An IPSec Accelerator Design for a 10Gbps In-Line Security Network Processor

Yun Niu
Institute of Microelectronics, Tsinghua University
Beijing, China
Email: niuy08@mails.tsinghua.edu.cn

Liji Wu and Xiangmin Zhang
Institute of Microelectronics, Tsinghua University
Beijing, China
Email: {lijiwu, zhxm}@ mail.tsinghua.edu.cn

***Abstract*—The IP security protocol (IPSec) is an important and widely used security protocol in the IP layer. But the implementation of the IPSec is a computing intensive work which greatly limits the performance of the high speed network. In this paper, a high performance IPSec accelerator used in a 10Gbps in-line network security processor (NSP) is presented. The design integrates the protocol processing and the cryptographic processing; the transport/tunnel mode of the AH, ESP security protocols and the AES, HMAC-SHA-1 cryptographic algorithms are realized by hardware. An efficient partial crossbar data transfer skeleton with iSLIP scheduling algorithm is adopted to realize the maximum utilization of the computation resources in the accelerator. The number of AH, ESP, AES, HMAC-SHA-1 cores in the design can be configured to meet the different applications. By simulation, with 8 protocol IP-cores and 24 crypto IP-cores connected to the crossbar in the IPSec accelerator, the design gives a peak throughput for the AH protocol transport mode of 11.28Gbps at the average of 512 bytes packet length under a clock rate of 300MHz. The hardware verification is implemented on a Virtex-5 XC5VSX95T based FPGA board. Low power design methods are also used in the design to reduce the power dissipation.**

***Index Terms*—IPSec, network security processor, 10Gbps Ethernet, cryptographic algorithm, crossbar switch**

I. INTRODUCTION

Network speed is increasing dramatically over the past decade and the total bandwidth of communication systems triples every twelve months according to the Gilder's Law. At the same time, more and more personal data will be exposed on the internet; the need to protect the information security becomes more urgent then before. At present the network security is becoming an essential network issue. The IPSec (Internet Protocol Security) security protocol [1] developed by the IETF (Internet Engineering Task Force) in 1998 is one popular solution to protect the data transfer at the IP layer.

The IPSec protocol can provide access control, connectionless integrity, data origin authentication, protection against replays, confidentiality, and limited traffic flow confidentiality. It includes three main protocols: the AH (Authentication Header) protocol, the ESP (Encapsulating Security Payload) protocol and the IKE (Internet Key Exchange) protocol. In this paper, the AH and ESP protocols are considered and implemented. The former provides data origin authentication and connectionless integrity, and the latter allows encryption and optionally authentication of the data. The IPSec supports two modes of operation, tunnel mode and transport mode. In transport mode, only the upper-layer protocol data segment of the IP packet is authenticated or encrypted and it is typically used for end-to-end protection of data packets between two hosts. In tunnel mode the entire IP packet is authenticated or encrypted within a new outer IP header. The tunnel mode can be used between security gateway (router or firewall) to create a VPN (virtual private network).

Now, the IPSec protocol is almost embedded into TCP/IP protocol stack via software in OS (operating system), such as Linux and NetBSD. But the IPSec has proved to be computationally very intensive [2] which greatly affects the performance of the network.

Data throughput in core routers has all ready achieved to terabits today, and the line card interface speed is already 10Gbps or above. But the high performance internet security device is far behind. The mean reason is the data processing for security is complex and time consuming. So, it is difficulty for the security devices to achieve the equal performance as the internet devices.

Hardware implementation of the IPSec may be a good solution. Hardware implementations of the security algorithm accelerators to improve the performance of the cryptographic processing needed by the IPSec can be seen in [3] [4] [5], which greatly off-loading the heavy computational intensive work of the IPSec. But they ignored the protocol processing which also have been shown to be time consuming [2] and can not satisfy the rapid speed of the network, especially for 10 Gigabit network now. In the works [6] and [7], although the IPSec protocol processing is presented, their performance is far below the 10Gbps data transfer rate. To improve the

doi:10.4304/jcp.8.2.319-325

performance of the network security, multi-core processor design is a hot research field [8] [9] today, but in these design, the IPSec protocol processing is still implemented by software.

