AES-XTS

AES-XTS Storage Encrypt/Decrypt Engine

The AES-XTS encryption IP core implements hardware encryption/decryption for sector-based storage data. It uses the AES block cipher, in compliance with the NIST Advanced Encryption Standard, as a subroutine. The core processes 128 bits per cycle, and is programmable for 128- and 256-bit key lengths.

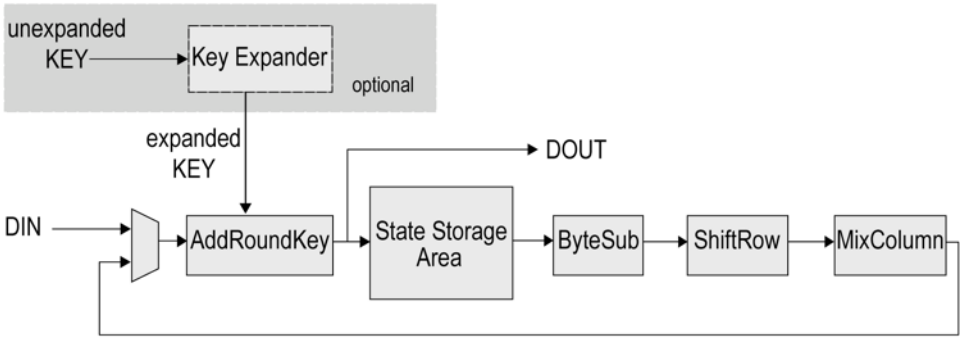
Two architectural versions are available to suit system size and throughput requirements. The **High Throughput** XTS-X is more compact and can process 128 bits/cycle independent of the key size. The **Higher Throughput** XTS-X2 can process 256 bits/cycle independent of the key size. Both versions have a 128-bit data path.

The AES-XTS cores are fully synchronous designs, have been evaluated in a variety of technologies, and are available optimized for ASICs or FPGAs.

FEATURES

- Encrypts and decrypts using the AES Rijndael Block Cipher Algorithm
- Implemented according to the IEEE P1619™/D16 standard
- NIST Certified
- Capable of processing 128 bits/cycle
- Employs user-programmable key size of 128 or 256 bits
- Two architectural versions:
  - The AES-XTS-X version is smaller and can process 128 bits/cycle for all key sizes
  - The AES-XTS-X2 version can process 256 bits/cycle for all key sizes
- Arbitrary IV length
- Works with the integrated 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

Block Diagram



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 two different sizes: 128 or 256 bits. 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 = 0	KSIZE = 1
Rounds	10	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.

XTS mode

The XTS mode of AES has been specifically designed to encrypt fixed size data where a possible threat has access to the stored data. The size of the ciphertext is the same as the plaintext.

Each data unit can be independently processed.

The last two properties allow one to transparently add encryption to a data storage system without changing the data layout of existing components.

Key Expansion

The AES algorithm requires an expanded key for encryption or decryption. The KEXP AES key expander core is included with the AES-XTS core.

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.

Implementation Results

The AES-XTS can be mapped to any Intel® FPGA device (provided sufficient silicon resources are available). The following are sample Intel results with all core I/Os assumed to be routed on-chip.

AES-XTS High Throughput (-X) Intel Results

Family	ALMs	RAM bits	Freq. (MHz)	Throughput (Gbps)
Arria 10 GX (-1)	4,674	1,843,200	170	21.76
Stratix V (-1)	4,831	1,843,200	210	26.88

AES-XTS High Throughput (-X2) Intel Results

Family	ALMs	RAM bits	Freq. (MHz)	Throughput (Gbps)
Arria 10 GX (-1)	8,526	3,547,136	150	38.4
Stratix V (-1)	8,921	3,547,136	180	46.08

The provided figures do not represent the higher speed or smaller area for the core. Please contact CAST to get characterization data for your target configuration and technology.

Related Products

AES in CBC, CCM, CFB, CTR, ECB, GCM, LRW, and OFB modes are also available as stand-alone cores.

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

Export Permits

This core implements encryption functions and as such it is subject to export control regulations. Export to your country may or may not require a special export license. Please contact CAST to determine what applies in your specific case.

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) formats, and includes everything required for successful implementation:

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