OL_AES
AES Core family
Rev 1.5

General Description

This core family implements various aspects of the AES (Advanced Encryption Standard) algorithm. Simple, fully synchronous design with low gate count.

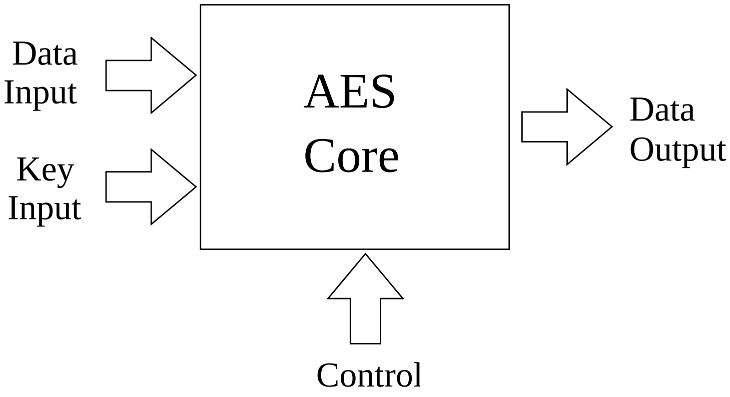
Applications

- Electronic financial transactions.
- Secure communications.
- Secure video surveillance systems.
- Encrypted data storage.

Features

- Implemented according to the FIPS 197 documentation.
- Also available in CBC, CFB and OFB modes.
- Key size of 128, 192 and 256 bits.
- Both encryption and decryption supported.
- Fully synchronous design.
- Available as fully functional and synthesizable VHDL or Verilog soft-core.
- Test benches provided.
- Xilinx and Altera netlists available.

Symbol



General Description

The OL_AES core family is a hardware implementation of various aspects of the AES algorithm as described in NIST's released documentation, suitable for a variety of applications.

The AES algorithm was selected by NIST on October 20, 2000 amongst a group of competing algorithms.
The algorithm chosen by NIST, Rijndael offers strong and secure encryption with the added flexibility of variable key block sizes.

Compared to the DES and the triple DES algorithms AES provides an even higher level of security.

An AES encryption operation consists in the transformation of a 128 bits 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.

A block diagram of the AES core is shown below.

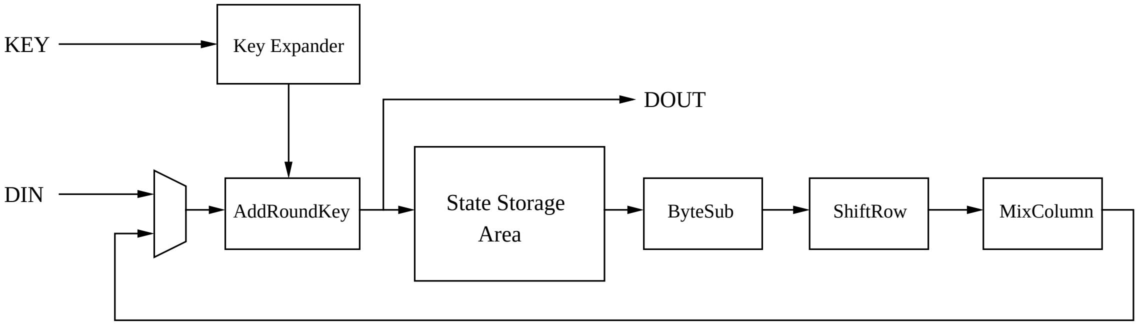


Figure 1 AES core block diagram.

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

Table 1 Number of rounds as a function of key size.

Initially, incoming data and 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 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.

After the final round the cipher is output.

Available Options

This section summarizes many of the options available when selecting a particular AES core.

Encryption and Decryption

Encryption and decryption in AES are reasonably different procedures. Consequently their hardware implementation can be quite different. Ocean Logic AES cores that support both encryption and decryption are highly optimized and share a lot of common hardware between the two functions. However, a core that supports both modes will be generally larger and slower than a core that supports encryption only.

