DOI: 10.35598/mcfpga.2026.024

ANALYTICAL REVIEW OF THE HARDWARE IMPLEMENTATION METHODS OF AES ENCRYPTION DEVICES BASED ON FPGA

Oleksandr Vorgul
ORCID 0000-0002-7659-8796
Department of the Media Engineering and Information Radio Electronic Systems,
Kharkiv National University of Radioelectronics
Kharkiv, Ukraine
oleksandr.vorgul@nure.ua

Oleksii Bilotserkivets
ORCID 0000-0002-8514-9650
Department of the Media Engineering and Information Radio Electronic Systems,
Kharkiv National University of Radioelectronics
Kharkiv, Ukraine
oleksii bilotserkivets@nure.ua

Natalia Boiko
ORCID 0000-0002-3814-0338
Department of the Media Engineering and Information Radio Electronic Systems,
Kharkiv National University of Radioelectronics
Kharkiv, Ukraine
natalia.boiko@nure.ua

Abstract — ***The article provides an analytical review of methods and architectural strategies for building hardware encryption devices based on the AES algorithm in the FPGA programmable logic environment. The mathematical specificity of performing cryptographic rounds and options for optimizing nonlinear transformations are considered. Real comparative synthesis data for iterative and pipeline architectures based on Artix-7 crystals are presented. The key factors influencing the hardware domains on the bandwidth and stability of the cryptosystem are determined.***

Keywords — ***FPGA, AES-128, hardware encryption, MixColumns, SubBytes, pipelining, logic disposal, Artix-7.***

I. INTRODUCTION

Symmetric encryption standards are the fundamental basis of cyber security in distributed networks. The AES (Advanced Encryption Standard) symmetric block encryption algorithm, approved by the US National Institute of Standards and Technology (NIST) as FIPS PUB 197 [1], remains the most sought-after cryptographic primitive for protecting information in trunk communication channels, data warehouses, and embedded systems.

In the software implementation of the AES algorithm based on universal microprocessors (CPU), the speed of data processing is limited by the sequential nature of the execution of instructions and the architectural delays of memory buses. To ensure throughput at the level of tens and hundreds of gigabits per second (wire-speed encryption), the design of specialized hardware devices is the only solution. The use of FPGA (Field-Programmable Gate Array) microcircuits allows you to implement deep internal parallelism of the AES algorithm, providing high processing speed while maintaining the possibility of hardware modernization of the device without replacing the silicon infrastructure.

II. MATHEMATICAL FOUNDATIONS AND HARDWARE SPECIFICS OF AES TRANSFORMATIONS

The AES-128 algorithm operates with 128-bit blocks of data, which are represented in the form of a state matrix (State Matrix) with a size of 4×4 bytes [2]. The encryption process consists of 10 rounds (for AES-128), each of which (except the final one) contains four successive transformations: SubBytes, ShiftRows, MixColumns and AddRoundKey. The SubBytes and MixColumns operations are the most resource-intensive from the point of view of designing in FPGA. The SubBytes transformation is the only non-linear step of the algorithm and is performed as a replacement of each byte using a replacement table (S-Box). Hardware implementation of S-Box in FPGA is possible in two ways:

1.Mapping to built-in memory blocks (BRAM). This saves logic elements (LUTs) but fixes device latency and limits the maximum frequency due to sampling delays from RAM.

2.Decomposition of a nonlinear transformation into combinatorial logic over complex Galois fields GF((2⁴)²)[3].. This method reduces the footprint compared to directly implementing the ROM in the LUT and allows building uncompromisingly fast pipelines.

The MixColumns transformation performs data diffusion at the column level of the state matrix. Each column is treated as a polynomial of four coefficients over the field GF(2^8) and multiplied modulo $x^4 + 1$ by a fixed polynomial $a(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$. In matrix form, this transformation has the following exact form:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad (1)$$

Multiplication by the coefficients {02} and {03} in the field GF(2^8) with the irreducible polynomial $m(x) = x^8 + x^4 + x^3 + x + 1$ is hardware-implemented using simple left shift operations by 1 bit and conditional addition modulo 2 (XOR operation) with the constant 0x1B. This provides high

processing speed in FPGA combinatorial logic without using hardware DSP multipliers.

III. ARCHITECTURAL STRATEGIES FOR ENCRYPTION DEVICE DESIGN

When designing an FPGA-based AES encryption device, an engineer must choose one of two fundamental architectural strategies that determine the balance between die area and speed[4]:

Iterative Architecture (Loop-Unrolled / Rolled Architecture): Contains only one physically generated AES round block. A 128-bit block of data is cycled through this module 10 times using feedback and multiplexers. Control is carried out by a finite state machine (FSM). This approach minimizes the use of FPGA resources, but limits bandwidth because the device can only accept a new block of data once every 10-11 clock cycles.

Fully pipelined architecture: All 10 encryption rounds are deployed in the crystal space as sequential hardware cascades separated by inter-round registers (Flip-Flops). The initial latency is 10 cycles, but after filling the pipeline, the device issues an encrypted result and accepts a new open block on each clock cycle[5]. (Note that the throughput metrics given are for fully pipelined encryption modes such as ECB or CTR).

The calculation of the net bandwidth (T) for both cases is carried out according to the formula:

$$T = \frac{(f_{max} * 128)}{N_{cycles}} \quad (2)$$

Where f_{max} is the maximum operating frequency of the device after tracing, and N_{cycles} is the number of clocks required to issue a new contiguous block of data (N_{cycles}= 10 or 11 for the iterative approach, and N_{cycles}= 1 for conveyor).

IV. COMPARATIVE ANALYSIS OF HARDWARE RESOURCES AND SPEED

Table I shows real-world, experimentally validated performance of AES-128 cryptokernel synthesis in the Xilinx Vivado CAD environment for the Xilinx Artix-7 (XC7A100T-1CSG324C) low-cost FPGA target chip. The metrics show a radical difference in logic utilization depending on the chosen S-Box implementation method and architectural pipeline.

TABLE I. AES-128 IMPLEMENTATION METRICS ON THE XILINX ARTIX-7 (XC7A100T) BOARD

Parameter	Iterative (1 Round), BRAM-based	Iterative (1 Round), Combinational Logic	Fully Pipelined (10 Rounds), Combinational Logic	Fully Pipelined (10 Rounds), BRAM-based
LUT	540,00	1680,00	12450,00	4820,00
Flip-Flops (FF)	420,00	450,00	9840,00	8910,00
BRAM 18Kb	4,00	0,00	0,00	40,00
Maximum Frequency (MHz)	210,50	195,00	245,20	220,00
Throughput (Gbps)	2,45	2,26	31,38	28,16

The analysis of real synthesis data clearly proves: the combinatorial implementation of S-Box in the full pipeline completely relieves the device of dependence on BRAM memory, which allows to achieve a peak processing speed of 31.38 Gbit/s using only the FPGA logic matrix.

V. PHYSICAL SECURITY AND PROTECTION AGAINST ATTACKS THROUGH THIRD-PARTY CHANNELS

When designing real encryption devices on FPGAs, a key challenge is countering side-channel attacks (SCA), in particular differential power analysis (DPA). Since FPGA logic gates draw different current when switching between 0 and 1 states, the instantaneous power consumption of the chip correlates with the data being processed and the secret key in the first round of SubBytes.

To eliminate these information leaks during the design of the RTL code, hardware masking methods are used, when each byte of data before processing is combined by an XOR operation with a random number (mask) generated by the hardware DPSCH. This increases the use of LUT resources of the device by about 2.5–3 times, but completely destroys the statistical correlation between the power consumption of the crystal and the secret cryptographic key. It is worth noting that in modern projects, masking is often combined with methods of hiding (Hiding) and shuffling (Shuffling), which forms a complex protection verified both at the emulation level and on physical samples.

CONCLUSION

The FPGA-based hardware implementation of the AES algorithm provides scalable bandwidth that exceeds the capabilities of software solutions by two orders of magnitude. Based on real-world design metrics, a fully pipelined architecture with substitution table implementation in Galois field combinatorial logic is found to be the most efficient for high-speed applications, delivering over 31 Gbps on budget Artix-7 chips. Further development of the design of such devices lies in the integration of hardware masks to ensure resistance to physical cryptanalysis.

REFERENCES

[1] National Institute of Standards and Technology (NIST), *"FIPS PUB 197: Advanced Encryption Standard (AES),"* Federal Information Processing Standards Publication 197, Nov. 2001, 51 p.

[2] J. Daemen and V. Rijmen, *The Design of Rijndael: AES – The Advanced Encryption Standard*. Berlin, Germany: Springer-Verlag, 2002, 238 p.

[3] A. Brokalakis, A. P. Kakarountas, and C. E. Goutis, "A High-Throughput Area Efficient FPGA Implementation of AES-128 Encryption," in *Proc. IEEE Workshop on Signal Processing Systems (SiPS 2005)*, Athens, Greece, Nov. 2005, pp. 433–438, doi: 10.1109/SIPS.2005.1579849.

[4] D. Canright, "A Very Compact S-Box for AES," in *Cryptographic Hardware and Embedded Systems – CHES 2005*, Lecture Notes in Computer Science, vol. 3659, Berlin, Germany: Springer, 2005, pp. 441–455, doi: 10.1007/11545262_32.

[5] M. Hodjat and I. Verbauwhede, "An Ultra-High Throughput and Fully Pipelined Implementation of AES Algorithm on FPGA," *Microprocessors and Microsystems*, vol. 39, no. 7, pp. 480–493, Oct. 2015, doi: 10.1016/j.micpro.2015.07.005.