This paper discusses an IPSec accelerator design for a 10Gbps in-line network security processor. The IPSec accelerator can implement the transport/tunnel mode of the AH and ESP protocol, and supports the AES-128/192/256, HMAC-SHA-1 algorithms. The AH/ESP protocols and AES/HMAC-SHA-1 algorithms are hardware implemented separately for the purpose of configurability. In the IPSec accelerator, an efficient crossbar switch data transfer skeleton with optimized scheduling algorithm is adopted between multi cores which fully utilized the computation resources in the design. By simulation, with 8 AH/ESP protocol cores and 24 AES/HMAC-SHA-1 algorithms cores configured in the design, the IPSec protocol processing can achieve 11.28Gbps wire speed at 300MHz clock frequency under the average 512 bytes packet length. Hardware verification with four parallel AH/ESP, AES/HMAC-SHA1 cores in the IPSec accelerator by FPGA at 100MHz is also realized.

The remainder of the paper is organized as follows. In section 2, the architecture of a 10Gbps in-line security network processor is described. Section 3 introduces the IPSec accelerator design in detail. Simulation and verification of the design is presented in section 4. Section 5 is the conclusion of this paper.

II. The In-line Network Security Processor Overview

The IPSec accelerator represented here is designed for a 10Gbps in-line network security processor (NSP) that can process IP packets in or out of bound from 10 Gigabit network at wire speed. The architecture of the in-line NSP system is shown in Fig. 1. It integrates the high speed data transfer, the IPSec protocol processing and the cryptographic operations in a single chip. The 10Gbps in-line NSP consists of an 32-bit embedded RISC CPU core, a IPSec accelerator, a packet processor, two 10Gbps MAC interface modules, a high speed SPD (Security Policy Database) look-up module, a on-chip SAD (Security Assistant Database) SRAM, an off-chip crypto-controller, a SDRAM controller for off-chip main memory, a SRAM controller for off-chip SAD database.

In the in-line NSP system, the 32-bit embedded CPU completes the data flow control by software, including inquiry the status of the packet processor, analyze the configuration information send by the SPD look-up module and allocate tasks to the IPSec accelerator based on the idle status of it. The packet processor buffers the ingress and egress packets, extracts IP header and distributes the packets from/to 10Gbps MAC interfaces. The cryptographic operation (data encryption/decryption and authentication) and AH/ESP protocol processing are implemented by the IPSec accelerator. The off-chip crypto controller can substitute the standard cryptographic and authentication algorithms with the custom-specific off-chip ones; therefore, the in-line NSP has algorithm-scalability. The SPD look-up module is designed for accomplishing SPD look-up and security policy analyses. The on-chip SAD SRAM can store 256 items security associations. To improve the data transfer efficiency, the crossbar switch structure is used in the system.

Software and hardware co-design method is used in the system. The 64-bit data width and 32-bit address width crossbar bus architecture using the WISHBONE protocol [10] is adopted between multi cores to improve the data throughput.

In this work, we will focus on the design of the IPSec accelerator for the 10Gbps in-line NSP.

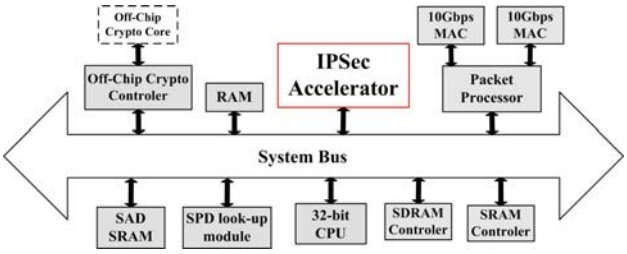

Figure 1. Architecture of the In-line NSP.

III. IPSec Accelerator Design

A. Structure Overview

To achieve high performance of the IPSec accelerator, we adopted the structure of Fig. 2. It is composed of the protocol processing modules AH and ESP, cryptographic processing modules AES and HMAC-SHA-1, and a data transfer module. Each core is designed carefully to meet the need of the 10Gbps data processing in the in-line NSP and the interfaces of these cores are unified for reusable. The crossbar switch architecture based on WISHBONE bus protocol is adopted to implement the simultaneous data transfer between the many cores.

In the design, the power dissipation is also considered. To reduce the power, a power management unit (PMU) is designed. For the cores not in use, the power to them is shutdown by control signal from power management unit. In addition, clock management method such as clock gating is also used in the design to reduce the power.

