AES GCM 100G for OTN Core, Xilinx Edition

November 7, 2014 Product Specification

Core Facts

Provided with Core	
Documentation	User Manual
Design File Formats	VHDL or Netlist
Verification	Test Bench
Instantiation templates	VHDL
Simulation Tool Used	
ModelSim	
Support	
Support provided by Algotronix	

Algotronix®
130-10 Calton Road
Edinburgh, Scotland
United Kingdom, EH8 8JQ
Phone: +44 131 556 9242
E-mail: cores@algotronix.com
URL: www.algotronix.com

Features

- Implementation of AES-GCM with 96 bit IV and 128 bit or 256 bit keys.
- Suitable for OTN and applications requiring AES-GCM with fixed packet sizes and relatively infrequent key changes
- Compatible with all modern Xilinx FPGA families
- Configurable number of compute units each containing fully pipelined AES implementation for performance between 20 and 400Gbit/sec
- Pass through functionality for packets not requiring encryption
- Comprehensive self-checking testbench
- Supplied as VHDL source code to allow security review

Applications

- Wired, Optical and Wireless Networking
- Network Test Equipment
- OTN

General Description

The Algotronix AES-GCM Core implements the Galois Counter Mode (GCM) of operation of the Advanced Encryption Standard (AES) algorithm. The AES-GCM mode of operation of AES was originally described in a proposal by Cisco to the National Institute of Standards and Technology (NIST) and later published as NIST Special Publication SP800-38D making it an officially endorsed mode of operation of the AES cipher. The AES-GCM-100G-OTN version of the product is optimised for applications, such as OTN, where all packets are relatively large, all the same size and keys change relatively infrequently. These characteristics allow an implementation with extreme levels of pipelining and parallelism.

AES-GCM is significantly more complex than the simple modes of AES such as ECB and CBC specified in NIST special publication SP800-38A because unlike simple modes of the cipher which provide only confidentiality, GCM provides both confidentiality and authentication. Authentication is the ability to detect tampering with the encrypted message as it passes between the sender and receiver and in most applications is essential for security. GCM mode is based on the counter (CTR) mode of AES which can be parallelized to achieve very high throughput. Thus GCM is the most suitable of the standard modes of AES to provide both authentication and confidentiality for very high speed networks.

The AES-GCM-100G-OTN core is an performance-optimised version of the standard AES-GCM-10G core with multiple compute units operating in parallel.   Area savings are obtained by having the user precalculate the key schedule and information to simplify the hash computation in software and load these pre-computed values into the core. This pre-computation approach makes sense in applications such as OTN where the key changes relatively infrequently. The core is further simplified by specifying a fixed length of packets as a configuration parameter (rather than allowing packet length to vary as in the MACSEC and IPSEC scenario) and limiting packet lengths to be an exact multiple of AES blocks (16 bytes or 128 bits) as will naturally be the case for OTN frames. Algotronix can customise the core on request to lift some or all of these restrictions.

The AES-GCM-100G-OTN core operates with a fixed 96 bit IV and can be configured to use a 128 bit or 256 bit key. For simulation a self-checking configuration of the core checks the response of the synthesisable code against a behavioral model. The user can instantiate this self-checking version of the core in their own simulation in order to check its response to stimulation from their design and track down any problems in the way the core is being driven more quickly. The core testbench uses the self checking version of the product and stimulates it with a user specified number of randomly generated vectors.

The GCM-100G-OTN core is supplied as VHDL source code and can be configured using a number of VHDL generic parameters to trade off performance against area. The core is an easy to use fully synchronous design with a single clock. The core has been designed for efficiency in modern FPGAs and makes full use of FPGA dual port memory blocks.

Implementation Statistics

The core can target all recent families of Xilinx chips and has a variety of configuration options, Algotronix will run the core through FPGA vendor design tools and provide figures for devices and options not listed in the tables below on request. The design was run with a clock constraints of 391MHz with 2 compute units and 195.5MHz with 4 compute units as required to deliver 100Gbit/sec.. A 256 bit key requires more pipeline stages and hence more resources than are required with the 128 bit key in these examples.

