VIETNAM JAPAN UNIVERSITY
VIETNAM NATIONAL UNIVERSITY, HANOI

Soc Design for RSA Asymmetric Cryptography

Students : Nguyen Duc Hieu
: Le Minh Kiet

Class : BCSE2022

Supervisors : PhD. Tran Thi Diem
: PhD. Nguyen Van Tinh

Hanoi, June 6, 2025

Abstract

This report details the implementation and analysis of an RSA cryptographic accelerator on a Xilinx Zynq UltraScale+ MPSoC platform. A custom RSA IP core was integrated into the programmable logic (PL) and connected to the processing system (PS) via an AXI4-Lite interface, enabling memory-mapped control and data transfer. The complete system-on-chip (SoC) design was successfully synthesized, implemented, and a bitstream was generated for the `xck26-sfvc784-2LV-c` device. This report presents a comprehensive analysis of the post-synthesis resource utilization, power consumption, and timing performance. The results indicate significant usage of LUTs and dedicated DSP slices, confirming the hardware acceleration of arithmetic operations. However, the design failed to meet timing constraints, highlighting the need for further pipeline optimization in future work.

1 Introduction

The Rivest-Shamir-Adleman (RSA) algorithm is a cornerstone of modern public-key cryptography. However, its computationally intensive nature, particularly the modular exponentiation operation, presents a significant bottleneck for software-only implementations. Hardware acceleration using Field-Programmable Gate Arrays (FPGAs) offers a path to substantially improve performance. This project focuses on the integration of a custom-designed RSA cryptographic module as an IP core within a Zynq UltraScale+ MPSoC, creating a complete hardware-software co-design platform. The goal is to deploy the accelerator in the programmable logic (PL) and make it accessible to the ARM processing system (PS), laying the groundwork for a high-performance cryptographic system.

2 System Architecture and Integration

The core of the design is a Zynq UltraScale+ MPSoC, which combines a powerful multicore processing system with flexible programmable logic. Our custom RSA module, packaged as an IP core named `MY_IP_0`, was instantiated in the PL.

Communication between the PS and the RSA accelerator is facilitated by the AXI protocol. An `AXI SmartConnect` IP was used to bridge the master AXI port from the Zynq PS to the slave `S_AXI` port of our RSA IP. This configuration allows the processor to control the accelerator by writing to and reading from its memory-mapped registers, treating the hardware as a peripheral device. The overall system architecture is depicted in Figure 1.

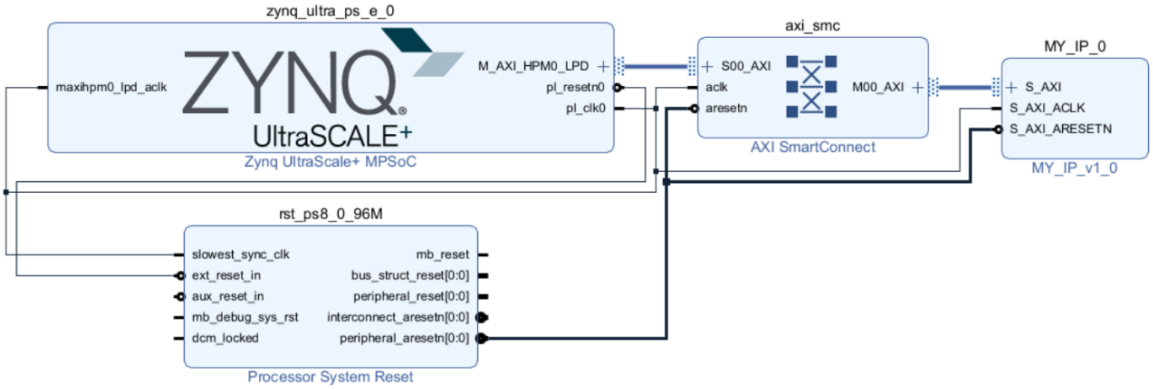

Figure 1: Vivado Block Design showing the Zynq MPSoC, AXI SmartConnect, and the custom RSA Accelerator IP (`MY_IP_0`).

3 Implementation Flow

The design was implemented using the Vivado Design Suite v.2022.2. The flow consisted of creating the block design in the IP Integrator, generating the HDL wrapper, running synthesis to translate the design into a netlist of logic gates, and finally, implementation (place and route) to fit the design onto the target device. The process concluded with the successful generation of a bitstream file.

- **Target Device:** xck26-sfvc784-2LV-c
- **Design State:** Synthesized, Implemented, Bitstream Generated.

