8 - Bit Processor for Cryptography

Documentation Report

Sayam Chakraborty[1], Naman Nagendra Bhat[2], Aryan Gujral[3], and Ashwani Kumar[4]

[1]Department of Electronics and Communication Engineering, Indian Institute of Space Science and Technology

December 15, 2024

Abstract

Data exchange between a sender and a recipient without third-party involvement demands robust security measures. Ensuring secure transmission and reception of data is essential to prevent unauthorized access to systems, websites, and databases. Security involves protecting sensitive information and maintaining its integrity during communication. With continuous advancements, innovative solutions are developed to address emerging security challenges. Cryptography has become a vital technique to improve data security. Facilitates the encryption and decryption of information, ensuring secure communication between parties. This methodology prevents unauthorized individuals from accessing sensitive data transmitted through communication channels. Encryption transforms the data into an unreadable format at the sender's end, while decryption restores it to its original form at the receiver's end. These processes ensure that transmitted data remain confidential and unaltered throughout the communication. Using cryptographic techniques, we can achieve the privacy, confidentiality, and integrity of the data exchanged.

To demonstrate the practical implementation of cryptography, we designed and developed a Verilog-based system for performing encryption and decryption. This approach validates the effectiveness of cryptography in safeguarding data transmission and showcases its utility in real-world scenarios. This report documents the design and implementation of an 8-bit processor. The processor is designed to perform encryption, decryption, and control operations. Verilog HDL was used for hardware description and simulation and synthesis were performed using Xilinx Vivado. The project aims to explore the fundamental principles of processor design and FPGA-based implementation.

Contents

1 Introduction

In this project, we designed a microprocessor dedicated to encryption and decryption processes. The microprocessor integrates an Arithmetic Logic Unit (ALU) capable of performing encryption and decryption using custom logic. The ALU operates with a Linear Feedback Shift Register (LFSR) to generate random keys for the encryption process, ensuring security and reliability.

1.1 Motivation

The rapid advancements in FPGA technology have made it an essential platform for prototyping custom hardware solutions. The 8-bit processor project serves as an excellent educational exercise in understanding the nuances of computer architecture, digital design, and how it can be implemented in real-life applications like cryptography.

1.2 Objectives

The primary goal is to create a hardware design capable of securely encrypting and decrypting 8-bit data. The system employs scalar multiplication, constant addition, swapping, and one-time padding with the pseudo-random keys generated from the LFSR module for encryption; while inverse operations are used for decryption.