Table 1: Example implementation statistics for AES-GCM-100G-OTN, 128 bit Key, 'Push Button' flow with Fmax specified by clock constraint. AES SBoxes implemented in LUTs. Two compute units with Inner loop pipelining.

Family	Example Device	Fmax (MHz)	FF	LUT	Memory LUT	GCLK	BRAM (18K)	Throughput (GBit/sec)	Design Tools
Virtex-7	XC7VX690tffg1930-3	391	14480	24657	261	1	6.5	100Gbit//sec	Vivado 2014.4

Table 2: Example implementation statistics for AES-GCM-100G-OTN, 128 bit Key, 'Push Button' flow with Fmax specified by clock constraint. AES SBoxes implemented in LUTs. Four compute units without Inner loop pipelining.

Family	Example Device	Fmax (MHz)	FF	LUT	Memory LUT	GCLK	BRAM (18K)	Throughput (GBit/sec)	Design Tools
Virtex-7	XC7VX690tffg1930 -3	195.5	13735	24393	261	1	170.5	100Gbit//sec	Vivado 2014.4

Functional Description

The main functional blocks are as shown in Figure 1, and explained below. The various I/O signals shown on the diagram are defined in Table 2.

AES-GCM Compute Units

The design includes one or more compute units operating in parallel, the number of compute units is set by a configuration variable and is chosen according to the required throughput and achievable clock frequency on the traget FPGA. Each AES-GCM compute unit consists of an AES unit to provide encryption and a GF-HASH unit for authentication. The AES core has a fully unrolled pipeline and starts an encryption on every clock cycle. The Galois Field multiplier required by the GHASH algorithm used for authentication is also pipelined to match the throughput of the AES unit.

Partial Hash Merge Unit

A separate unit merges the partial hash data from the multiple compute units to form the complete Integrity Check Value (ICV) for the whole data stream. With the AES-GCM-100G-OTN core, unlike the standard AES-GCM-10G core, it is the responsibility of user circuitry to calculate authentication success for the receive channel by comparing the computed ICV and an ICV sent with the received data. In the OTN application there is no standard specifying which field in the received data will be used for the expected ICV and it may be convenient to send the ICV for the current frame in the header of the subsequent frame. Therefore separating the authentication success computation from the AES-GCM unit provides more flexibility. User circuitry must also ensure that if authentication is not successful the corresponding frame is discarded.

GCM Control and Precalculated Data Memory

This block contains a state machine that generates various control signals for the rest of the design. It also contains a memory to hold the precalculated data (including the AES key schedule) for each compute unit. The memory has an active bank controlling the datapath and an inactive bank into which the processor writes new data. When the key needs to be changed a new set of data can be written to the inactive bank by the processor without distrubing the datapath which is running from the data in the active bank. The 'activate_key' signal is then pulsed high two clock cycles before the 'first' signal starts a new frame, swapping the key over without interrupting processing or requiring an additional gap between frames.

