

## Ascon-128 AEAD Encryptor IP Core

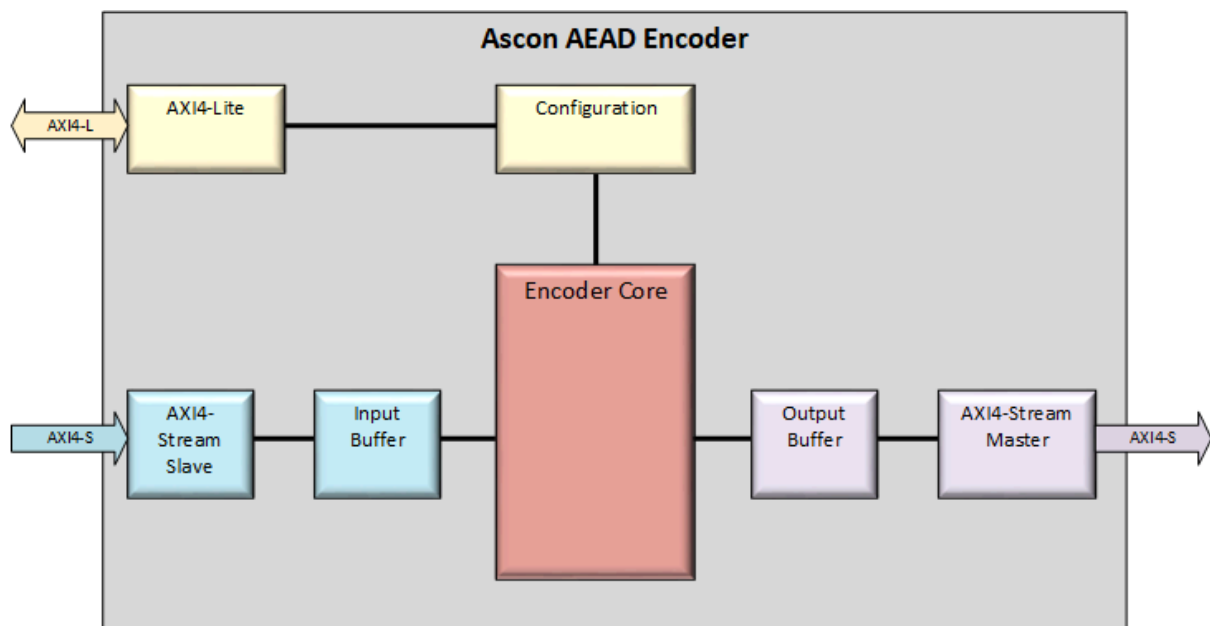
High-Performance, Lightweight Authenticated Encryption for Secure Data Streams

### Overview

The Ascon-AD128-ENC is a high-efficiency, silicon-proven hardware IP core that implements the Ascon-128 authenticated encryption with associated data (AEAD) algorithm, a NIST Lightweight Cryptography Standard (2023). Designed for seamless integration into System-on-Chip (SoC) architectures, this core provides robust confidentiality, integrity, and authenticity for high-speed data streams with minimal resource footprint and low latency.

It is the ideal solution for securing data in constrained environments such as IoT edge devices, embedded systems, automotive networks, and aerospace applications, where performance, power, and area (PPA) are critical.

### Block Diagram



## Key Features & Benefits

Feature	Benefit
<b>NIST Standard Compliance</b>	Implements the certified, future-proof Ascon-128 algorithm, ensuring interoperability and regulatory acceptance.
<b>AXI4-Stream Interfaces</b>	Plug-and-play integration with modern SoC dataflow architectures. Simplifies connection to DMA controllers, communication IPs, and processors.
<b>AXI4-Lite Control Interface</b>	Standard, simple register programming model for easy CPU control and integration.
<b>High Degree of Configurability</b>	Tailor the IP to your exact performance, area, and security requirements. Optimize for your target technology node and application.
<b>Side-Channel Resistance</b>	Optional <b>Random Delay</b> feature disrupts timing patterns, providing a hardened layer of defense against simple power and timing analysis (SPA/STA).
<b>Multi-Context Operation</b>	Up to 4 independent configuration sets (Key, IV, Mode) allow instant switching between different security contexts or data streams without software overhead.
<b>High-Throughput Task Queue</b>	Submit up to 16 encryption/decryption tasks in a single burst, freeing the software to perform other operations and maximizing data movement efficiency.
<b>Small &amp; Efficient</b>	Exceptionally low

	resource utilization (see below) makes it suitable for the most area-sensitive designs without sacrificing performance.
--	---

## Performance & Utilization

### Summary

The following resource utilization is for a minimally configured instance (prioritizing smallest area) synthesized for a typical FPGA target.

Resource Type	Utilized
<b>Slice LUTs</b>	1,883
<b>Slice Registers</b>	2,045
<b>Bonded IOBs</b>	164

*Note: Utilization will scale with configuration choices (e.g., higher RNDs\_PER\_CLK). The above figures demonstrate the core's inherently lightweight nature.*

## Technical Specifications

### Interfaces

- Data Plane: AXI4-Stream
  - Separate slave (input) and master (output) interfaces.
  - Configurable data width (AXIS\_DATA\_WIDTH): 32, 64, or 128 bits.
  - Supports packetized data with TLAST signal.
- Control Plane: AXI4-Lite
  - Standard slave interface for CPU control.
  - Used to program the Key, Nonce, Associated Data Length, and Payload Length.

### Key Configuration Parameters

#### (Pre-Synthesis)

Parameter	Description
LOG2_DATA_LEN	Max length (in bits) for both Payload & Associated Data.
SHADOW_REGS	Defines the number of independent configuration register sets (1 to 4). Each set contains a complete configuration (Key, IV, Control settings), enabling instant context switching for different data streams.
MAX_TASKS	Defines the depth of the internal encryption task queue (1 to 16). This allows the software to submit multiple encryption tasks in a single burst, improving overall

	throughput and software efficiency.
RNDS_PER_CLK	Number of Ascon rounds computed per clock cycle. A higher value increases throughput at the cost of increased logic area. The optimal choice depends on the target device's speed and the area budget.
RANDOM_DELAY	Introduces random stalls to thwart side-channel attacks.
INPUT/OUTPUT_BUF_LINES	FIFO depth for clock domain crossing and rate matching.

### Typical Use Cases

- IoT Device Security: Securing sensor data and command streams between a microcontroller and a network interface.
- Automotive Ethernet: In-line encryption/authentication for data traversing an in-vehicle network.
- Secure Boot & Firmware Update: Ensuring the integrity and confidentiality of firmware images loaded from external memory.
- Military & Aerospace Communications: Protecting data links in SWaP-constrained (Size, Weight, and Power) platforms.



## Why Choose This IP Core?

1. Proven Standard: Built on the NIST-selected Ascon algorithm, guaranteeing long-term viability and security.
2. Architectural Flexibility: From a tiny, sequential (RNDS\_PER\_CLK=1) core to a high-throughput, unrolled (RNDS\_PER\_CLK=4) version, you pay only for the performance you need.
3. Designed for Integration: Industry-standard AXI interfaces drastically reduce integration time and risk.
4. Security by Design: The optional Random Delay feature provides a tangible defense against a major class of physical attacks.
5. Superior System Efficiency: The integrated task queue decouples the processor from the encryption engine. Your software can queue a stream of data and return to other tasks,

- significantly reducing CPU overhead, leading to higher overall system performance.
6. Silicon-Efficient: The hierarchical utilization report confirms a lean and optimized design, preserving precious silicon real estate for your application logic.
  7. Thoroughly Verified: 98% statement coverage, 91% branch coverage.

---

## Getting Started

We provide the complete, synthesizable RTL source code (Verilog), a comprehensive testbench with verification suite, and detailed integration documentation.

Contact us today to discuss your specific requirements, request a datasheet for your target technology node, or to license the Ascon-AD128-ENC IP core.

---

*Disclaimer: Ascon is a NIST-standardized algorithm. This IP core is a hardware implementation of that algorithm. Performance and utilization figures are representative and will vary based on target technology, synthesis tools, and configuration settings.*