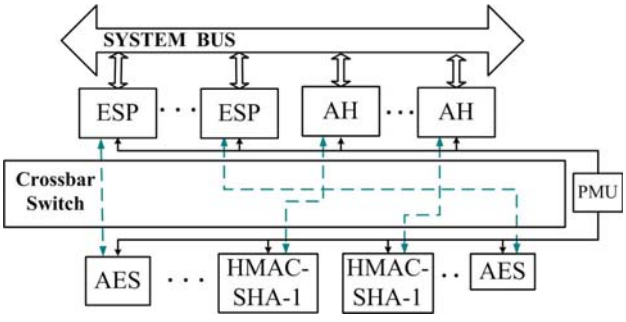

Figure 2. Structure of the IPSec accelerator

B. Data Flow

To improve the data throughput of the design, we studied the feature of the IPSec processing in the cryptographic and protocol components and broke the whole data path into three stages: protocol pre-processing, data crypto-processing, protocol post-processing. Two FIFO are inserted before and after crypto-processing stage which forms a three stage pipeline in the data path shown in Fig. 3 (a).

The data flow of the IPSec Accelerator is shown in Fig. 3 (b). First, the AH/ESP core read key and data from the SAD SRAM and the packet processor respectively, after that, the pre-processing is invoked. Based on the information of the pre-processing, the proper AES/HMAC-SHA-1 core is selected by the crossbar switch, and the data encryption/decryption or authentication is performed. After that, the crypto-processed data is send back to AH/ESP core to do post-processing.

Take the advantage of the crossbar switch structure, after the AH/ESP finishing the pre-processing and sending the data to the AES/HMAC-SHA-1 core, the AH/ESP core can read and process the next packet, needing not wait the AES/ESP core finished the data crypto-processing, this reduced the number of the AH/ESP core and improved the utilization of the cores. Through simulation, the number of AH/ESP core can be reduced from 16 to 8 (2 AH cores and 6 ESP cores), and the AES cores is reduced from 16 to 8, the HMAC-SHA-1 cores is reduced from 32 to 24. This greatly saves the area of the design and eases the design complexity at the same time.

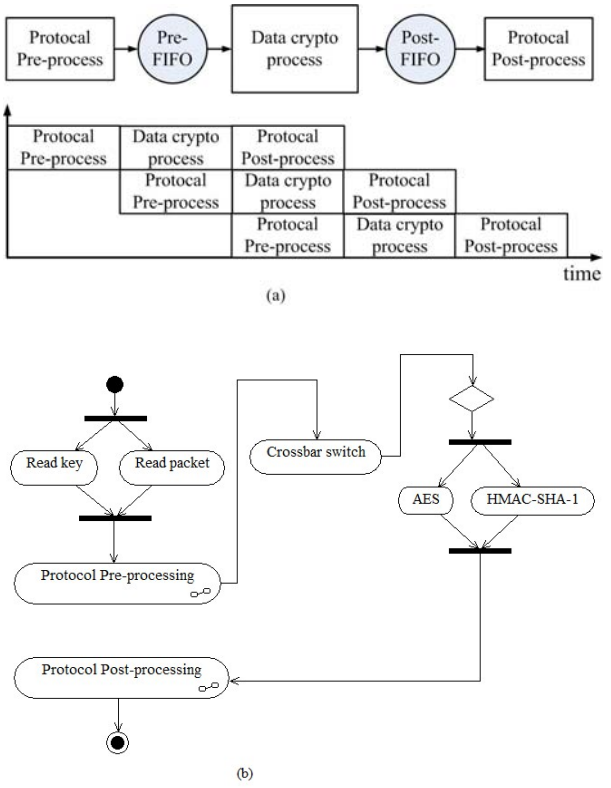

Figure 3. The IPSec accelerator: (a) the pipeline (b) Data flow

C. AH/ESP Core Design

The functions of the AH and the ESP cores include modifying original IP header and generating new IP header; generating and removing the AH/ESP header based on the operating mode; invoking the corresponding cryptographic algorithm core to encrypt, decrypt or verify the data.

It is reported that nearly 48% of the IPSec processing are spent in protocol processing [11]. To reduce the time consuming of the protocol processing, we analyzed the characteristics of the protocol processing, and adopted the block diagram of the AH and ESP core shown in Fig. 4 (a), (b) separately. Take the ESP core for example, it includes an ESP controller, a key engine, a pre-crypto engine, a pre-verify engine and a post-engine.

