IPsec ESP Engine (Helion)

January 18, 2008 Product Specification

Helion Technology

Ash House, Breckenwood Road, Fulbourn,
Cambridge CB21 5DQ, England
Phone: +44 1223 500 924
Fax: +44 1223 500 923
E-mail: helioncores@heliontech.com
URL: www.heliontech.com

AllianceCORE™ Facts

Provided with Core	
Documentation	User Guide
Design File Formats	Xilinx netlist
Constraints Files	.ucf
Verification	VHDL or Verilog test bench; VHDL or Verilog Simulation models
Instantiation templates	VHDL ,Verilog
Reference designs & application notes	
Additional Items	Example ModelSim scripts
Simulation Tool Used	
ModelSim PE 6.1e	
Support	
Support provided by Helion Technology Limited	

Features

- Supports all three ESP security service types
  - Confidentiality only
  - Integrity only
  - Confidentiality and Integrity
- Suitable for use in IPv4 or IPv6 applications
- Suitable for use in IPsec ESP Transport or Tunnel mode operation
- Modular architecture supports all specified ESP encryption and authentication algorithms
  - TripleDES-CBC, AES-CBC and AES-CTR encryption algorithms
  - HMAC-SHA-1-96, HMAC-MD5-96 and AES-XCBC-MAC-96 authentication algorithms
- Support for proposed combined mode algorithms: AES-CCM and AES-GCM
- May be supplied with any combination of ESP encryption and authentication algorithms
- Automatic ESP padding generation and checking
- Supports Traffic Flow Confidentiality padding generation
- Extended Sequence Number support for IKEv2 compatibility
- Available with High or Low rate AES encryption to optimise area vs performance
- Available under terms of the SignOnce IP License

Table 1: Example Implementation – High Rate AES-CBC and HMAC-SHA-1-96 for Xilinx® FPGAs

Family	Example Device	Fmax[1] (MHz)	Slices	IOB[2]	GCLK	BRAM	MULT	DCM/ DLL	MGT	Design Tools
Spartan™-3	XC3S1500-5	99	3040	247	1	18	0	0	N/A	ISE™9.2.03i
Virtex™-4	XC4VLX25-11	185	3042	247	1	18	0	0	0	ISE™9.2.03i
Virtex™-5	XC5VLX30-3	253	2048	247	1	0	0	0	0	ISE™9.2.03i

Notes:

1) Fmax is quoted assuming all core inputs are sourced from flip-flops, and all core outputs drive flip-flops; this has been done to best represent real applications where the core is typically embedded within a larger system.

