subcomponents and connecting them with the internal signals. All the modules in the design hierarchy were modeled in behavioral style.

A. Design Hierarchy

The proposed AES128 hardware model is a 3-level hierarchical design as shown in Figure1. The root module in the hierarchy is the AES128_cipher_top. It has two 128- bit inputs for receiving the cipher key and the plaintext. There is also a single bit input signal, 'Ld', which is used to indicate the availability of a new set of plaintext or cipher key on the input ports. The completion of the encryption process is indicated by asserting the 'done' single bit output.

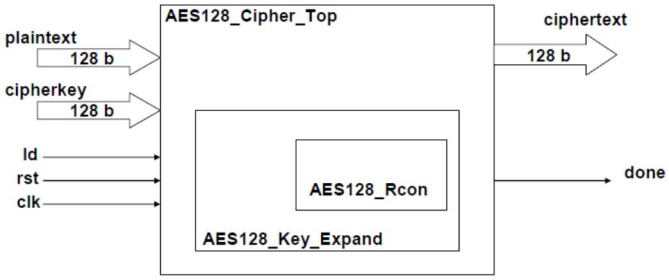


Fig 1: Design Hierarchy

B. AES128 Encryption Process

There are ten rounds of transformations represented by *r1* to *r10* states. The *r0* state corresponds to the initial *AddRoundKey* transformation in Figure 2.

After leaving the *Reset* state, the AES128_Cipher_Top module waits for assertion of the '*Ld*' signal, which indicates that a valid set of plaintext and cipher key is available on the input ports. After reaching the *r0* state, there is a transition on every clock cycle for the next ten cycles, as ten rounds of encryption is applied to the *State*.

After going through ten rounds of transformations, the '*done*' signal is asserted to indicate the completion of cipher and availability of the ciphertext on the corresponding output port.

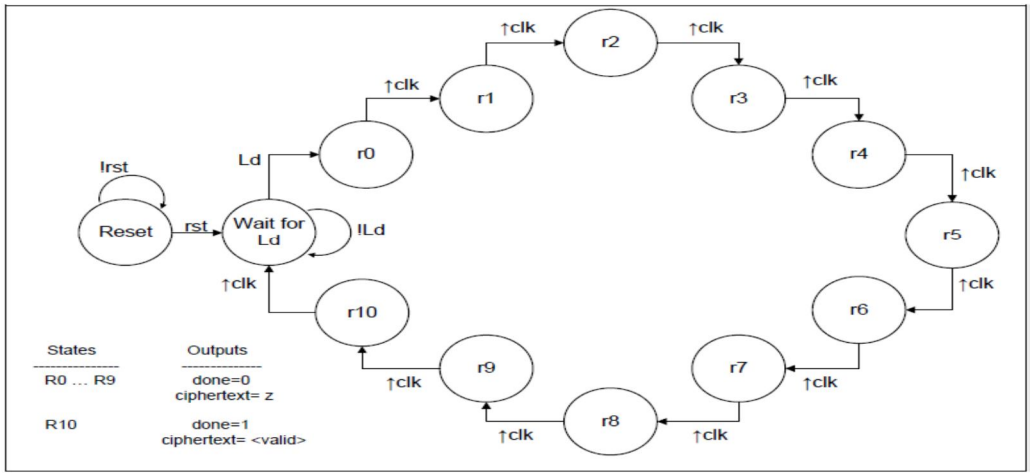


Figure 2: AES128 Cipher Top Module State Diagram

The AES128_Key_Expand module generates four 32-bit keys for each round of the encryption process, by using the cipher key. The cipher key is passed to this module through a 128-bit input port, and the round keys are generated on the four output ports.

Fig 3: AES128 Key expand Module

III. HARDWARE USED

A. FPGA (Field Programmable Gate Array)

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by the customer or designer after manufacturing—hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC) (circuit diagrams were previously used to specify the configuration, as they were for ASICs, but this is increasingly rare). FPGAs can be used to implement any logical function that an ASIC could perform.

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together"—somewhat like a one-chip programmable breadboard.

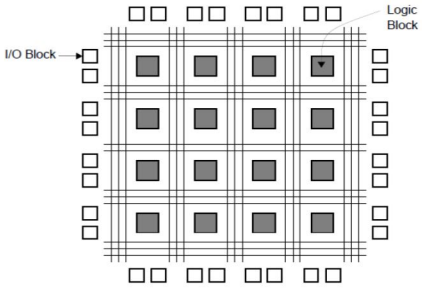


Fig 4: Structure of FPGA

Dedicated memory blocks offer data storage and can be configured as basic single-port RAMs, ROMs (read only memory), FIFOs (first in first out), or CAMs (Content Addressable m\Memory). Data processing or the logic fabric of these FPGAs varies widely in size with the biggest Xilinx Virtex-II Pro™ offering up to 100K LUT4s.