2 Processor Architecture

2.1 Block Diagram

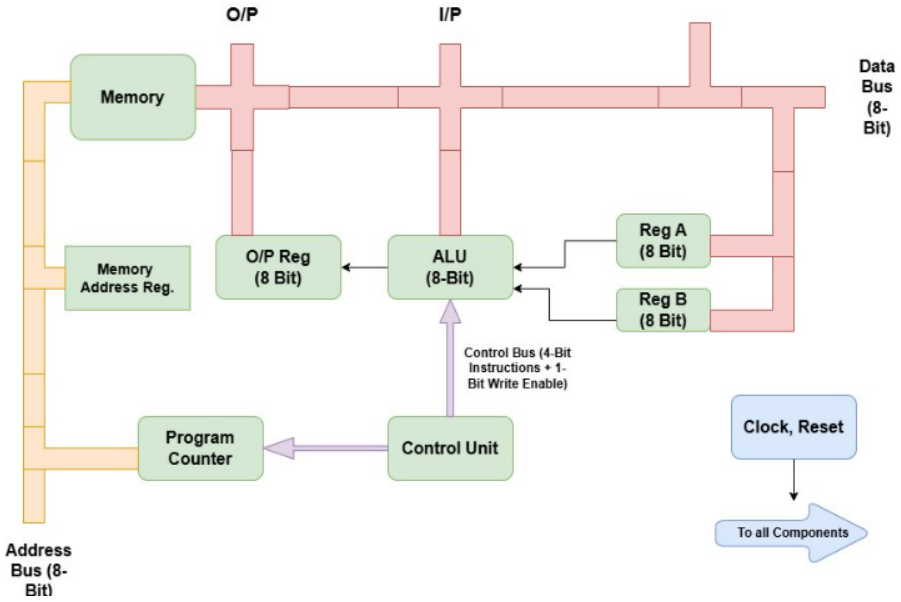

Figure 1: Block Diagram of 8-bit Processor

2.2 Modules

- **Arithmetic and Logical Unit (ALU):** Performs arithmetic and logical operations.
- **Control Unit:** Decodes instructions and manages data flow.
- **Registers:** Stores intermediate results and data.
- **Instruction Memory:** Stores the program instructions.
- **Data Memory:** Stores data used during computation.

3 Methodology of Encryption and ALU Design

3.1 ALU Design for Encryption and Decryption

The core of our processor is a custom-designed Arithmetic Logic Unit (ALU), dedicated to the processes of encryption and decryption. The ALU operates with an 8-bit architecture and integrates multiple operations to ensure secure and efficient cryptographic transformations. Its primary functions include scalar operations, pseudo-random key generation using a Linear Feedback Shift Register (LFSR), and data swapping, culminating in the encryption of input data. The same ALU also performs the reverse operations to achieve decryption.

3.2 Key Operations in the ALU

3.2.1 Scalar Multiplication and Addition

- The input data undergoes scalar multiplication by a fixed value. This operation serves as the first layer of transformation, introducing complexity to the encryption process.
- A scalar quantity is then added to the result, further modifying the data before the application of the cryptographic key.

3.2.2 LFSR-Based Pseudo-Random Key Generation

- A Linear Feedback Shift Register (LFSR) is employed to generate an 8-bit pseudo-random key. The LFSR operates by shifting data through registers and applying a circular feedback mechanism.
- The feedback is created by XORing the data at specified tap positions, which are carefully chosen to maximize randomness and periodicity.
- The LFSR is initialized with a fixed seed value, ensuring predictable and reproducible pseudo-random key sequences. This property makes it a reliable component of our encryption and decryption system.

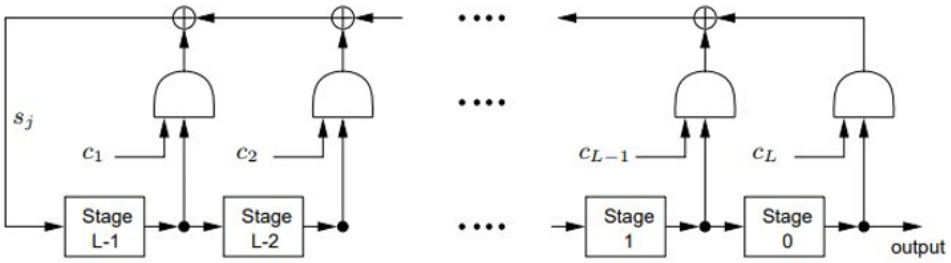

Figure 2: A Linear Feedback Shift Register of length L

3.2.3 Data Swapping

- Before applying the XOR operation with the pseudo-random key, the data undergoes a bit-swapping process.
- This involves keeping the initial and final bits of the data unchanged while reversing the order of the intermediate bits. This step adds an additional layer of security, making it more resistant to unauthorized decoding.

3.2.4 XOR with Random Key

- The output of the previous stages is XORed with the pseudo-random key generated by the LFSR.
- This final step ensures that the encrypted data is both unique and difficult to reverse without the corresponding decryption process.

3.3 Decryption Process

The ALU performs decryption by reversing the encryption steps:

- XORing with the same pseudo-random key.
- Re-swapping the bits.
- Subtracting the scalar quantity.
- Dividing by the scalar multiplier.

These operations sequentially restore the original data with high accuracy, maintaining data integrity.

3.4 Advantages of the ALU Design

The modular and systematic design of the ALU ensures:

- Efficient hardware implementation with an 8-bit architecture.
- High-security encryption through the combination of pseudo-random key generation, data swapping, and arithmetic operations.
- Reproducibility and predictability using a seed-based LFSR for key generation.