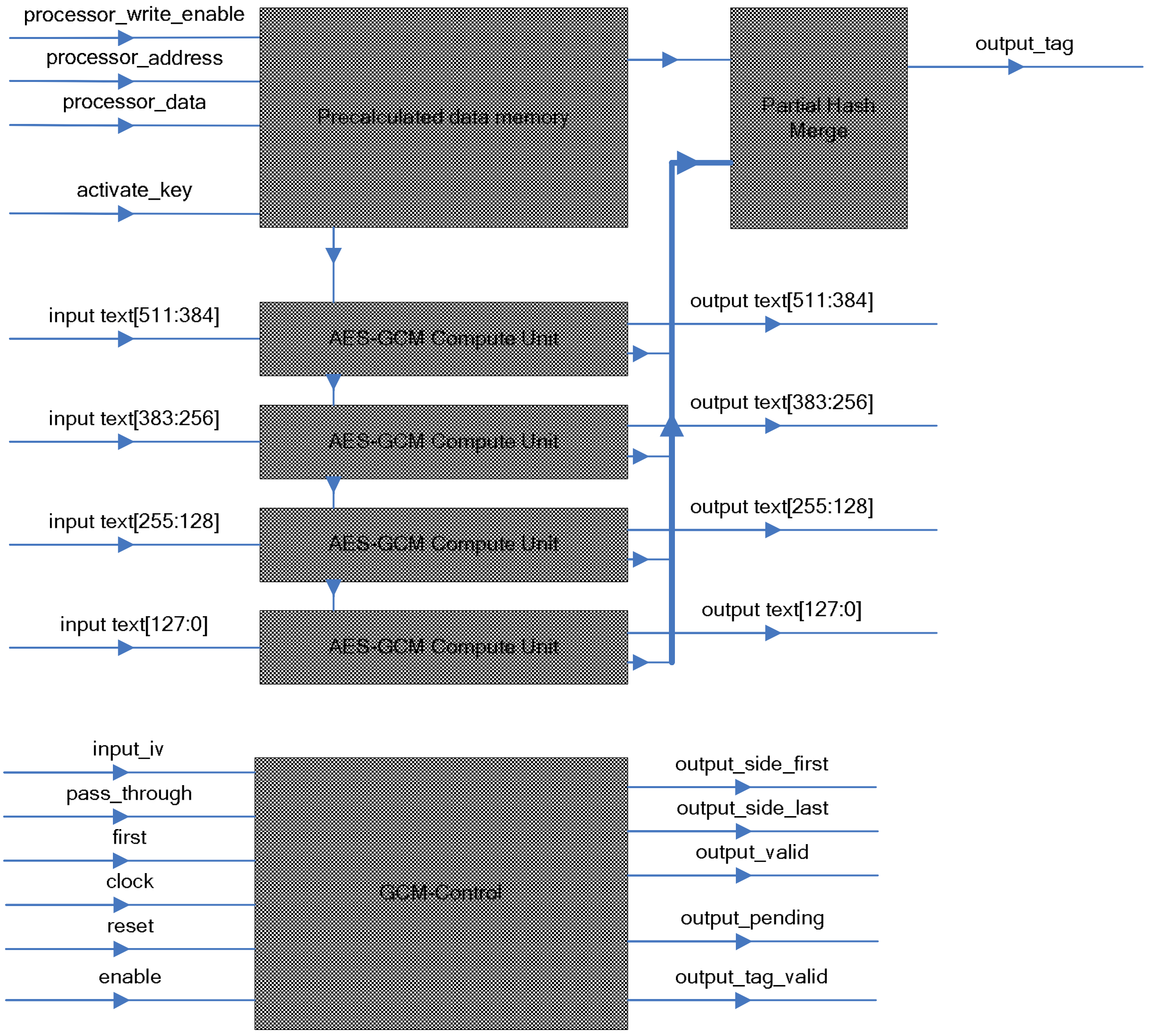

Figure 1, AES-GCM-100G Core Block Diagram (configured with 4 compute units)

Description of Operation

In the description below ‘block’ is used to refer to an AES block of 128 bits of data, ‘packet’ is used to refer to an AES-GCM processing unit consisting of IV, AAD and plaintext or ciphertext (depending on whether it is an encrypt or decrypt operation).

The AES-GCM-100G-OTN core implements the AES portion of the algorithm using one or more compute units each of which has a 128 bit wide datapath and a 10 stage pipeline for 128 bit keys (or 14 stage pipeline for 256 bit keys which require more rounds of AES processing). This AES unit can encrypt or decrypt a 128 bit block of text with a latency of 10 clock cycles (14 clock cycles for 256 bit keys) and a throughput of one block every clock cycle. This timing follows directly from the iterative nature of the AES algorithm which requires ten cycles of inner loop processing when the key length is 128 bits. The implementation of the GHASH algorithm used in the calculation of the Integrity Check Value (ICV) (or 'tag') is based on the Kurutsaba algorithm. There is a pipelined partial hash computation within each compute unit to offer the same throughput as the AES implementation within the unit. The separate partial hash computations from each compute unit are then combined together and encrypted in the partial hash merge unit to form the ICV.