B. Altera DE0 Board

The DE0 board has many features that allow the user to implement a wide range of designed circuits, from simple circuits to various multimedia projects. The following hardware is provided on the DE0 board: Altera Cyclone® III 3C16 FPGA device, Altera Serial Configuration device – EPCS4

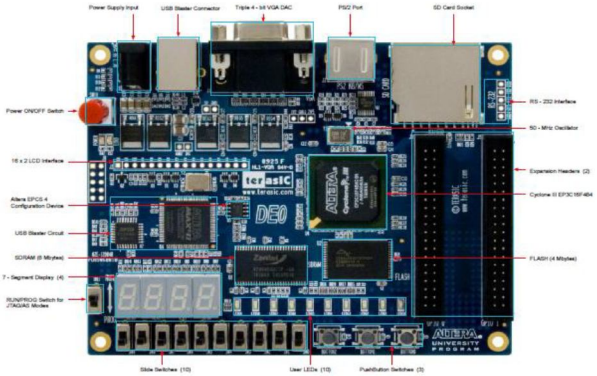


Fig 5: DE0 Board

IV. SOFTWARE USED

We have used Modelsim, and Quarters II.

A. ModelSim

Mentor Graphics was the first to combine single kernel simulator (SKS) technology with a unified debug environment for Verilog, VHDL, and SystemC. The combination of industry-leading, native SKS performance with the best integrated debug and analysis environment make ModelSim the simulator of choice for both ASIC and FPGA design. The best standards and platform support in the industry make it easy to adopt in the majority of process and tool flows.

ModelSim-Altera Edition

- Recommended for simulating all FPGA designs (Cyclone®, Arria®, and Stratix® series FPGA designs)
- 33 percent faster simulation performance than ModelSim®-Altera® Starter Edition.