4 Results and Analysis

After successful bitstream generation, a series of reports were generated to analyze the performance and resource footprint of the design.

4.1 Resource Utilization

The post-synthesis utilization report provides a detailed summary of the hardware resources consumed by the RSA accelerator. The key metrics are summarized in Table 1.

Table 1: Post-Synthesis Resource Utilization Summary.

Site Type	Used	Available	Util %
CLB LUTs	27,268	117,120	23.28
- LUT as Logic	26,951	117,120	23.01
- LUT as Memory	317	57,600	0.55
CLB Registers (Flip Flop)	2,677	234,240	1.14
CARRY8	3,145	14,640	21.48
DSPs	44	1,248	3.53
Block RAM Tile	0	144	0.00

The report reveals several key insights:

- **Logic Complexity:** The design consumes a significant portion of LUTs (23.28%), indicating complex combinational logic, which is expected for cryptographic algorithms.
- **Hardware Multiplication:** The use of 44 DSP slices is a critical finding. It confirms that the multiplication-heavy operations within the RSA algorithm are being mapped to dedicated, high-performance hardware blocks rather than being implemented in general-purpose logic.
- **Memory Usage:** The design does not use any Block RAM (BRAM) tiles. This suggests that any state or intermediate values are stored in distributed RAM (using LUTs) or directly in flip-flops.

4.2 Timing and Power Analysis

Timing Performance: The post-implementation timing summary reported a Worst Negative Slack (WNS) of **-153.507 ns**. This is a critical result, indicating that the design **failed to meet timing constraints**. The logical paths within the design are too long to propagate signals within a single clock cycle of the target frequency. This is a common challenge in complex designs and points to a clear area for optimization.

Power Consumption: The power analysis, shown in Figure 2, estimated a total on-chip power consumption of **2.788 W**. It is important to note that the vast majority of this power (2.487 W, or 89%) is dynamic power, and most of that is consumed by the Zynq Processing System (PS) itself (2.21 W). The custom logic implemented in the PL accounts for a relatively small fraction of the total power.

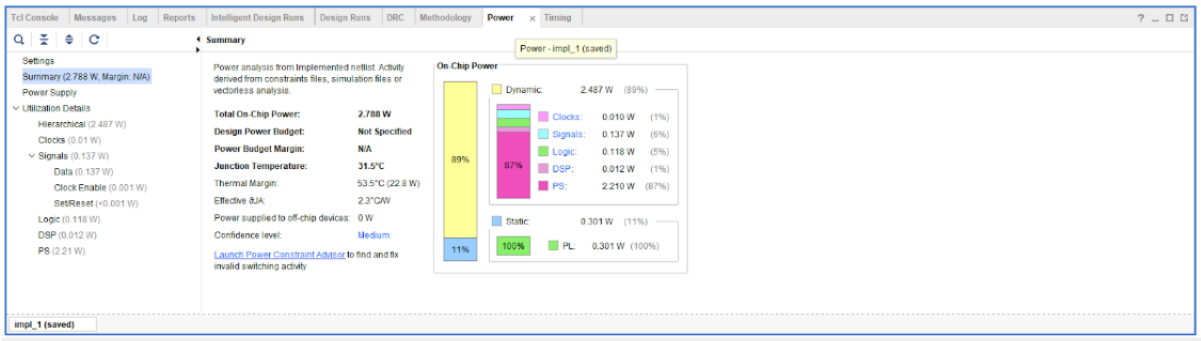

Figure 2: Post-implementation power consumption report.

5 Conclusion and Future Work

This project successfully demonstrated the integration of a custom RSA accelerator into a Zynq UltraScale+ MPSoC design. The entire flow from block design to bitstream generation was completed, and a detailed analysis of the implementation results was performed. The resource utilization report confirms that the design leverages dedicated DSP slices for arithmetic acceleration.

The most significant challenge identified is the failure to meet timing requirements, as indicated by the large negative slack. This must be addressed before the system can be reliably deployed.

Based on these findings, future work should focus on the following areas:

1. **Timing Optimization:** The Verilog HDL for the RSA module must be refactored. Introducing more pipeline stages within the critical path of the modular exponentiation algorithm will be the primary strategy to reduce logic delays and meet timing.

2. **Clock Management:** Experiment with a lower clock frequency for the programmable logic to find a frequency at which the current design can operate correctly while optimizations are being developed.

3. **Functional Verification:** Once timing is met, develop a bare-metal C/C++ application to run on the PS. This software will control the accelerator via the AXI interface to perform encryption/decryption tasks and verify the functional correctness of the hardware in a real-world scenario.