The core can be configured with 1, 2, 4, 8 or 16 compute units. With a clock frequency of 195.5MHz this corresponds to throughputs of 25, 50, 100, 200 and 400GBit/sec. With a clock frequency of 391MHz this corresponds to 50, 100, 200, 400 and 800Gbit/sec. At the time of writing the 400 and 800Gbit options are unlikely to be possible in practice due to difficulty of obtaining timing closure and acceptable power consumption: 391MHz is readily achievable on the most modern FPGA chips with a stand alone AES-GCM-100G-OTN IP core with 1 or 2 compute units but timing closure may be challenging when the IP core is combined with the rest of the user design. As new FPGAs become available 391.5MHz and two compute units should become an attractive option at 100Gbit/sec because of the signficant area savings over the 4 compute unit with 195.5MHz clock configuration.

The core relies on user software to precompute information used in the AES GCM algorithm and load that information into the precalculated data memory within the core through a processor interface separate from the data interface which transfers packets. The software required to compute the information is relatively simple and there is behavioral VHDL code in the testbench to perform this function that can serve as a starting point. The processor interface is active even when the enable signal to the data path is low. Writing information to the precalculated data memory to set up a new key does not disturb normal operation of the core processing packets using the information from the previous key. The user activates new keys by pulsing the transmit_activate_key (or receive_activate_key) signal high two clock cycles before the first packet to be processed with the new key enters the core. The precalculated information includes the key schedule for AES encryption and the value of H used in the GHASH algorithm.

Processing a packet of GCM data is initiated by pulsing the 'first' signal high. The first block of text, the pass_through control signal and input_iv must be valid at this time. The pass_through and iv signals are latched internally and subsequent changes during period while the core is processing data will have no effect. There may be a gap between the previous packet finishing and the 'first' signal for the next packet being activated but this is not required: it is possible for a new packet to start on the clock cycle after the previous one finishes. A signal to mark the end of packets is not required in the OTN application because the packet size is fixed and specified as a compilation parameter of the core. Data is input every clock cycle until the specified size of the Additional Authenticated Data (AAD) and text regions is reached. The AAD is authenticated but not encrypted and always comes at the start of the packet: this is specified in the definition of the AES-GCM algorithm. AAD is useful for packet header information which must be visible to network hardware without decrypting the packet.

Ideally, the user circuit should synchronize to the core using the output_valid signals rather than by assuming a set number of clock cycles between input and output. Future updates to the core may have slightly different latency as a consequence of performance optimisations or new capabilities.

There are many possible timing scenarios depending the various configuration parameters for the core. For reasons of space and convenience detailed timing charts for the different scenarios are not provided in the product description but are produced on request using the testbench and supplied separately.

Compilation Options

The core can be configured easily using a set of VHDL generic parameters. Normally, it is unnecessary for users to modify the design source code although the code is supplied and they are free to do so if they wish. Algotronix can also customise the core as a service for users with requirements which are not met by the standard product.

- **crypt_size -** specifes 128 or 256 bit encryption. Specifying 256 bit encryption configures the core with 14 rather than 10 pipeline stages in each AES encryptor which significantly increases area.
- **number_of_compute_units -** number of compute units operating in parallel. The design currently supports configurations with 1, 2, 4, 8 or 16 compute units.
- **cipher_function -** specifies whether the core should be configured as encrypt only, decrypt only or switchable between encrypt and decrypt.
- **aad_length_in_blocks -** number of 128 bit blocks (16 byte) blocks of Additional Authenticated Data (AAD) in each packet. The AAD precedes the text to be encrypted/decrypted and is authenticated but not encrypted. AAD is useful for header information which must be immediately accessible to intermediate nodes on the communications channel but protected against tampering.
- **text_length_in_blocks -** number of 128 bit blocks (16 byte) blocks of text (plaintext or ciphertext) in each packet. The text is encrypted for privacy and authenticated.
- **implement_sboxes_in_ram** - specifies whether the AES SBoxes are implemented using FPGA RAM blocks or using FPGA LUT resources. This choice will have a major impact on the balance of resources in the design and a significant effect on achievable clock frequency.
- **use_inner_loop_pipelining** - specifies that an additional pipeline register is used within each AES round processing unit. This doubles the latency through the AES pipeline but allows a higher clock frequency to be used. This switch can be useful to reach 100Gbit/sec performance with only two compute units.
- **target_device** – This edition of the core can target all families of Xilinx FPGAs. Alternative editions target other FPGA manufacturers as well as a 'platinum' edition which can target all leading FPGAs or ASIC. Only relatively modern, high performance FPGA families will have sufficient resources and performance to support 100Gbit data rates.

