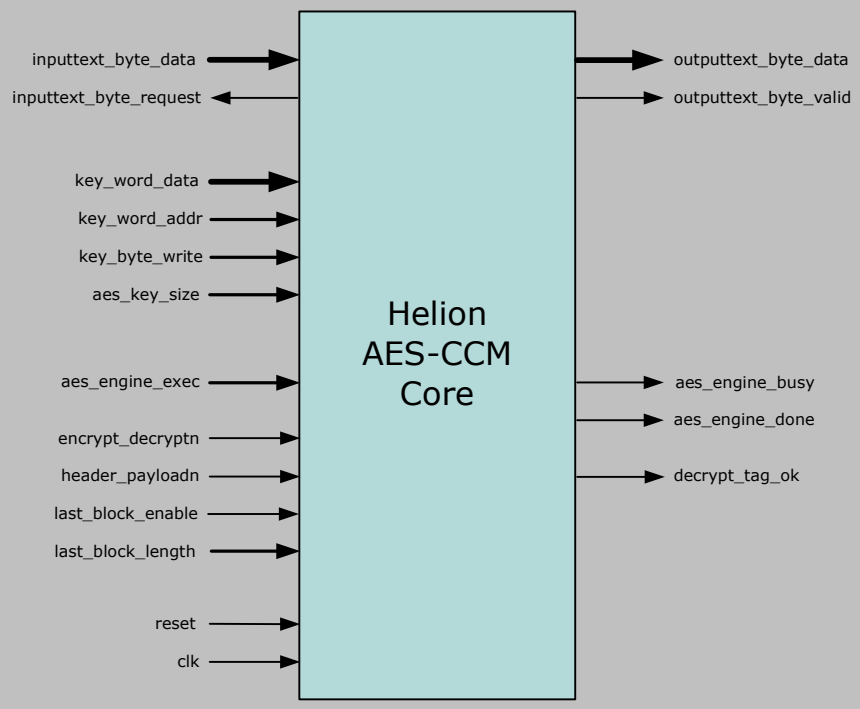
Helion Technology

PRODUCT BRIEF | **AES-CCM IP cores for FPGA**



Features

- Implements Counter with CBC-MAC (CCM) authenticated encryption mode to NIST SP800-38C
- Supports all AES key sizes (128, 192, and 256 bits) with integrated key expansion
- Performs all CCM counter management, block chaining, block masking, tag appending and checking
- Simple 8-bit data interface for easy system integration
- Suitable for use in 802.11, 802.15 and 802.16 wireless applications
- Available in multiple versions providing optimal area/performance AES-CCM solution in FPGA

Deliverables

- Target specific netlist or fully synthesisable RTL source code
- VHDL/Verilog simulation model and testbench
- User documentation

Overview

AES-CCM is an authenticated encryption block cipher mode which provides data confidentiality, integrity and origin authentication based on a single secret key, and is described formally in NIST Special Publication SP800-38C. The implementation of CCM described in this brief targets medium throughput applications, with emphasis on low resource usage and ease of use via a byte-wide interface.

The Helion AES-CCM core integrates all of the underlying functions required to implement AES in CCM mode including round-key expansion, counter management, block chaining, final block masking, and tag appending and checking features. The only external logic required is to form the Nonce block from various application specific packet header fields. Support is provided for both optional header and zero-length payload, thus supporting all three IEEE wireless standards: 802.11, 802.15 (including 802.15.4 and ZigBee™ with a CCM* variant) and 802.16.

Helion Technology Limited
Ash House, Breckenwood Road,
Fulbourn, Cambridge CB21 5DQ, England

Functional Description

Background
The Helion AES-CCM core internally performs two distinct AES operations; AES-CTR mode to provide data encryption or decryption, and AES-CBC-MAC mode to provide message authentication. Both AES operations use the same key. Data to be processed will be a series of atomic messages, each of which has a unique Nonce/IV and will have its own authentication tag appended. Within each message, the data may be classed as "header" (which is not encrypted, but is authenticated) or "payload" (which is both encrypted and authenticated). This maps well into typical network packet formats, where the header must be unchanged through the encryption process, but needs to be protected by the authentication. Messages with no header or no payload are also supported.

Getting started
The user drives the core by issuing a sequence of commands via the *aes_engine_exec* port, and the core's status is indicated at each stage by the *aes_engine_busy* and *aes_engine_done* flags. Before any data processing can occur, a master AES key must be loaded into the core using the byte-writable 32-bit key interface. Key pre-processing must then be initiated by the user issuing an EXEC_KEY command to the core, and indicating the AES key size to be used (via *aes_key_size*).