2) Assuming all core I/Os and clocks are routed off-chip

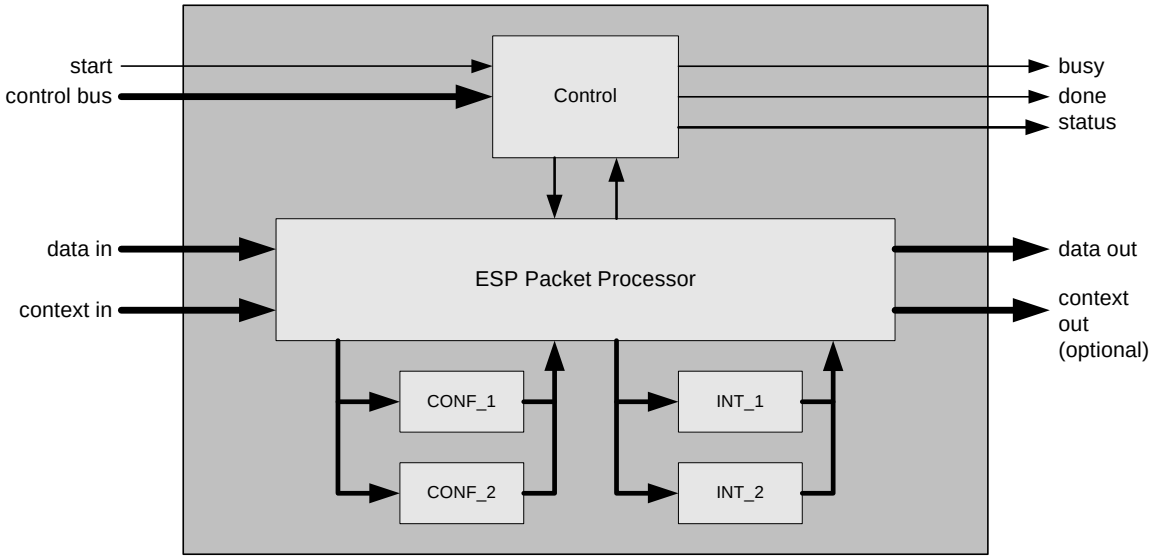

Figure 1: IPsec ESP Security Engine Block Diagram

Applications

IPsec (short for IP security) defines a set of protocols which were developed by the Internet Engineering Task Force (IETF) to allow the secure exchange of IP packets over a public network such as the Internet to establish a Virtual Private Network (VPN). These protocols provide the encryption, authentication, data integrity and key exchange mechanisms required to ensure that only authorised users are able to access the network. The most commonly used security protocol is the Encapsulating Security Payload (ESP) which is designed to provide confidentiality, data origin authentication, and data integrity checking for IPv4 and IPv6 packets.

General Description

The Helion ESP Engine is designed to provide hardware acceleration of the ESP packet processing required to implement an IPsec compliant device. In addition to greatly increasing the data throughput of the underlying encryption and authentication algorithms, offloading ESP packet processing to hardware allows a system CPU to concentrate on the higher level IPsec protocols such as establishment and management of Security Associations, anti-replay detection and key exchanges with communicating IPsec network devices. Although it may easily be used in a system datapath as an ESP processing module, the Helion ESP Engine makes an ideal companion for optimum performance PowerPC and Microblaze IPsec solutions. Taking as an example a user application implementing an IPsec security gateway operating in tunnel mode to illustrate the functions performed by the engine:

For encryption, the engine performs all packet processing required to convert an incoming IP packet from the local private network into an output ESP packet. The security gateway then appends an outer IP header prior to transmission on to the public network.

For decryption, the engine performs all packet processing required to convert an incoming ESP packet from the public network (the outer IP header having been removed) into an output IP packet ready for forwarding to the local private network. In the process, the ESP Engine also detects and reports any integrity check or ESP padding errors to the user application.

Functional Description

The ESP Engine contains the following major design modules as shown in Figure 1. Note that the engine has separate interfaces for data and context. The data interface allows the engine to transfer packet data from and to the user application. The context interface is used to transfer additional data specific to the packet Security Association (SA) context setup by the user application. The context includes ESP header specific information, encryption and authentication keys, and Initialisation Vectors (IV).

Generally the main data input and output interfaces would be connected to FIFOs in the user application. The context interface would be connected to a dual-port RAM buffer containing the IPsec Security Association data setup by the user application prior to starting the ESP Engine.

Control

The control block is responsible for interfacing with the external user application and overall control of the main packet processor datapath.

Packet Processor

The packet processor contains the main datapath elements between the data input and output interfaces of the ESP Engine. It pulls in packet data from the input interface when it is required, switches it through the appropriate confidentiality and integrity modules to perform data encryption and authentication, and pushes out packet data to the output interface when it is valid. At the same time it also pulls in context information relating to the Security Association of the incoming packet; this includes encryption and authentication keys, as well as SPI and Sequence Number ESP header fields.

On encrypting, the ESP Engine takes as input an IP packet, appends ESP header and trailer fields, and outputs an ESP packet. As part of this process it may encrypt the ESP packet data and/or generate and append an integrity checksum value (ICV) as required by the security service specified by the user application.