Core I/O Signals

Descriptions of all I/O signals are provided in Table 2. In most cases these signals will connect to signals in the surrounding user design, not directly to I/O pins on the FPGA.

Differences in interface from AES_GCM_10G

The I/O signals for the AES-GCM-100G-OTN core differ from those on the AES-GCM-10G core as follows:

1. In configurations with multiple compute units multiple 128 bit blocks of data enter the core every clock cycle.
2. The core assumes the size of the AAD and text will always be an exact number of AES blocks, as is the case for OTN. As a result the '*_width' inputs and outputs used to specify the number of bytes in the last block of AAD or text are not required.
3. The number of blocks of AAD and text is known in advance and specified as a configuration parameter. Therefore there is no need for the '*_kind' fields which specify whether the current input or output text is AAD or text.
4. The end of the packet can be determined by counting input or output blocks since all packets are a fixed size. Therefore there is no need for the '*_final' signals on the input and output side which mark the final block of text.
5. There are no signals associated with loading keys since the key information is provided in advance through the processor interface.
6. The 'start' signal on AES-GCM-10G input side occurs one clock before the data and is used to trigger calculation of the keyschedule is replaced with a 'first' signal which goes high concurrently with the first data word.
7. A processor interface and install_key signal are added to deal with loading and enabling key dependent data.

Signal	Signal Direction	Description
clock	input	Clock – active on rising edge.
reset	input	Reset – active high. For Xilinx FPGA implementation, unless security considerations mandate an asynchronous reset, it is recommended to specify that the reset signal is implemented synchronously. A configuration constant USE_ASYNCHRONOUS_RESET to specify the style of reset is provided in aes_package.vhd. On Xilinx a synchronous reset can result in reduced area and improved performance.
enable	input	Global enable signal which controls the whole design except for memory accesses through the processor interface. If not required it should be tied to '1' to allow synthesis to optimise out the enable circuitry. This may reduce area and routing congestion and therefore allow a higher clock frequency.
processor_write_ enable	input	Indicates the data on the data_from_processor input should be written to the internal precalculated data memory at the address specified in the processor_address signal
processor_address[8:0]	input	Address of data to be written in the inactive bank of precalculated data memory. The processor cannot access the memory bank which is currently controlling the datapath. The address is specified wide enough to deal with the 16 compute unit case.
data_from_ processor[31:0]	input	Data to be written to the precalculated data memory. A 32 bit data bus is supported.
activate_key	input	Pulsed high two clock cycles before the first signal indicates the first block of a new packet to swap over the active and inactive banks of precalculated data memory. The processor must have written all the required data before this signal is asserted.
first	input	Pulsed high on the first block of input_text for a new packet to be processed. The pass_through and input_iv inputs are sampled and these parameters are fixed for the next block of operations. There can be a gap between the last block of text of the previous packet entering the core and this signal pulsing high to mark the start of the new packet.
pass_through	input	Sampled when first = '1', specifies that the packet is to be passed through the core without any security processing. This is a convenience feature so that the user does not have to provide a separate path around the GCM.
input_iv [95:0]	input	Data input. A complete 96 bit AES IV is transferred in a single clock cycle when first = '1'.
input_text [number_of_compute_units - 1 : 0] [127:0]	input	Data input. An array of complete 128 bit AES block is transferred on each clock cycle, one block for each of the compute units.
output_ text_valid	output	Valid flag – high when output_text is valid for all compute units.
output_text_ first	output	High on the first block of output text for a packet.
output_text [number_of_compute_units - 1 : 0] [127 : 0]	output	Data output. An array of complete 128 bit AES block is transferred on each clock cycle, one block for each of the compute units.
output_tag_ valid	output	High when output_tag is valid and contains the ICV computed from the input data. This is several clock cycles after the final word of output_text for the packet.