Key Expansion

The AES algorithm requires for the key used for encryption or decryption to be expanded. Ocean Logic can provide an AES key expander together with an AES core.
During decryption the expanded key must be fed to the core backwards. A key expander core that can expand the key both forward and backwards is also available.

Core Throughput

The throughput of an Ocean Logic AES core is influenced by the width of the datapath as well as the clock frequency. Clock frequencies can be expected to range from over 250 MHz in 0.18 micron ASIC to over 100 MHz in Xilinx FPGA. The datapath width can be 32 or 128 bits. Fully pipelined designs are also available for a throughput of over 25 Gbps.

Core Performance

The table below provides an indication of the trade off between performance and area that can be expected for some of the options discussed in this section.

Core	Datapath Width	Approx. Area	Throughput bits/cycle	Throughput at 200 MHz
Encryption only core without key expander	32	4 Kgates	~2.9	~580 Mbit/s
	128	16 Kgates	~11.6	~2.32 Gbit/s
Encryption/Decryption core without key expander	32	6 Kgates	~2.9	~580 Mbit/s
	128	24 Kgates	~11.6	~2.32 Gbit/s
Key expander core	32	8 Kgates	~2.9	~580 Mbit/s
	128	32 Kgates	~11.6	~2.32 Gbit/s

Table 2 AES core family area performance trade off.

The following sections show an example of two particular cores available: OL_AES_ED and OL_KEXP_ED. OL_AES_ED is an encryption/decryption AES core with no key expander. OL_KEXP_ED is an AES key expander core. Both cores are illustrated in the 32 bit datapath version.

AES Encryption/Decryption core without Key Expander

The OL_AES_ED core supports both encryption and decryption according to the AES algorithm. The key must be provided to the core already expanded. During decryption, this core follows the

Equivalent Inverse Cipher algorithm as outlined in the AES documentation. Consequently encryption and decryption pre-expanded keys are not equivalent. Ocean Logic can provide an additional module so that encryption and decryption pre-expanded keys are the same.
The symbol of the core is shown below.

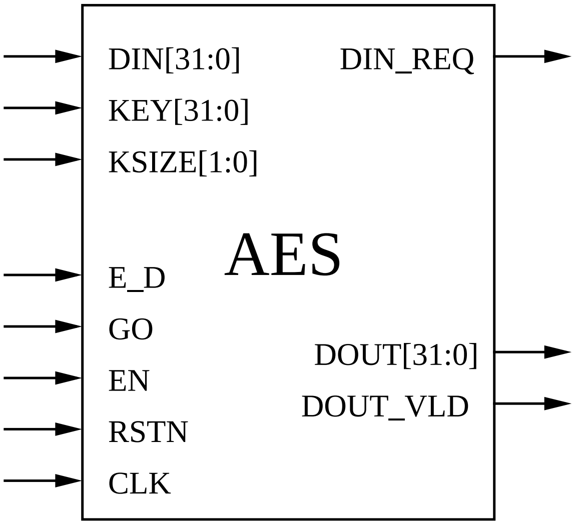


Figure 2 OL_AES_ED symbol

Pin Description

Name	Type	Description
RSTN	Input	Core reset, active low.
CLK	Input	Core clock signal.
EN	Input	Synchronous enable signal. When LOW the core ignores all its inputs and all its outputs must be ignored.
GO	Input	When HIGH, a cryptographic operation is started.
E_D	Input	Encryption is performed when LOW, decryption when HIGH.
KEY[31:0]	Input	Pre-expanded input key.
KSIZE[1:0]	Input	Input key size.
DIN[31:0]	Input	Input data.
DIN_REQ	Output	Input data request signal.
DOUT[31:0]	Output	Output data.
DOUT_VLD	Output	Output data valid.

Functional description

Rising the input on the GO port triggers the beginning of a cryptographic operation on the data DIN using the KEY as key.
The key size selection can be performed on the core by the KSIZE input. Valid values for KSIZE are “00”,”01” and “10” selecting 128, 192 or 256 bits respectively. The KSIZE inputs must not be changed while the data is processed.

The core then raises the DIN_REQ signal requesting the data block. It then starts to process the state according to the AES algorithm.
The timing diagram below shows how the data is fed to the core at the start.

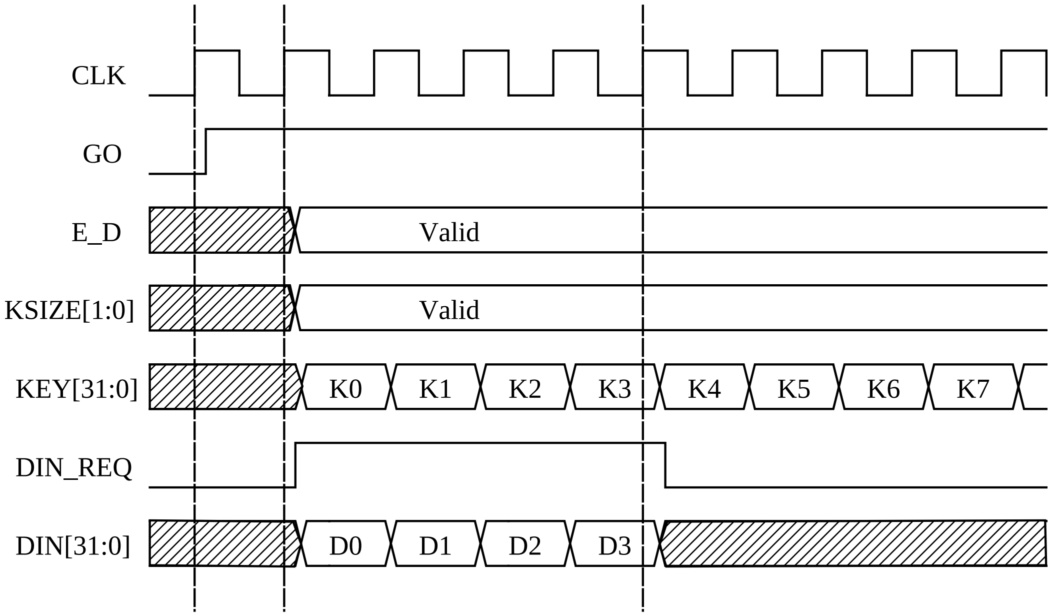


Figure 3 Key and data input at the start of encryption.

The KSIZE parameter is passed to the core after the GO signal is raised. Input of the KEY data continues for all the duration of the cryptographic operation.

Both data and key are input serially, 32 bits at the time. The diagram above shows the case where the input data is 128 bits. The ordering of the data is shown in the figure below.

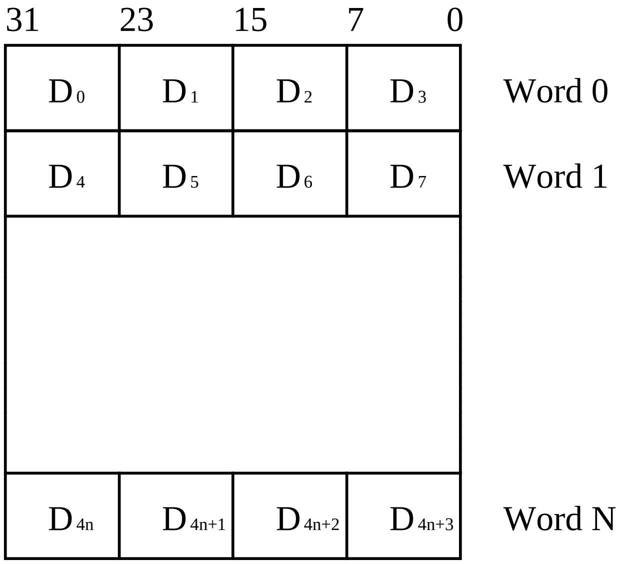


Figure 4 AES core data ordering.

When all the rounds are completed the DOUT_VLD signal is raised and the encrypted data starts to flow out. This is shown in the timing diagram below.

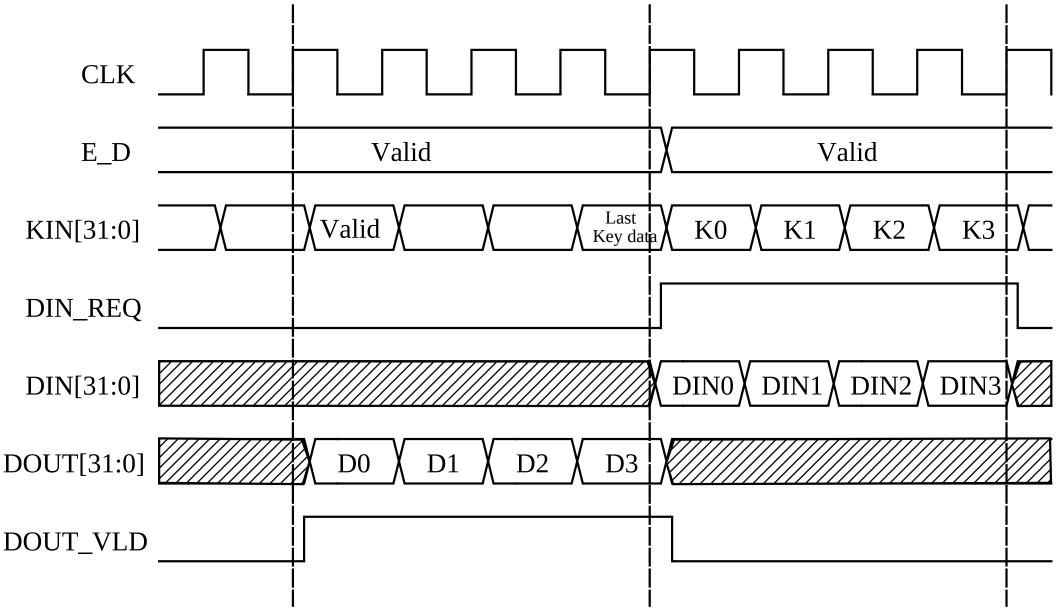


Figure 5 Cipher text from a previous operation is being output while new plaintext is input.

It is possible to start a new cryptographic operation as soon as the data from the previous one is output. The core can also switch between encryption and decryption with no gaps.
The absence of gaps allows sustaining the throughput listed in the table below.

	KSIZE = 00	KSIZE = 01	KSIZE = 10
Cycles	44	52	60

Table 3 Number of cycles as a function of key size.

It is possible to order an OL_AES_ED core that supports a slightly higher throughput (same as in **Error! Reference source not found.**) but with different IO/s.
A cryptographic operation can be aborted at any time by lowering the GO signal for at least one clock cycle.

Two other versions of this core exist: OL_AES_E and OL_AES_D. OL_AES_E supports encryption only, while OL_AES_D supports decryption only. Both cores are slightly smaller and faster than OL_AES_ED.

AES Key Expander

The OL_KEXP_ED core is a highly integrated implementation of the AES key expansion. The input key is expanded and it can be used during encryption on the fly, without the need to store the whole key in a buffer.
During decryption the core can expand the same key backwards.

The symbol of the core is shown below.

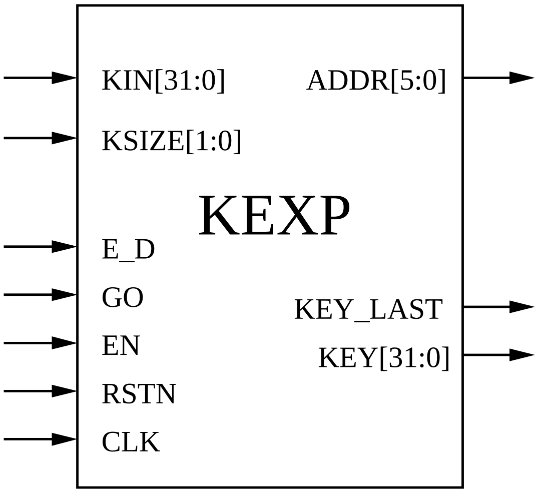


Figure 6 OL_KEXP_ED symbol

Pin Description

Name	Type	Description
RSTN	Input	Core reset, active low.
CLK	Input	Core clock signal.
EN	Input	Synchronous enable signal. When LOW the core ignores all its inputs and all its outputs must be ignored.
GO	Input	When HIGH, key expansion is started.
E_D	Input	Key is expanded for encryption or decryption.
KSIZE[1:0]	Input	Input key size.
KIN[31:0]	Input	Unexpanded key input.
KEY_REQ	Output	Input key request signal.
ADDR[5:0]	Output	Address for the key expanded data
KEY[31:0]	Output	Expanded key data.
KEY_LAST	Output	Last expanded key data.

Functional description

Rising the input on the GO port triggers the beginning of the expansion of the KEY input.
The key size selection can be performed on the core by the KSIZE input. Valid values for KSIZE are “00”,”01” and “10” selecting 128, 192 or 256 bits respectively. The KSIZE inputs must not be changed while the data is processed.

The core then raises the KEY_REQ signal requesting the key. It then starts to expand the key according to the AES algorithm.

The timing diagram below shows how the data is fed to the core at the start in the case of a 128 bit key.

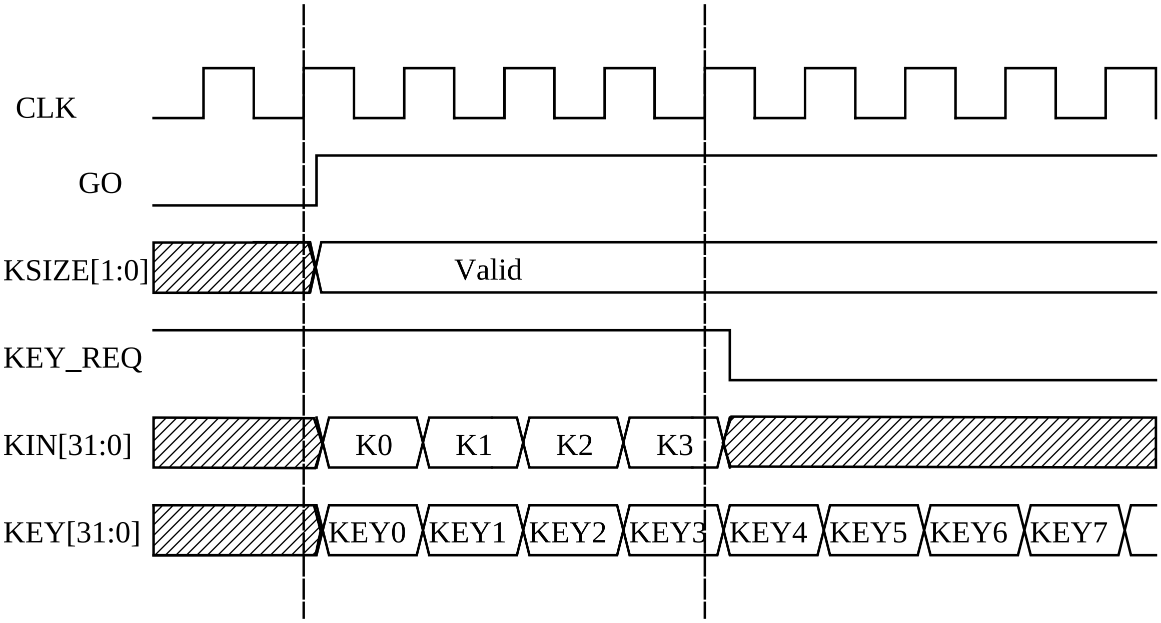


Figure 7 Key input at the start of expansion.

During the expansion process, the expanded key data is available at the output KEY.
At the end of the expansion operation, the signal KEY_LAST is raised. The core is immediately ready for another expansion operation and, in fact, the KEY_REQ signal is raised immediately after.
The diagram below illustrates this.

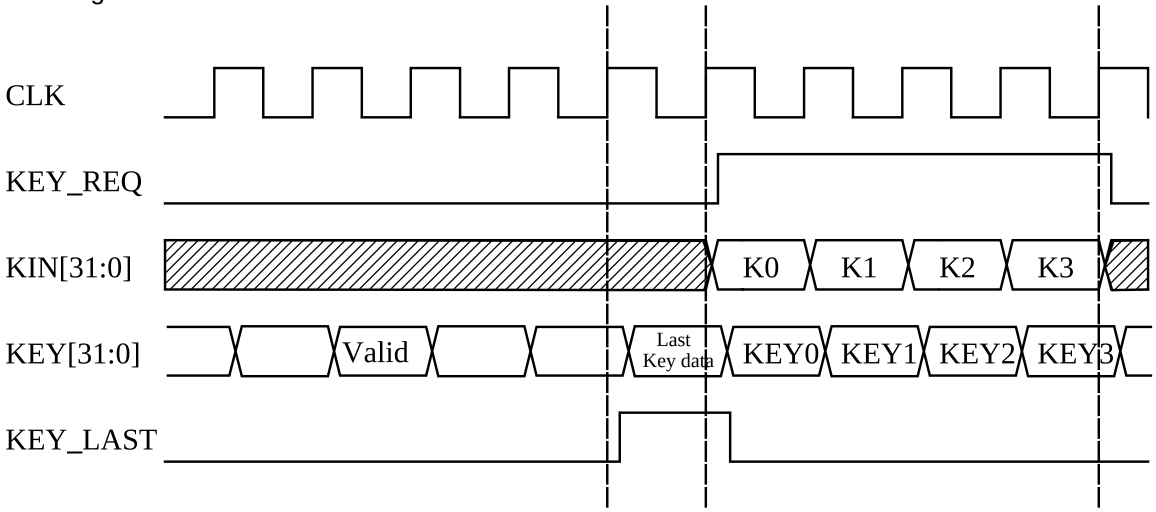


Figure 8 Last expanded key data and start of new expansion.

Any expansion operation can be aborted at any time by lowering the GO signal. Also, the core can be stalled at any time by lowering the synchronous enable signal EN.

Table 3 shows the number of cycles required for a key expansion operation as a function of key size.

OL_KEXP_ED and OL_AES_ED Performance

Performance figures of the cores in ECB mode, implemented with some particular technologies, are shown in the table below

Technology	Area	Speed	Throughput with 128 bit key
ASIC 0.13 u	5.5-7.3 Kgates	300-500 MHz	872-1454 Mbit/s
Virtex E-8	197 slices + 2 RAM Blocks	96 MHz	278.4 Mbit/s
VirtexIIP-7	85 slices + 2 RAM Blocks	206 MHz	599.2 Mbit/s

Table 4 Performance of the OL_KEXP_ED core.

Technology	Area	Speed	Throughput with 128 bit key
ASIC 0.13 u	5-8.3 Kgates	152-400 MHz	442-1163 Mbit/s
Virtex E-8	193 slices + 2 RAM Blocks	72 MHz	209.4 Mbit/s
Virtex IIP-7	133/145 slices + 2 RAM Blocks	200-216 MHz	581-628 Mbit/s

Table 5 Performance of the OL_AES_ED core.

As significantly faster but slightly larger version of OL_AES_ED is also available.

Available Versions

All the Ocean Logic AES cores are available in ECB, CFB, CBC and OFB mode, with or without key scheduler/expander. A key expander is available as stand alone core. Customizations are welcome.

Export Permits

The core is available for export to all the countries of the world with the exception of the following:

Iran	North Korea	Libya	Cuba	Sudan
Syria	Iraq			

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

Deliverables

Netlist available for most Xilinx and Altera devices.
Synthesizable VHDL or Verilog RTL.
Complete HDL testbench.

Ocean Logic Pty Ltd
PO BOX 768 - Manly NSW 1655 - Australia
Tel: +61-2-99054152 Fax: +61-2-99050921
E-Mail: info@ocean-logic.com URL : http://www.ocean-logic.com/