On decrypting, the ESP Engine takes as input an ESP packet, checks and removes the ESP header and trailer fields, and outputs an IP packet. As part of this process it may decrypt the ESP packet and/or check its integrity and authenticate its origin using the ICV as required by the ESP security service specified by the user application.

Confidentiality Modules

One or more Confidentiality modules (CONF_1 to CONF_N) may be present to provide the required ESP data encryption algorithms e.g. TripleDES-CBC, AES-CBC, AES-CTR.

Integrity Modules

One or more Integrity modules (INT_1 to INT_N) may be present to provide the required ESP data authentication and integrity algorithms e.g. HMAC-SHA-1-96, HMAC-MD5-96, AES-XCBC-MAC-96.

Core Modifications

The ESP Engine is supplied as a pre-configured netlist by Helion to meet the ESP processing requirements of the users application. It is designed to be extremely flexible and through its modular architecture provides support for one or more encryption and authentication algorithms. For instance, it would allow an existing ESP Engine solution using TripleDES-CBC encryption and HMAC-SHA-1-96 authentication to be easily upgraded to also implement the AES-CBC algorithm without any changes to the application interface should system requirements change.

In order to provide the optimum trade-off between logic area and performance for any IPsec application, the ESP Engine may be provided with either High or Low rate AES security modules based on the Helion Fast or Standard AES cores respectively.

Table 2: Example Implementation – High Rate AES-CBC, TripleDES-CBC and HMAC-SHA-1-96

Family	Example Device	Fmax[1] (MHz)	Slices	IOB[2]	GCLK	BRAM	MULT	DCM/ DLL	MGT	PPC	Design Tools
Spartan-3™	XC3S1500-5	100	3802	247	1	18	0	0	N/A	N/A	ISE9.2.03i
Virtex-4™	XC4VLX25-11	191	3796	247	1	18	0	0	0	0	ISE9.2.03i
Virtex5™	XC5VLX30-3	243	2342	247	1	0	0	0	0	0	ISE9.2.03i

Notes:

1) Fmax is quoted assuming all core inputs are sourced from flip-flops, and all core outputs drive flip-flops; this has been done to best represent real applications where the core is typically embedded within a larger system.
2) Assuming all core I/Os and clocks are routed off-chip

Table 3: Example Implementation – Low Rate AES-CBC, TripleDES-CBC and HMAC-SHA-1-96

Family	Example Device	Fmax[1] (MHz)	Slices	IOB[2]	GCLK	BRAM	MULT	DCM/ DLL	MGT	PPC	Design Tools
Spartan-3™	XC3S1500-5	110	2550	247	1	5	0	0	N/A	N/A	ISE9.2.03i
Virtex-4™	XC4VLX25-11	189	2553	247	1	5	0	0	0	0	ISE9.2.03i
Virtex5™	XC5VLX30-3	232	1500	247	1	0	0	0	0	0	ISE9.2.03i

Notes:

1) Fmax is quoted assuming all core inputs are sourced from flip-flops, and all core outputs drive flip-flops; this has been done to best represent real applications where the core is typically embedded within a larger system.
2) Assuming all core I/Os and clocks are routed off-chip

Tables 2 and 3 show the comparable resource estimates, and maximum clock frequency for the Helion ESP Engine configured for AES-CBC and TripleDES-CBC encryption algorithms, and HMAC-SHA-1-96 authentication algorithm. Comparing the two tables it can be seen that the Low rate ESP Engine is significantly smaller in both logic and Block RAM resource, and represents the ideal solution for low to mid-rate ESP applications. This option is often appropriate for data rates up to a few hundred Mbps, depending on the Xilinx technology, the security suite required and the lengths of the packets. However, for the highest data rates (typically up to 1Gbps) it is necessary to specify the High Rate ESP Engine which consequently requires higher device resource usage. For full details on the performance of the Helion ESP core in specific situations, please contact Helion directly.