For each message
Before the start of each message to be encrypted or decrypted, the Nonce/IV must also be loaded by issuing an EXEC_INIT command to the core. The 128-bit Nonce/IV is then transferred into the core using the byte-wide data input interface (*inputtext_byte_data*).

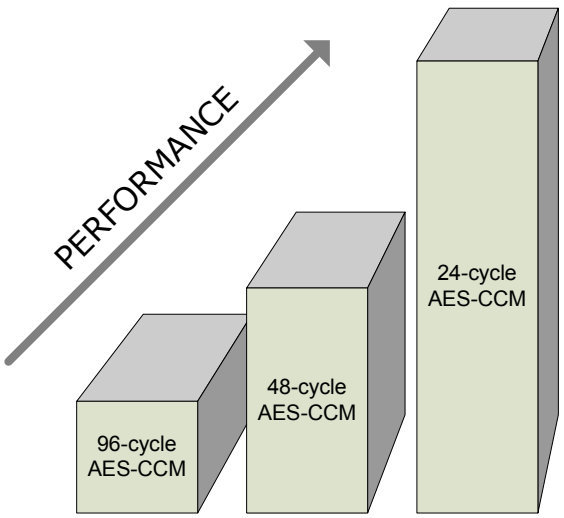
Processing the message data
Once the core has been initialised with a valid Key and Nonce/IV, the Message data can be processed. This is performed using multiple 128-bit block encrypt/decrypt operations, initiated by issuing one or more EXEC_DATA commands to the core. Control inputs are used to indicate the direction (*encrypt_decryptn*) and data type (*header_payloadn*) for each operation.

Input Data blocks enter the core via the byte-wide data input interface (*inputtext_byte_data*), and the resulting output data blocks emerge from the core via the byte-wide data output interface (*outputtext_byte_data*). Simple handshaking controls indicate when input data bytes are required or when output data bytes are valid.

At the end of the message
The last header or payload block may be less than the full 128 bits, and so its presence and length in bytes are indicated to the core using the "last_block" control inputs. Once the last message block has been encrypted/ decrypted, the tag will either be appended to the output data (encrypt direction), or will be checked against the received tag (decrypt direction) and the tag check output flag (*decrypt_tag_ok*) will be driven accordingly. The process can then be repeated for the next message, by loading a new key (if required) or the next Nonce/IV.

Core Choice

Helion always offer a range of solutions so that the throughput requirements of any application can be closely matched with optimum area efficiency. In this case, Helion have three levels of performance available; we name them to reflect the nominal number of clock cycles taken to process each 16-byte data block. NOTE. The actual number of cycles taken by the core to process this block varies with exact core choice and the keysize selected (see table on next page).



The smallest member of the family is the **"96-cycle" AES-CCM** core which takes a minimum 96-clock cycles to encrypt or decrypt each 16-byte data block using a 128-bit key.

For higher throughputs, the **"48-cycle" AES-CCM** core offers twice the performance of the 96-cycle core while using less than twice its logic area. It takes a minimum 48-clock cycles to encrypt or decrypt each 16-byte data block using a 128-bit key.

The highest performance member of the family is called the **"24-cycle" AES-CCM** core, which offers nominally twice the performance of the 48-cycle core while using less than twice its logic area. It takes a nominal 24-clock cycles to encrypt or decrypt each 16-byte data block using a 128-bit key.

Each version of the core is available with support for one, two or all three AES key sizes (128, 192 and 256-bit).

HELION

Core Throughput

The tables below show the number of cycles and the maximum data throughput as a function of core clock frequency, for each version of the AES-CCM core, for each supported key size.

version	AES-CCM 24-cycle		AES-CCM 48-cycle		AES-CCM 96-cycle	
key option	128 only	Allsizes 128/192/256	128 only	Allsizes 128/192/256	128 only	Allsizes 128/192/256
clock cycles used per 16-byte block	23	28/28/31	48	48/56/64	96	96/112/128
data throughput (Mbps per MHz)	5.5	4.5/4.5/4.1	2.6	2.6/2.2/2.0	1.3	1.3/1.1/1.0

For any specific application, a core version can be chosen that will achieve the required throughput, with an appropriate and achievable core clock frequency, not forgetting to take into account the inevitable per-message overheads which may be incurred (e.g Key setup, Nonce/IV setup and message finalisation). Note that the two types of 24-cycle core are individually optimised to minimise logic area, and so have differing cycle counts for 128-bit keys. Other options are available if the listed performance above is not appropriate.

Logic Utilisation and Performance

Helion cores are fully characterised in all supported FPGA types (please see overleaf for supported FPGA technologies), with specific results available for each device type and speed grade. This yields a huge amount of data, so we don't include it in this Product Brief. Please contact Helion with your vendor requirements, and we will be delighted to send the appropriate information.

For general guidance however, the typical maximum achievable clock rates in the latest fast FPGA silicon might be ~350MHz in a mid speed grade part (depending on the exact version and device), whilst in lower cost FPGA devices this figure may be closer to ~200MHz. These figures can be used as a starting point to determine which version of the core could be suitable for your requirements. A selection of the most popular combinations are also shown on our AES-CCM core web pages, at http://www.heliontech.com/aes_ccm.htm.

Looking for Higher Rates?

The AES-CCM algorithm does have certain inherent limitations in terms of the throughput it can support, due mainly to the use of block-by-block feedback in its CBC-MAC function. This is a limitation of the algorithm, but Helion does have faster AES-CCM cores available than are presented here, should this be an absolute requirement.

Alternatively, if you have flexibility in terms of the encryption mode you are using, it may be worth looking at AES-GCM as an alternative, since this offers similar functionality without the same throughput limitations. Please take a look at our AES-GCM webpage at http://www.heliontech.com/aes_gcm.htm, or contact Helion for more information on faster AES-CCM or AES-GCM solutions.

Ordering Information

Before ordering it is necessary to decide which of our family of AES-CCM cores will best fit your application. First decide between the **96-cycle**, **48-cycle**, and **24-cycle** cores according to the data throughput required and logic resources available. Then determine which AES key sizes you would like to support as well as any other special requirements your application may have.

If some of these choices are unclear, or you would just like to go over the options available, we are always happy to discuss the alternatives and help select the best solution for your application.

FPGA Technology Support

Helion has a long history in high-end FPGA design, and takes a great deal of care when implementing IP cores. As a result, these cores have been designed from the ground up to be highly optimal for each individual FPGA technology - they are not simply based on a synthesised generic RTL ASIC design. Helion cores always make use of the architectural features available in each FPGA technology to achieve the highest performance combined with the most efficient logic resource utilisation.

Helion is an accredited IP partner with **Altera, Lattice, Microsemi (Actel)** and **Xilinx**, and supports all current and many legacy FPGA technologies from these vendors. Please feel free to contact Helion if your FPGA technology of choice is not listed here.

ALTERA | LATTICE SEMICONDUCTOR | Microsemi | XILINX ALL PROGRAMMABLE ALLIANCE PROGRAM CERTIFIED MEMBER

About Helion

Founded in 1992, Helion is a long established British company based in Cambridge, England. We offer a range of product-proven Data Security and Lossless Compression IP cores, backed by a team of highly experienced engineers, proudly developing and supporting a world-class portfolio.

Our aim is to offer our customers…

Innovation
Helion works hard to anticipate, understand and then deliver great solutions for its customers. As an example, Helion offered the world's first commercial AES core back in 2001, even before the industry had fully adopted the algorithm. This process continues unabated today, with new products in development that will lead the field.

High Performance
Helion IP is specially designed and optimised for each target technology. This means lots of work for us, but this approach yields amazing results for our customers. We always aim for the best in class performance and lowest utilisation in any given ASIC or FPGA target.

High Quality
IP should be problem free, so we always go the extra mile to ensure a smooth and trouble free integration phase for our products. We realise that our customers are putting their faith in us, and want to repay that with an outstandingly easy deployment.

Ease of Use
Helion engineers have many years of real product development experience, and so our IP is designed to be used in realistic situations. It is flexible and well thought through - the result being that it is simple to drop into your system.

See how we achieve all this by visiting our Clients page at http://www.heliontech.com/clients.htm

More Information

For more detailed information on this or any of our other products and services, please contact Helion and we will be pleased to discuss how we can assist with your individual requirements.

Helion Technology Limited
Ash House, Breckenwood Road,
Fulbourn, Cambridge CB21 5DQ, England

tel: +44 (0)1223 500 924 email: info@heliontech.com
fax: +44 (0)1223 500 923 web: www.heliontech.com