For the outbound data of the ESP in the tunnel mode, at first, the ESP controller generates the control signals based on the information (key, IV, SPI, sequence number and other control messages) stored in the configure registers which is written by the embedded 32-bit CPU of the in-line NSP. The key is processed by the key engine and some configure information is added to the key. The packet read from the packet processor through system bus interface is processed by the pre-crypto engine first. In the pre-crypto engine, the checksum of the inner IP header is handled and the packet is padded to fulfill the need of the AES. Then, the AES core is activated to complete data encryption. The pre-verify engine processed the data encrypted by AES to fulfill the need of the HMAC-SHA-1 and generated the ESP header. The encrypted and authenticated packet is sent to the post-engine, the ESP tunnel header based on the information extracted from the configure registers is added. Finally, the processed packet with ESP tunnel header between IP header and payload is sent out through system bus interface to the packet processor.

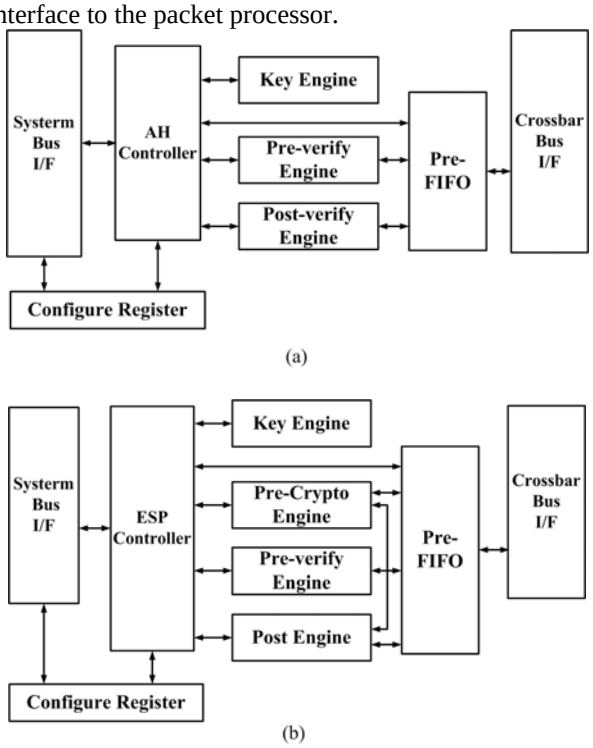

Figure 4. Block diagram of IPSec Cores: (a) AH (b) ESP

Also, a 4-stage pipeline is adopted, and the time-consuming of the protocol processing is reduced to 18.5% of the whole IPSec processing as shown in Fig. 5. The AH and ESP cores are implemented by the Verilog HDL(Hardware Design Language), and simulated, synthesized based on 65nm CMOS technology, the simulation results are shown in Table 1.

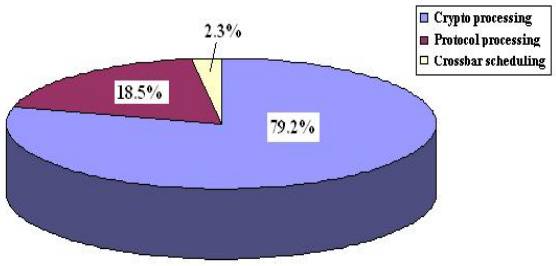

Figure 5. Occupied time of the protocol processing

TABLE I.

SIMULATED AND SYNTHESIZED RESULTS OF AH AND ESP PROTOCOL CORES

Module	Frequency	Throughput	Gate Counts	Power Estimate
AH	300MHz	0.91Gbps	32,886	6.79mW
ESP	300MHz	0.61Gbps	39,194	9.27mW

D. Cryptographic Algorithm Core Design

The design of the cryptographic algorithm cores AES and HMAC-SHA-1 are based on our previous work [12] [13]. The specific area/performance tradeoffs and hardware implementation features of each kind of crypto cores were considered to optimize the performance by adopting pipeline design methodologies. The block diagram of AES and HMAC-SHA-1 core is shown in Fig. 6.

The AES crypto core implements the block size of 128 bits and key lengths of 128, 192 and 256 bits. The crypto module implements one round of a Rijndael encryption in a fully parallel non-pipelined fashion and the Rijndael encryption can be completed in one clock cycle per round. The dynamic key generator module uses the on-the-fly scheme and generates subkey simultaneously with the encryption process in every round. AES uses 16 256-byte substitution boxes (Sboxes) for each encryption round and another 16 256-byte inverse Sboxes for each decryption round.