output_tag [127 : 0]	output	Output bus for the ICV computed from the input data (AAD and text).
output_pending	output	Indicates that the core is currently processing input data and there will be further output_valid cycles.

Table 2: Core I/O Signals.

Programming Interface

The core provides memory mapped configuration resources with a conventional data bus, address bus and write_enable signal. The data bus is 32 bits wide to suit common embedded processors. Reading back of memory locations is not currently supported since it would be relatively expensive in FPGA resources. Algotronix can provide a readback capability on request.

The memory map is specified to support the configuration with the maximum number of resources (i.e. 256 bit keys and 16 compute units). All configurations have the same memory map, writing to locations which are not required by the specified configuration has no effect.

All resources within the configuration memory are 128 bit AES blocks mapped into four 32 bit words with address[1:0] selecting a 32 bit word within the block. The more significant words have the lower addresses i.e. bits [127:96] of the AES block are mapped into the word with address[1:0] = B"00" and bits [31:0] to address[1:0] = B"11". Bits [8:2] of the address bus are used to select one of 71 AES blocks.

Memory locations 0 through 13 are used to store round keys for the AES units, 14 round keys are required for a 256 bit key. The last 4 locations are provided in the memory map but not used by the hardware for a 128 bit key.

Memory map (in terms of 128 bit AES blocks).	
Address (decimal)	Function
0..13	AES Round keys (shared by all compute units)
14	Precomputed common GHASH information supplied to all units
15..19	Precomputed GHASH information for Compute Unit 0 (4 blocks)
20..23	Precomputed GHASH information for Compute Unit 1
...	...
65..69	Precomputed GHASH information for Compute Unit 15
70	Precomputed GHASH information for Hash Merge Unit

Table 3: Memory Map

High level behavioral VHDL code to compute the required AES round keys and precomputed GHASH information for loading into the memory is provided with the core as part of the testbench and can easily be translated to C. The specifics of the pre-computed GHASH information are not disclosed here but are available under Non-Disclosure Agreement.

Information written to the configuration memory goes into a separate memory bank from that used to control the datapath and writing new informantion does not disturb datapath operation. Complete configuration information for the channel must be written before asserting 'activate_key' to swap the new information into the active memory bank controlling the hardware.

The compute unit consists of a series of pipeline stages. When a new key is activated and a new packet started as the first data for the new packet makes its way through the pipeline the earlier pipeline stages will be operating with the new key but the later pipeline stages will still be processing the previous packet and using the previous key. There is a period of 18 clock cycles for a 256 bit key (or 14 clock cycles for a 128 bit key) immediately following the activate_key pulse during which this transition is occurring and the user must take additional care when writing to the control memory. A write to a memory stage which is not yet swapped over to the new key will leave the core incorrectly configured resulting in errors. Two simple strategies are available to prevent this happening:
a. Wait for 18 clocks after the activate_key pulse before starting to write new information so that the transition is complete for all stages.
b. Write the new information in order starting with stage 0 and working in sequence to the last stage. With a 32 bit databus the processor will take at least four write cycles for each 128 bit stage (whereas the hardware will flip over a complete stage every clock) so the processor will always be working on stages that the hardware has already dealt with.

This is an important consideration in simulation where it is easy for a testbench to write key information every clock. However, since the processor cores found on FPGAs usually require multiple clock cycles to transfer a word of information to the core configuration of the core in hardware from software running on an embedded processor is unlikely to be fast enough to trigger the problem.