ModelSim-Altera Starter Edition

- Support for simulating small FPGA designs
- 10000 executable line limitations

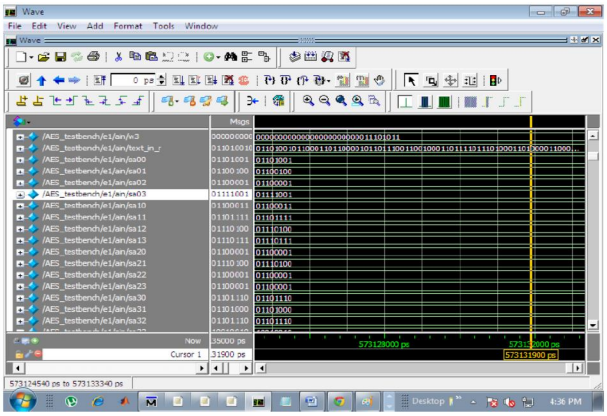


Fig 6: Modelsim Output

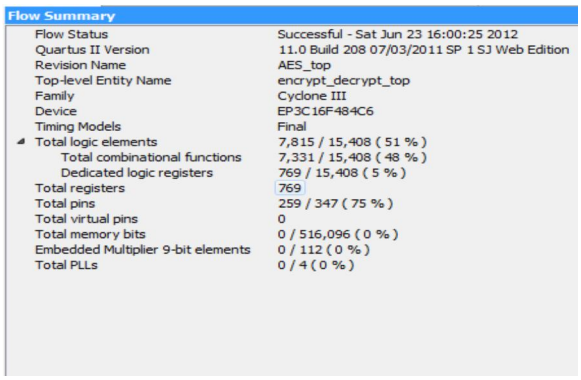


Fig 7: Flow Summary Report

B. QUARTUS II:

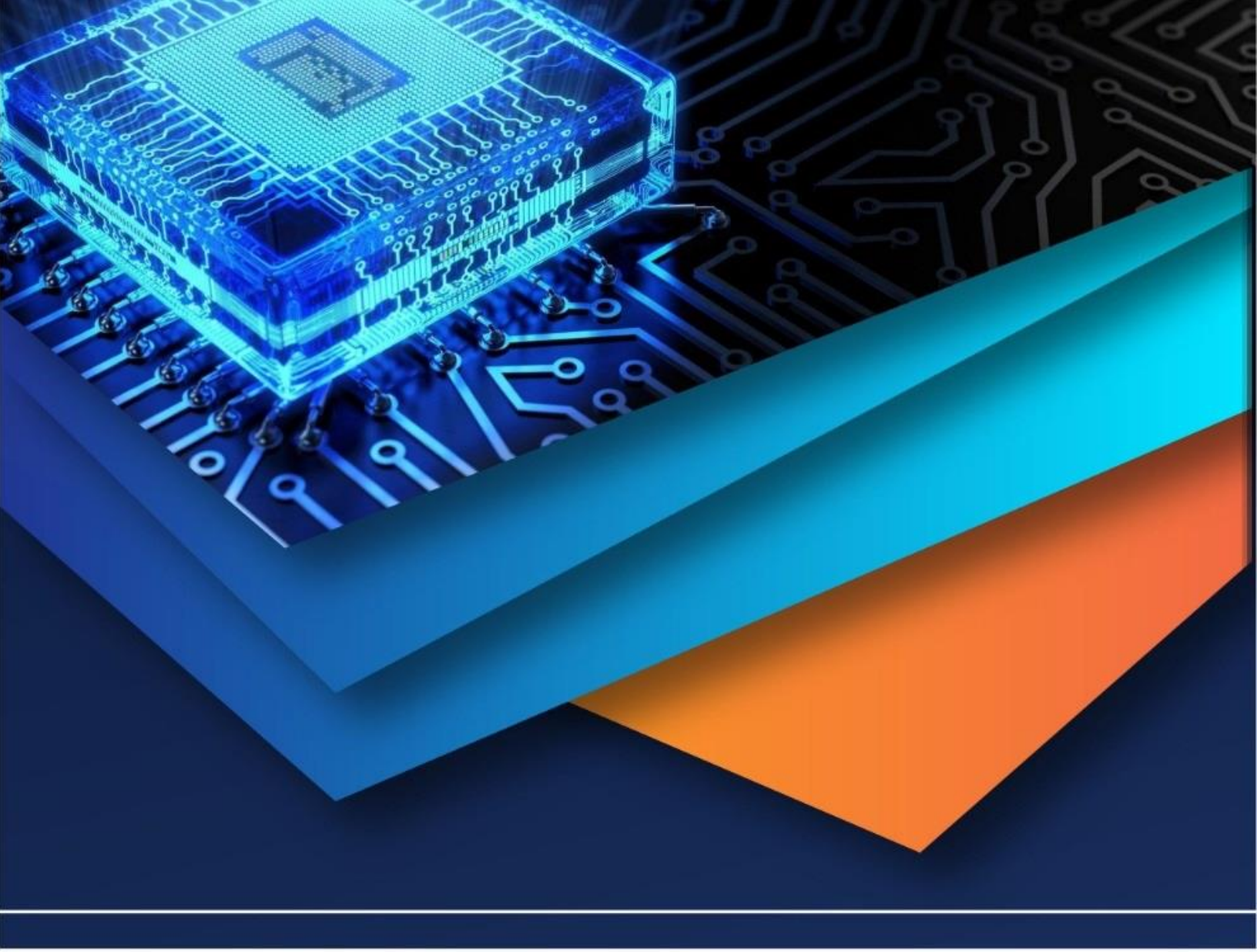
Quartus II is a software tool produced by Altera for analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer.

V. CONCLUSION

Here in this we carried out implementation of Nano-AES cryptographic algorithms for IoT with scan-based testing futures. Compared to regular scan tests, this technique has no impact on the quality of the test or the model-based fault diagnosis. Here we proved that RSFF based AES will give better hardware complexity & power optimization with considerable delay enhancement. An accurate SFF-based analysis approach was introduced for AES core with single and multi-FF characterizations. The proposed approach was derived from the SFF method. The method avoids the use of a large number of masking parameters to minimize the required resources for area and power-efficient built-in testing applications. Modelsim based pre simulation results of an AES implementation showed the feasibility of the approach. For a QUARTUS II based hardware synthesis report proved the efficiency of proposed method.

REFERENCES

[1] Ferrari, P.; Sisinni, E.; Bellagente, P.; Rinaldi, S.; Pasetti, M.; de Sa, A.O.; Machado, R.C.S.; Carmo, L.F.R.D.C.; Casimiro, A. Model-Based Stealth Attack to Networked Control System Based on Real-Time Ethernet. IEEE Trans. Ind. Electron. 2021, 68, 7672–7683.
[2] Zeebaree, S.R.M. DES encryption and decryption algorithm implementation based on FPGA. Indones. J. Electr. Eng. Comput. Sci. 2020, 18, 774–781.
[3] Shahbazi, K.; Ko, S.-B. Area-Efficient Nano-AES Implementation for Internet-of-Things Devices. IEEE Trans. Very Large Scale Integr. VLSI Syst. 2021, 29, 136–148.
[4] M. Akkar and C. Giraud, "An Implementation of DES and AES, Secure against Some Attacks," In Proc. of the Workshop on Cryptographic Hardware and Embedded Systems (CHES2001), Paris, France, pp. 315-325, May 2001.
[5] http://www.altera.com/products/software/products/quartus2/qts-index.html
[6] R. Anderson, E. Biham, and L. Knudsen, "Serpent: A Proposal for the Advanced Encryption Standard," AES algorithm submission, June 1998.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY