Neosyn · IP Core DATASHEET · REV A (PRELIMINARY)

AES

AES-128 / 192 / 256 encryption & decryption core

○ AVAILABLE

Product code: NSN-SEC-AES · Implemented in C⏚ · Source-included · Verilog / VHDL output

The Neosyn AES core implements the Advanced Encryption Standard (FIPS 197) for 128-, 192- and 256-bit cipher keys, with hardware key expansion and an iterative round datapath tuned for FPGA resources.

The key size is selected at build time: each generated instance is a single key size (128, 192 or 256). It encrypts and decrypts 128-bit blocks. Written in readable C⏚ and shipped with source, the implementation is fully auditable — important for security review — rather than an opaque encrypted netlist.

The block core is intended to be composed with an external mode of operation (CBC, CTR, GCM, …) chosen by the integrator.

Scope. This core implements AES **encryption and decryption** for all three key sizes, verified against the FIPS-197 known-answer tests in both directions. Cipher and inverse cipher are separate compile-time datapaths (one direction per generated instance).

KEY FEATURES

- AES-128/192/256
- Encrypt & decrypt
- Compile-time key size
- Iterative round datapath
- FPGA-optimized
- FIPS-197 KAT verified
- NIST GFSbox vectors
- Readable C⏚
- Source-included

Architecture

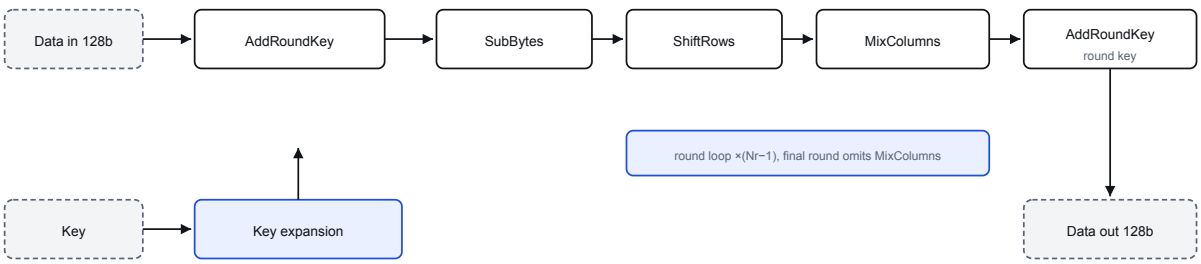

Figure 1. AES encryption round datapath. Each round applies SubBytes, ShiftRows, MixColumns and AddRoundKey; the key-expansion block supplies per-round keys. The final round omits MixColumns. The inverse cipher (decryption) mirrors this with InvSubBytes, InvShiftRows and InvMixColumns, consuming the same round keys in reverse.

1 Overview

The core implements the AES block cipher per FIPS 197 for 128-, 192- and 256-bit keys. A key-expansion block derives the round keys (11, 13 or 15 by key size) and stores them in a small round-key memory; the round datapath applies the standard transformations — SubBytes, ShiftRows, MixColumns and AddRoundKey — over the standard number of rounds (10, 12 or 14).

Decryption uses the inverse cipher (FIPS-197 §5.3): InvShiftRows, InvSubBytes, AddRoundKey and InvMixColumns, consuming the same expanded round keys in reverse order. It reuses the identical key expansion and round-key memory as the encryptor.

The organisation is iterative: one round datapath is reused across all rounds, trading throughput for area. Key expansion runs once per key, not once per block. The key size is a compile-time parameter, so each generated instance is fixed to one size.

2 Features

- **Key size.** AES-128, AES-192 and AES-256, selected at build time (one key size per generated instance).
- **Operations.** Encryption and decryption of 128-bit blocks; the direction is selected at build time.
- **Architecture.** Iterative round datapath with hardware key expansion, optimised for FPGA resources.
- **Auditable.** Readable C⏚ source — reviewable for security assurance.
- **Composable.** Block core; chain externally for CBC/CTR/GCM modes.
- **Portability.** Vendor-independent RTL generated from C⏚.

3 Functional description

Refer to Figure 1.

3.1 Key expansion

The cipher key is presented as 32-bit words, high-order word first (four, six or eight words for AES-128/192/256), and expanded into the full round-key schedule used by every AddRoundKey step. The schedule is computed once per key and held in a round-key memory.

3.2 Round datapath

Each round applies SubBytes (S-box substitution), ShiftRows, MixColumns and AddRoundKey. An initial AddRoundKey precedes the rounds, and the final round omits MixColumns, per the standard. The round count follows the key size (10, 12 or 14 rounds).

3.3 Inverse cipher (decryption)

Decryption implements the straightforward inverse cipher of FIPS-197 §5.3: an initial AddRoundKey with the last round key, then rounds of InvShiftRows, InvSubBytes, AddRoundKey and InvMixColumns, with the round keys consumed in reverse. It shares the encryptor’s key expansion and round-key memory unchanged; only the round transformations are inverted.

> **Security note.** The core implements the AES primitive only, for encryption and decryption with a 128-, 192- or 256-bit key. Mode of operation, IV/nonce management and key handling are the integrator’s responsibility. No side-channel countermeasures (masking, constant-time guarantees against power/EM analysis) are claimed.

4 Interfaces & signals

The core exposes a key input, a data input and a data output, each with a valid handshake.

GROUP	DIRECTION	DESCRIPTION
Key	in	128 / 192 / 256-bit cipher key, as 32-bit words, high-order word first
Data in	in	128-bit input block (plaintext to encrypt / ciphertext to decrypt), with valid handshake
Data out	out	128-bit output block (ciphertext / plaintext), with valid handshake
Clock / reset	in	Core clock domain and synchronous reset

Detailed signal-level pinout (per-bit widths, polarities, timing) is available on request and ships with the core.

5 Standards compliance

ITEM	DETAIL
Cipher	AES — FIPS 197
Key size	128 / 192 / 256-bit (compile-time)
Block size	128-bit
Operations	Encryption & decryption (compile-time)
Modes	Block core (chain externally for CBC/CTR/GCM)

6 Performance

PARAMETER	VALUE	NOTES
Block size	128-bit	
Throughput	Available on request	iterative; per configuration
Latency	Available on request	10 / 12 / 14 rounds by key size, plus key setup once per key
Max clock (f_{MAX})	Available on request	per target device

7 Resource utilization

Representative utilization per target FPGA family is provided on request from current synthesis reports.

TARGET FAMILY	LUTS / CELLS	REGISTERS	BLOCK RAM	F_{MAX}
Lattice ECP3	on request	on request	on request	on request
AMD/Xilinx 7-series	on request	on request	on request	on request
Intel Cyclone	on request	on request	on request	on request

Figures are supplied per project from characterised synthesis runs to avoid quoting unverified numbers.

8 Verification & validation

- Encryption and decryption are validated against the FIPS-197 known-answer tests for all three key sizes — Appendix B (AES-128) and Appendix C.2 / C.3 (AES-192 / 256). The decryption tests reuse the encryption vectors reversed, so each pair proves an encrypt/decrypt round trip. Encryption is additionally checked against the seven NIST AESAVS GFSbox vectors for AES-128.
- All are run as an automated regression gate on every build, in cycle-accurate Verilog simulation and in the C⏚ simulator, together with a bit-exact check of the expanded round keys.

9 Deliverables

- C⏚ source for the core (readable, modifiable)
- Generated synthesizable Verilog (VHDL on request)
- Self-checking testbench
- Integration guide and this datasheet
- Email integration support per the licensed tier

10 Ordering & licensing

ITEM	DETAIL
Product code	NSN-SEC-AES
License	Single-project or perpetual; full C⏚ source included
Pricing	Quoted per use (project, volume, support tier) — contact Neosyn
Support	Email integration support; custom development available
Contact	neosyn.io/contact · info@neosyn.io

11 Revision history

REV	DATE	CHANGE
A	2026	Preliminary datasheet (Neosyn / C⏚ release).