The HMAC-SHA-1 core is designed on the basis of implementing hash function SHA-1 by using iteration method. In the SHA-1 design, we reduced the critical data path and improve the throughput by scheduling the single step algorithm.

To satisfy the whole IPSec accelerator design, the interface of AES and HMAC-SHA-1 is modified, and a FIFO is added to realize pipeline data path.

The AES core and the HMAC-SHA-1 core are also synthesized and implemented based on 65nm CMOS technology separately; the results are shown in Table 2.

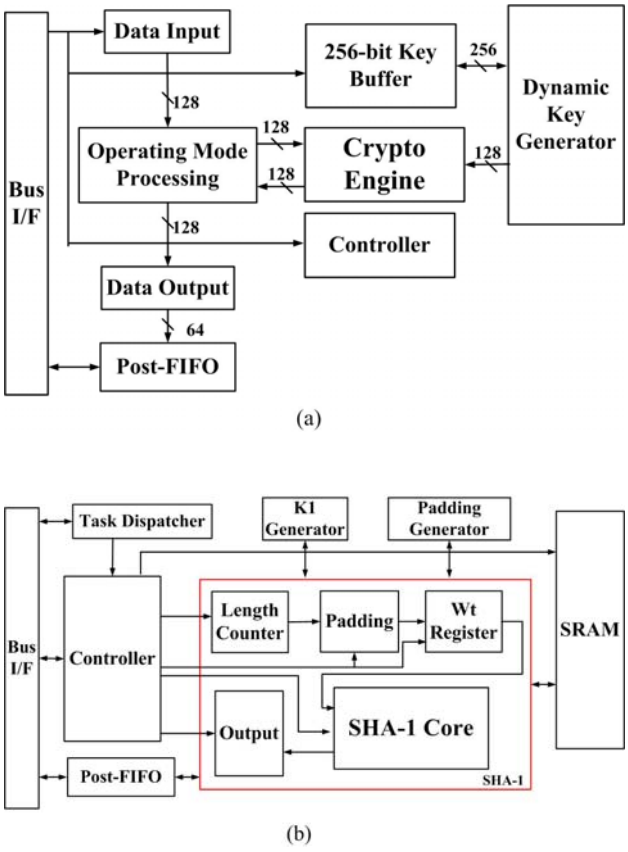

Figure 6. Block diagram of the cryptographic algorithm core (a) AES (b) HMAC-SHA-1

TABLE II.

SIMULATED AND SYNTHESIZED RESULTS OF AES AND HMAC-SHA-1 CORES

Algorithm	Frequency	Throughput	Gate Counts	Power Estimate
AES-256	300MHz	3.1Gbps	62,289	9.2mW
HMAC-SHA-1	300MHz	3.4Gbps	22,575	4.8mW

E. Crossbar Switch Design

In the IPSec accelerator, the choice of data transfer method between AH/ESP core and AES/HMAC-SHA-1 cores is a key factor to achieve the high performance. Traditionally used hierarchical shared bus communication architectures such as those proposed by AMBA [14] are not scalable to cope with the demands of very high performance systems. Network-on-Chip (NoC) based communication architectures have discussed widely for multi-core system design, but it implemented much more complexity, and not fit our design. So, the crossbar switch is adopted for multi-core data transfer in the design. It can provide multiple independent paths for concurrent data transfer between the cores that greatly improve the transfer efficiency. Also, the number of ports can be configured to meet the different application. The crossbar switch is designed based on WISHBONE bus protocol. It is a simple, compact and easy to implement protocol and can provide point-to-point, data flow, shared bus and crossbar interconnects.

The block diagram of the crossbar switch is shown in Fig. 7. It includes a scheduler and MUX/DEMUX controlled by the input/output decision signals generated by the scheduler. Based on the fact that the AH core only communicate with HMAC-SHA-1 core, we adopted partial crossbar bus, that is AH cores only connected with HMAC-SHA-1 cores, which greatly reduce the connection complexity of the design.

In the crossbar switch design, the scheduler is the most important part. The delay between the grant and accept arbiters directly affects the speed of the data transfer. The iSLIP [15] scheduling algorithm is chosen to perform the scheduling between the AH/ESP cores and the AES/HMAC-SHA-1 cores. The iSLIP algorithm includes three phases: request, grant and accept in the following:

Step 1: Request. Each unmatched input sends a request to every output which for which it has a queued cell.

Step 2: Grant. In an unmatched output receives any requests, it chooses the one that appears next in a fixed, round-robin schedule starting from the highest priority element. The output notifies each input whether or not its request was granted. The pointer to the highest priority element of the round-robin schedule is incremented to one location beyond the granted input if the grant is accepted in Step 3 of the first iteration.

Step 3: Accept. If an unmatched input receives a grant, it accepts the one that appears next in a fixed, round-robin schedule starting from the highest priority element. The pointer to the highest priority element of the round-robin schedule is incremented to one location beyond the accepted output only if this input was matched in the first iteration.

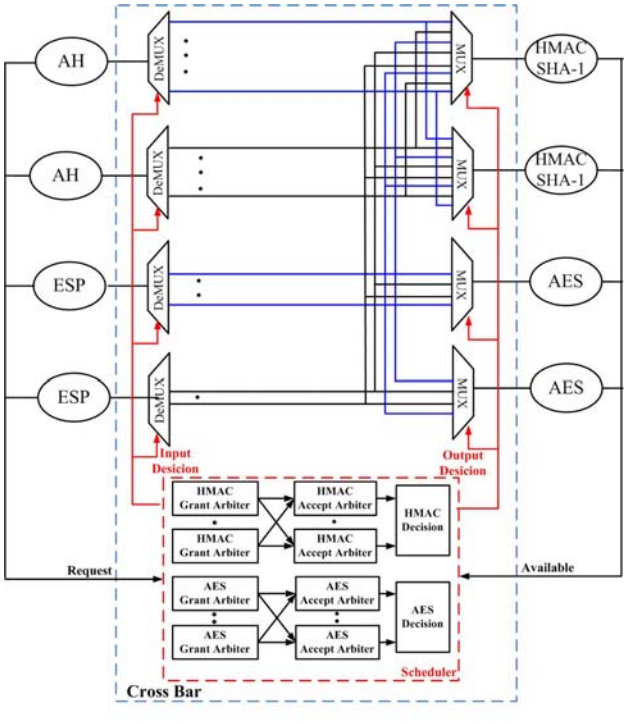

Figure 7. Block diagram of crossbar switch

Fig. 8 (a) illustrates the arbiter design [16] chosen for this implementation. The arbiter is based on a simple round-robin arbiter. It includes an update_enable signal to allow the iSLIP algorithm to only update the priority under certain circumstances. The PPE (Priority Process Engine) chosen for this design was shown in Fig. 8 (b) [17]. The input signals P_enc to the Simple_PE_thermo block is masked by a thermometer encoding based on the programmed priority level. The non-masked Simple_PE does not take a priority, and determines the output when the masked PE does not find any 1-input between the programmed priority and input. It showed that this design produces minimized delay, and has a smaller area [17].

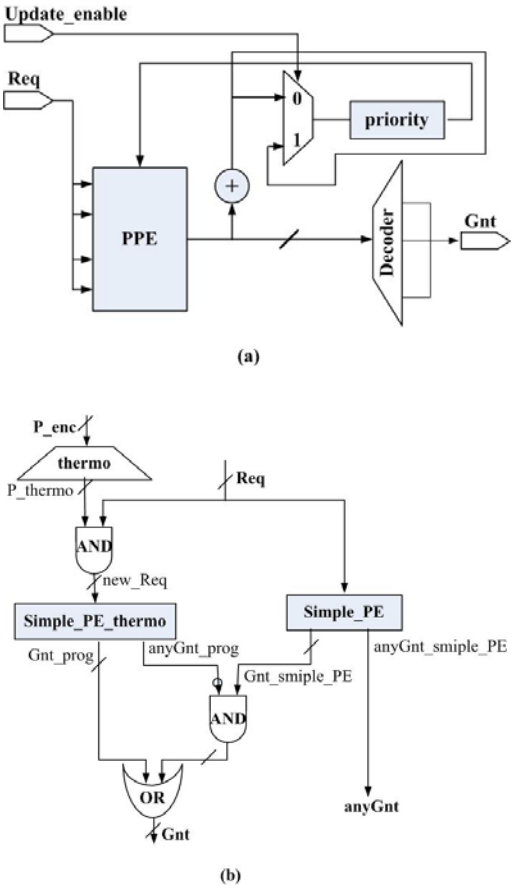

Figure 8. (a) Arbiter diagram (b) PPE structure

IV. Simulation and Verification

A. Modeling and Simulation

To realize 10Gbps in-line performance, the number of the AH/ESP and AES/HMAC-SHA-1 cores to be used must be determined. A transaction level simulation model based on the SystemC [18] language shown in Fig. 9 is established. SystemC is based on C++, so it can provide a mechanism crucial to modeling hardware while using a language environment compatible with software development. The basic components in the SystemC are module, channel, interface and port. In the model, the functions of the AH/ESP and the AES/HMAC-SHA-1are implemented through modules, crossbar switch is implemented by the channel, the connections between the modules are realized through ports and interfaces, the communication between the modules is accomplished by

the channel. The number of AH/ESP and AES/HMAC-SHA-1 modules can be configured to acquire the optimum performance.