Verification Methods

The testbench includes a self-checking configuration of the top level entity in the VHDL design which uses the behavioral model of the AES-GCM algorithm to check the results from the synthesisable implementation code. This is implemented using the VHDL facility to provide multiple architecture definitions for a particular entity: the top level entity in the design has a self_checking and a synthesis architecture defined. As shown in Figure 2, the self checking architecture has an identical interface to the synthesisable architecture and instances the synthesisable architecture within itself but also contains behavioral code to capture all input and output signals and check their values against expected values computed using a behavioral model. When errors are detected assertions are triggered and the simulation is stopped with an error message.

This self-checking configuration of the AES-GCM-100G-OTN core can be instantiated within the user's own simulations. This makes it easy to verify the core operates properly when connected to the user circuitry surrounding the core. In addition, the assertions within the self checking code will detect and report many situations where the user design is not driving the core correctly simplifying the task of integrating the core with the larger user design easier.

The AES-GCM-100G-OTN testbench supplied with the core also makes use of the self checking configuration of the core. The testbench stimulates the self checking core with a random sequence of packets, writes of key information and key activations and the self checking core takes responsibility for detecting any errors.

To verify the AES-GCM behavioral model used in the self checking duplex core it is checked against the simpler AES-GCM behavioral model from the AES-GCM-10G core. The AES-GCM-10G core behavioral model has itself been verified using NIST known answer tests and known answer tests produced from well known software implementations of AES GCM.

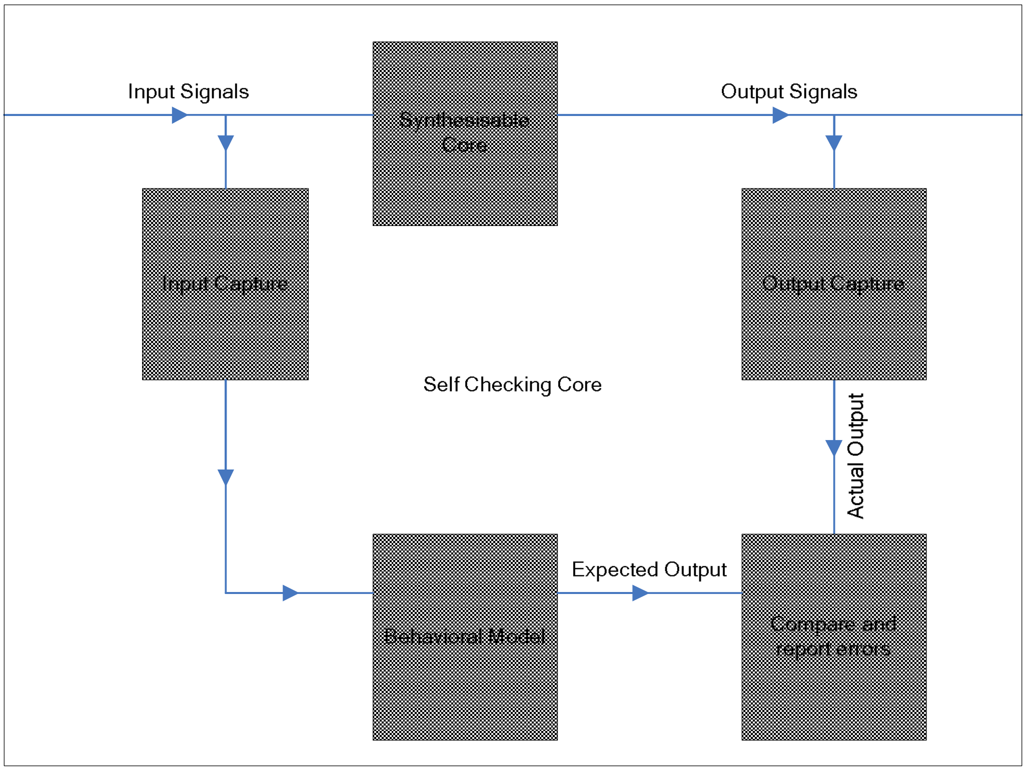

Figure 2, AES-GCM-100G Self Checking Architecture

Recommended Design Experience and Security Considerations

It is recommended that the user is familiar with the VHDL language and with the Xilinx design flow and simulation tools. The core can also be instantiated inside a wrapper to allow use with a Verilog design flow.