This ALU serves as the cryptographic backbone of our processor, enabling robust encryption and decryption for secure data transmission.

4 Control Unit

The Control Unit (CU) is responsible for controlling the flow of data within the processor. It receives input from external data and memory, decodes the instructions, and issues control signals to memory and registers. The CU also interacts with the ALU to manage encryption and decryption operations.

4.1 Control Unit Functionality

The Control Unit is implemented using Verilog HDL and processes the instructions and manages data flow. It fetches instructions from memory, decodes them, and provides appropriate control signals to memory, ALU, and registers.
The CU also has a manual write enable signal that allows for manual data manipulation.

4.2 Various Control Signals

We have 7 Control Signals:

- **Clock, Reset** – To all components.
- **Manual Write Enable** – To RAM from user.
- **Control Write Enable** – To RAM.
- **3 Bit Instructions** – To Instruction Decoder.

5 Instruction Set

The processor uses a simple instruction set to manage encryption and decryption operations. The instruction set of our processor is very small compared to a general purpose processor due to its specific application. The key instructions in the instruction set include:

- **LD IMM A**: Load immediate data into register A.
- **LD IMM B**: Load immediate data into register B.
- **ALU ENCRYPT**: Perform encryption using the ALU.
- **ALU DECRYPT**: Perform decryption using the ALU.

These instructions interact with the Control Unit to trigger corresponding actions in the processor, enabling secure communication by encrypting and decrypting data.

6 Results and Analysis

6.1 Simulation Results for Individual Components

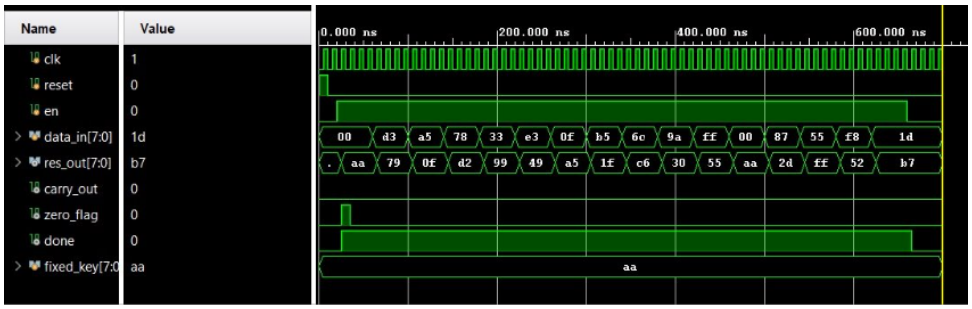

Figure 3: Simulation Waveforms for ALU Operations

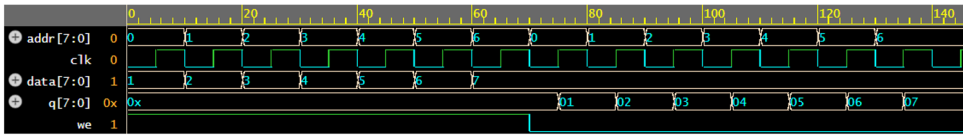

Figure 4: Simulation Waveforms for RAM Operations

6.2 Simulation Results for Complete Processor

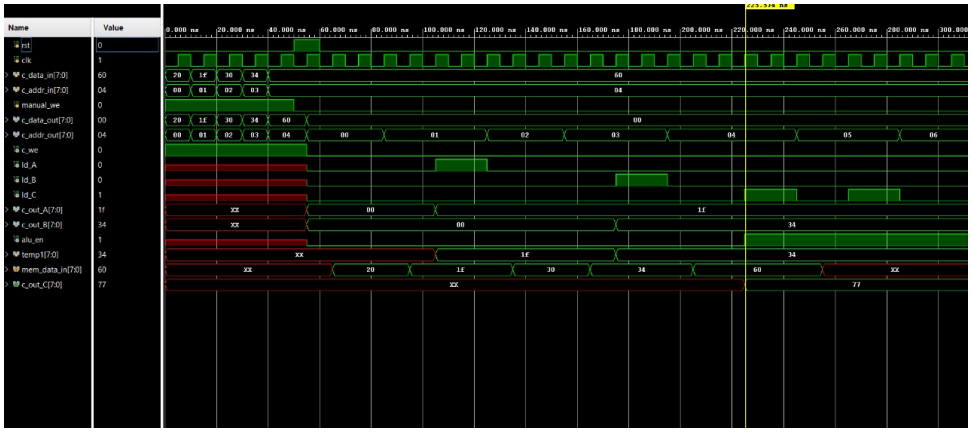

Figure 5: Simulation Waveforms for One Complete Cycle of Encryption - Decryption by Processor