In the simulation, the AH/ESP modules take as the initiators and use the SCV (SystemC Verification Library) to generate variable length test packets. The length of the test packets is from 64 bytes to 1500 bytes. The AES/HMAC-SHA-1 modules take as the responses. The crossbar switch channel implements the iSLIP scheduling algorithm.

By configure the number of AH/ESP and AES/HMAC-SHA-1 modules, we got that with 8 AH/ESP modules and 24 AES/HMAC-SHA-1 modules, the IPSec accelerator can realize the capability of the 10Gbps in-line data processing.

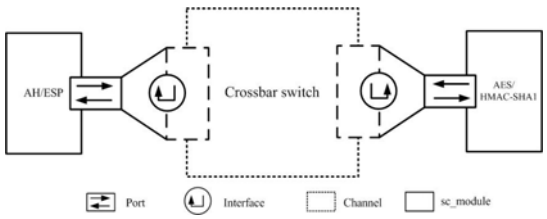

Figure 9. IPSec accelerator simulation modelling

B. Verification

Based on the simulation results above, an IPSec accelerator composed with 8 AH/ESP protocol cores and 24 AES/HMAC-SHA-1 algorithms cores connected to a crossbar switch is implemented at Register Transfer Level (RTL) and simulated and synthesized based on the 65nm CMOS technology. The simulation results showed that the data throughput of the IPSec accelerator achieved up to 11Gbps under the clock frequency of 300MHz. The simulation and synthesized results are shown in Table 3.

Hardware verification on a FPGA board shown in Fig. 10 is accomplished. The FPGA board mainly includes a 10Gbps media connector, a Virtex-5 XC5VSX95T FPGA, and a JTAG port for downloading the IPSec accelerator code. For the content limitation of the FPGA on the board, the AH protocol and the ESP protocol verification are performed separately. The results are shown in Table 4.

TABLE III.

SIMULATED RESULTS OF IPSEC ACCELERATOR WITH 8X32 CROSSBAR SWITCH

Mode @Protocol	Frequency (MHz)	Throughput (Gbps)	Gate Counts	Power Estimation (mW)
Transport @AH	300	11.89	1118,166	258.56
Tunnel @ESP	300	11.28		

TABLE IV.

FPGA VERIFICATION RESULTS OF THE IPSEC PROCESSOR AT DIFFERENT CONFIGURATIONS

Module	Frequency (MHz)	LUT Numbers	FIFO Numbers
AH x4 HMAC-SHA-1 x4	100	27832	17
ESP x4 AES x4 HMAC-SHA-1 x4	100	30366	40