Core I/O Signals

The core signal I/O have not been fixed to specific device pins to provide flexibility for interfacing with user logic. Descriptions of all signal I/O are provided in Table 4.

Table 4: Core I/O Signals.

Signal	Width	Signal Direction	Description
clk	1	Input	master clock input
reset	1	Output	asynchronous reset; 1 = reset
data_in_required	1	Output	data word in required
data_in	32	Input	data word in
data_in_words_avail	5	Input	number of words available in input buffer
data_out	32	Output	data word out
data_out_valid	1	Output	data word out valid
data_out_free_space	5	Input	number of words of free space available in output buffer
data_out_reserve_space	1	Output	reserve ahead space for one word in output buffer
ctxt_word_in	32	Input	SA context word in
ctxt_in_select	3	Output	SA context input select
ctxt_in_index	3	Output	SA context input index
ctxt_in_done	1	Output	SA context input complete
ctxt_word_out	32	Output	SA context word out
ctxt_word_out_valid	1	Output	SA context word out valid
ctxt_out_select	3	Output	SA context output select
ctxt_out_index	3	Output	SA context output index
encrypt_ndecrypt	1	Input	ESP processing direction; 1=encrypt, 0=decrypt
engine_start	1	Input	ESP engine start request
engine_status	4	Output	ESP engine status outputs; individual bit flags indicate padding or integrity check failures (Decrypt direction only)
engine_busy	1	Output	ESP engine busy indicator
engine_done	1	Output	ESP Engine done pulse – signals that status output is valid
esn_enable	1	Input	64-bit Extended Sequence Number (ESN) processing enable
security_type	8	Input	security service type
input_packet_length	16	Input	length of input packet in bytes
tfc_padding_enable	1	Input	Traffic Flow Confidentiality padding generation enable
next_header_in	8	Input	ESP trailer Next Header byte in (Encrypt direction only)
output_packet_length	16	Output	length of output packet in bytes
next_header_out	8	Output	ESP trailer Next Header byte out (Decrypt direction only)
esp_padding_length	8	Output	ESP trailer Padding Length in bytes (Decrypt direction only)

Verification Methods

The Helion ESP Engine has been thoroughly verified under simulation, using a large suite of standard and Helion generated test vectors. In addition, it has been fully proven in real Xilinx hardware, and has been deployed by our customers within real products out in the field.

Recommended Design Experience

Users should be familiar with HDL methodology and Xilinx design flows including VHDL/Verilog component instantiation, synthesis, implementation and simulation.

Ordering Information

Since the ESP Engine is fully configurable and can support one or more encryption and authentication algorithms the required security modules must be specified at the time of ordering.

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

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

Export

Strong encryption technology such as the Triple-DES and AES algorithms are governed internationally by export regulations. Immediate export is permitted to the following countries;

Austria	Australia	Belgium
Bulgaria	Canada	Cyprus
Czech Republic	Denmark	Estonia
Finland	France	Germany
Greece	Hungary	Ireland
Italy	Japan	Latvia
Lithuania	Luxembourg	Malta
New Zealand	The Netherlands	Norway
Poland	Portugal	Romania
Slovakia	Slovenia	Spain
Sweden	Switzerland	United Kingdom
United States		

Please contact Helion to discuss delivery to other destinations; approval is subject to the applicable export licenses being granted. Please note that licensees are responsible for complying with the applicable requirements for re-export of electronics containing Triple-DES and AES technology.

Related Information

For more detailed information on the IPsec protocol, the RFC “Security Architecture for the Internet Protocol” may be downloaded from the IETF website :

http://www.ietf.org/rfc/rfc4301.txt

For more detailed information on the ESP security protocol, the RFC “IP Encapsulating Security Payload (ESP)” may be downloaded from the IETF website :

http://www.ietf.org/rfc/rfc4303.txt

For more general information, including a full list of IPsec related RFC's and other links :

http://en.wikipedia.org/wiki/IPsec

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