7 Hardware Testing

The Processor was successfully implemented using Xilinx Vivado and tested on the FPGA.

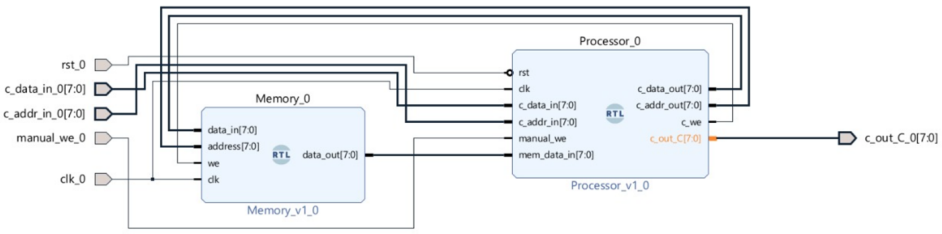

Figure 6: Vivado generated Block Diagram of the Processor

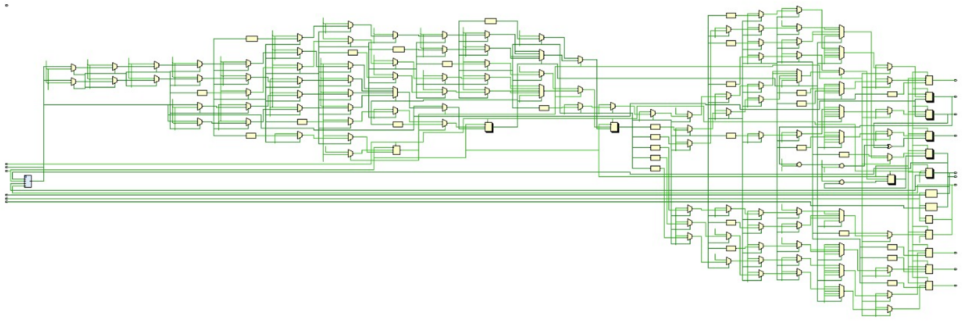

Figure 7: Vivado generated Schematic of the Processor

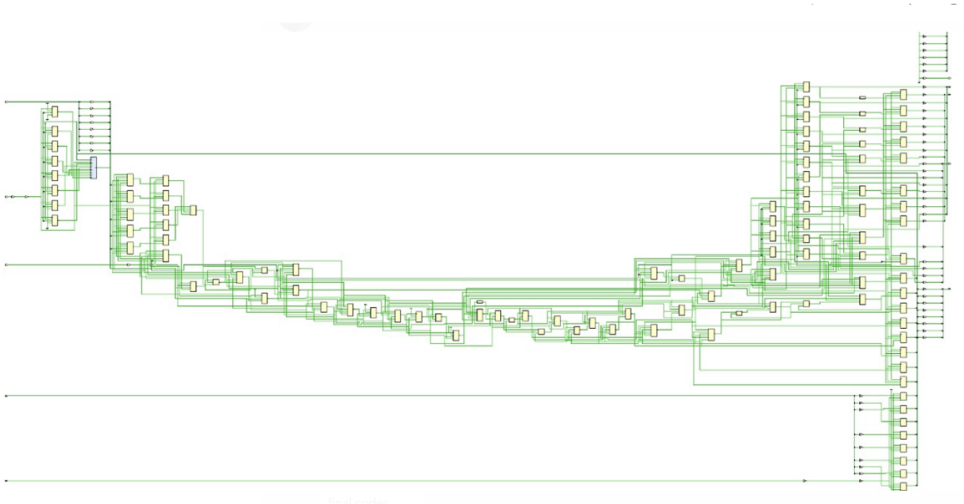

Figure 8: Vivado generated Synthesized Schematic of the Processor

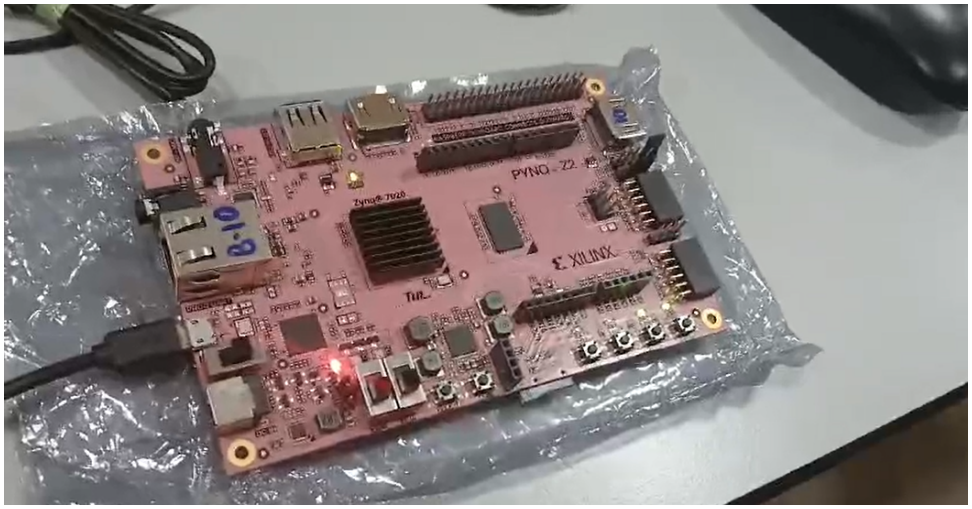

Figure 9: FPGA Implementation of the Processor

8 Conclusion

This project presented the design of an 8-bit processor dedicated to encryption and decryption, leveraging a modular architecture comprising an Arithmetic Logic Unit (ALU), memory, and control unit. The ALU integrates essential cryptographic operations, including scalar transformations, pseudo-random key generation using an LFSR, and data swapping, ensuring robust encryption and decryption capabilities. The systematic and hardware-efficient design ensures data security and integrity, addressing the critical need for secure communication in modern embedded systems.

The processor's ability to perform reversible encryption and decryption, with high accuracy and reproducibility, highlights its potential for integration into secure IoT devices, lightweight cryptographic systems, and embedded applications requiring minimal hardware overhead.

8.1 Future Prospects

The current design forms a foundation for further enhancements in the field of secure processors. Future research can focus on:

- Scalability: Extending the architecture to support higher bit-widths for more complex encryption schemes.
- Advanced Cryptographic Algorithms: Integrating support for standard algorithms such as AES or RSA, alongside custom designs.
- Dynamic Key Generation: Developing adaptive or multi-seed LFSR configurations for improved randomness and enhanced security.
- Power and Area Optimization: Refining the design to meet stringent constraints in low-power and compact embedded systems.
- Secure Communication Protocols: Implementing the processor in end-to-end secure communication systems with real-time performance evaluation.

- Hardware Security Features: Adding countermeasures against hardware-based attacks, such as side-channel or fault injection attacks.

This work establishes a pathway for developing more secure and efficient processors, contributing to the advancement of secure digital communication and embedded system design.

9 References

1. John L. Hennessy and David A. Patterson, *Computer Architecture: A Quantitative Approach*, Morgan Kaufmann, 2017.

2. Samir Palnitkar, *Verilog HDL: A Guide to Digital Design and Synthesis*, Prentice Hall, 2003.

3. Xilinx Documentation, `https://www.xilinx.com/`

10 Acknowledgments

We would like to express our gratitude to our professor, **Dr. B.S. Manoj**, for his valuable guidance and support throughout the project. We, the **Team 8-Bit Johnny** appreciate his insights and encouragement, which have been pivotal in the completion of this project.