

CAST

AES

Advanced Encryption Standard Core

The AES encryption IP core implements Rijndael encoding and decoding in compliance with the NIST Advanced Encryption Standard. It processes 128-bit blocks, and is programmable for 128-, 192-, and 256-bit key lengths.

Two architectural versions are available to suit system requirements. The Standard version (AES32) is more compact, using a 32-bit datapath and requiring 44/52/60 clock cycles for each data block (128/192/256-bit cipher key, respectively). The Fast version (AES128) achieves higher throughput, using a 128-bit datapath and requiring 11/13/15 clock cycles for each data block.

Various cipher modes can be supported (ECB, CBC, OFB, CFB, CTR, CCM, GCM and LRW). The core works with a pre-expanded key, or with optional key expansion logic.

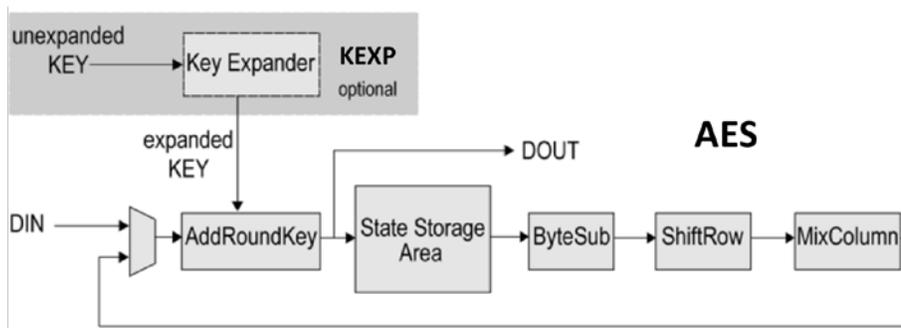
The AES core is a fully synchronous design and has been evaluated in a variety of technologies. It is available optimized for ASICs or FPGAs, with complete deliverables.

Applications

The AES can be utilized for a variety of encryption applications including:

- Protected network routers
- Electronic financial transactions
- Secure wireless communications
- Secure video surveillance systems
- Encrypted data storage

Block Diagram



Features

- Encrypts and decrypts using the AES Rijndael Block Cipher Algorithm
- Satisfies Federal Information Processing Standard (FIPS) Publication 197 from the US National Institute of Standards and Technology (NIST)
- Processes 128-bit data in 32-bit blocks
- Employs user-programmable key size of 128, 192, or 256 bits
- Smallest version supports a single block cipher mode, Electronic Codebook (ECB); these modes can be added as needed:
 - Cipher Block Chaining (CBC)
 - Cipher Feedback (CFB)
 - Output Feedback (OFB)
 - Counter (CTR)
 - Counter with CBC-MAC (CCM)
 - Galois/Counter (GCM) and
 - Liskov-Rivest-Wagner (LRW)
- Two architectural versions:
 - Standard is more compact: 32-bit data path size
Processes each 128-bit data block in 44/52/60 clock cycles for 128/192/256-bit cipher keys, respectively
 - Fast yields higher transmission rates: 128-bit data path
Processes each 128-bit block in 11/13/15 clock cycles for 128/192/256-bit cipher keys, respectively
- Works with a pre-expanded key or can integrate the optional key expansion function
- Simple, fully synchronous, reusable design
- Available as fully functional and synthesizable VHDL or Verilog, or as a netlist for popular programmable devices
- Complete deliverables include test benches, C model and test vector generator

Functional Description

An AES encryption operation transforms a 128-bit block into a block of the same size. The encryption key can be chosen among three different sizes: 128-, 192- or 256-bit. The key is expanded during cryptographic operations.

The AES algorithm consists of a series of steps repeated a number of times (rounds). The number of rounds depends on the size of the key and the data block. The intermediate cipher result is known as state.

	KSIZE = 00	KSIZE = 01	KSIZE = 10
Rounds	• 10	• 12	• 14

Number of rounds as a function of key size

Initially, the incoming data and the key are added together in the AddRoundKey module. The result is stored in the State Storage area.

The state information is then retrieved and the ByteSub, Shiftrow, MixColumn and AddRoundKey functions are performed on it in the specified order. At the end of each round, the new state is stored in the State Storage area. These operations are repeated according to the number of rounds.

The final round is anomalous as the MixColumn step is skipped. The cipher is output after the final round.

Key Expansion

The AES algorithm requires an expanded key for encryption or decryption. The CAST KEXP AES key expander core is available as an AES core option.

During encryption, the key expander can produce the expanded key on the fly while the AES core is consuming it. For decryption, though, the key must be pre-expanded and stored in an appropriate memory before being used by the AES core. This is because the core uses the expanded key backwards during decryption.

In some cases a key expander is not required. This might be the case when the key does not need to be changed (and so it can be stored in its expanded form) or when the key does not change very often (and thus it can be expanded more slowly in software).

Available Versions

The AES core is available in ECB, CFB, CBC, OFB, and CTR modes, and for different datapath sizes. The related KEXP key expander core is available as a standalone core.

Related Products

A smaller and faster AES-E core is also available for encoding-only applications.

AES in CCM and GCM modes are available as stand-alone cores.

AES-P is a run-time programmable AES core supporting ECB, CFB, CBC, OFB, and CTR modes.

Export Permits

This core is approved for export to Australia and the United States for military applications, and to all other countries for non-military applications, except for the following:

Cuba Iran Iraq Libya
North Korea Sudan Syria

It is the customer's responsibility to check with relevant authorities regarding the re-export of equipment containing the AES technology.

Support

The core as delivered is warranted against defects for ninety days from purchase. Thirty days of phone and email technical support are included, starting with the first interaction. Additional maintenance and support options are available.

Verification

The core has been verified through extensive synthesis, place and route and simulation runs. It has also been embedded in several products, and is proven in FPGA technologies.

Deliverables

The core is available in ASIC (RTL) or FPGA (netlist) forms, and includes everything required for successful implementation. The ASIC version includes

- HDL RTL source
- Sophisticated HDL Testbench (self checking)
- C Model & test vector generator
- Simulation script, vectors & expected results
- Synthesis script
- User documentation