It is also recommended that the user has a background in data security or takes appropriate advice when considering how to implement AES-GCM-100G-OTN in a larger system.

A specific security consideration for the AES-GCM algorithm is that the system must ensure that the counter created from the 96 bit IV and 32 bit internal counter can never wrap round so that the same counter value is used twice with the same key. The usual response to detecting that the counter is about to wrap round is to shut down the link until the a new key is installed.

A second security consideration is that a mechanism should be included to detect packet replay (i.e. resending a copy of a valid packet), one way to do this is a packet counter which is expected to increase in value on each received packet. The packet counter can also be used to form part of the IV. Packet replay will not be detected by the authentication mechanism in AES-GCM because the replayed packet will have a valid ICV.

If an incoming packet on the receive channel fails authentication (computed ICV does not match expected ICV transmitted with the packet) or if a replay is detected it must be discarded and not passed on to higher layers in the communication system which may act on it.

Ordering Information

This product is available directly from Algotronix under the terms of the SignOnce IP License. Please contact Algotronix for pricing and additional information about this product using the contact information on the front page of this datasheet. To learn more about the SignOnce IP License program, contact Algotronix or visit the web:

Email:	commonlicense@xilinx.com
URL:	www.xilinx.com/ipcenter/signonce

Export Control

Strong encryption technology such as AES is the subject of international export regulations. Algotronix is located in the United Kingdom and export of this core is regulated by the UK government.

The core is freely available within the European Union and in addition can be supplied immediately to the following countries: United States, Australia, New Zealand, Canada, Norway, Switzerland, Japan.

Export to other countries requires an export licence. The UK Department of Business, Enterprise and Regulatory Reform publishes information on their website (www.berr.gov.uk) which gives an indication of average export licence processing times for various countries and the percentage of licence requests which are granted. For many countries obtaining an export licence can be done relatively quickly and with only a small amount of additional paperwork.

It is the the responsibility of the customer to comply with all applicable requirements with respect to re-export of products containing the AES technology.

Related Information

Industry Information

The AES standard documents FIPS197, SP800-38A and AESAVS, the original GCM proposal to NIST (“The Galois/Counter Mode of Operation (GCM)” by David McGrew and John Viega) and the NIST special publication SP800-38D document describing GCM mode are available from the National Institute of Standards and Technology, Computer Security Resouce Center website (www.csrc.nist.gov).

Xilinx Programmable Logic

For information on Xilinx programmable logic or development system software, contact your local Xilinx sales office, or:

Xilinx, Inc.
2100 Logic Drive
San Jose, CA 95124
Phone: +1 408-559-7778
Fax: +1 408-559-7114
URL: www.xilinx.com

Algotronix® is a registered trademark of Algotronix Ltd. in the United States and United Kingdom and a trademark of Algotronix Ltd. in other countries.
The supply of the product described in this document is the subject of a separate license agreement with Algotronix Ltd. which defines the legal terms and conditions under which the product is supplied. This product description does not constitute an offer for sale, a warranty of any aspects of the product described or a license under the intellectual property rights of Algotronix or others. Algotronix products are continuously being improved and are subject to change without notice. Algotronix products are supplied 'as is' without further warranties, including warranties as to merchantability or suitability for a given purpose. Algotronix' products are not intended for use in safety critical applications.

URL: www.algotronix.com

Version Control Information	
Subversion Revision Number	**66**
Date	2015/03/12 16:25:03
Document	AES GCM 100G For OTN Data Sheet, Xilinx Edition
Status (blank field indicates OK/no warnings)	
	(Table auto-updates, do not edit field values by hand)