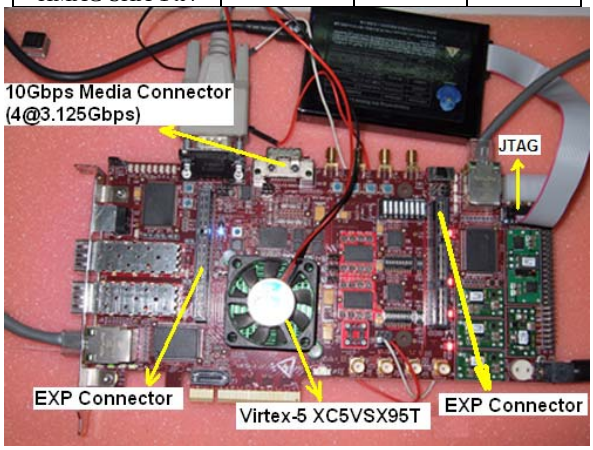

Figure 10. Verification Board with Virtex-5 XC5VSX95T FPGA

V. CONCLUSION

A high performance configurable IPSec accelerator for a 10Gbps in-line NSP is presented. The IPSec protocol processing and cryptographic operations are integrated in the design. The crossbar switch interconnect network adopted greatly improves the data transfer efficiency and makes the number of the protocol processing cores (AH and ESP) and cryptographic algorithm cores (AES and HMAC-SHA-1) in the design can be configured. A simulation model based on the SystemC language is established. The model can implement the transaction level simulation and improve simulation efficiency. By using the power shut off and clock gating design methods, the power dissipation is well controlled in a rational range. The simulation results show that with 8 AH cores, 8 AES cores and 16 HMAC-SHA-1 cores connected on the crossbar switch, the IPSec accelerator throughput at AH transport mode is 11.28Gbps at the clock frequency of 300MHz which satisfied the need of the 10Gbps in-line NSP system. Hardware verification on a Viretex-5 FPGA board is also implemented. Take the advantage of the configurability of the design, the design can satisfy the 40G/100G Gigabit Ethernet for next internet if more IP cores are configured in the configurable IPSec accelerator.

ACKNOWLEDGMENT

This work is supported by the Core-High tech-Basic Important National Specific project of the Ministry of Industry and Information Technology of China (2011ZX01034-002-002-003).

REFERENCES

[1] S. Kent, and R. Atkinson, “Security architecture for the internet protocol,” IETF network working group, RFC2401, 1998.

[2] Alberto Ferrante, Vincenzo Piuri, and Jeff Owen, “IPSec Hardware Resource Requirements Evaluation,” Next

Generation Internet Networks (NGI 2005), April 2005. pp.240-246, “doi:10.1109/NGI.2005.1431672”
[3] Salman Ahmad, Marcin Rogawski, Jens-Peter Kaps, “Efficient hardware accelerator for IPSec based on partial reconfiguration on Xilinx FPGAs,” 2011 International Conference on Reconfigurable Computing and FPGAs, Nov. 2011. pp.242-248, “doi:10.1109/ReConFig.2011.33”
[4] Wang Mao-Yin, Su Chih-Pin, Horng Chia-Lung, Wu Cheng-Wen, Huang Chih-Tsun, “Single and multi-core configurable AES architectures for flexible security,” IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol.18(4), pp.541-552, April 2010.
[5] Haixin Wang, Guoqiang Bai, and Hongyi Chen, “Zodiac: System Architecture Implementation of a High Performance Network Security Processor,” Proc. IEEE International Conference on Application-Specific Systems, Architectures and Processors, Jul. 2008. pp.91–96, “doi:10.1109/ASAP.2008.4580160”
[6] Alberto Ferrante, Vincenzo Piuri, “High-level Architecture of an IPSec-dedicated System on Chip,” Next Generation Internet Networks, 3nd EuroNGI Conference, May 2007. pp.159-166, “doi: 10.1109/NGI.2007.371211”
[7] T. Blackwell, “Speeding up protocols for small messages,” Proceedings of the 1996 ACM SIGCOMM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, 1996. pp.85–95, “doi: 10.1145/248157.248165”
[8] Yizhen, Daxiong Xu, Wuying Song, and Zhixin Mu, “Design and Implementation of High Performance IPSec Applications with Multi-core Processors,” International Seminar on Future Information Technology and Management Engineering, Nov. 2008. pp.595-598, “doi: 10.1109/FITME.2008.88”
[9] Brian Miller, Derek Brasili, Tim Kiszely, Rob Kuhn, Rahul Mehrotra, et al. “A 32-Core RISC Microprocessor with Network Accelerators, Power Management and Testability Features,” International Solid-State Circuits Conference (ISSCC), Feb. 2012. pp.58-60, “doi: 10.1109/ISSCC.2012.6176877”
[10] Richard Herveille, “WISHBONE System-on-Chip (SoC) Interconnection Architecture of portable IP Cores,” http://opencores.org/, 2002.
[11] Nachiketh R. Potlapally, Srivaths Ravi, Anand Raghunathan, Ruby B. Lee, and Niraj K. Jha, “Impact of Configurability and Extensibility on IPSec Protocol Execution on Embedded Processors,” Proceedings of the 19th International Conference on VLSI Design, Hyderabad, India 2006. pp.299-304, “doi: 10.1109/VLSID.2006.102”
[12] Liji Wu, Yingjie Ji, Xiangmin Zhang, Xiangyu Li, and Yongsheng Yang, “Power analysis resistant AES crypto engine design for a network security co-processor,” Journal of Tsinghua University Science and Technology, vol.49, No. S2, pp.2097-2102, Dec. 2009.
[13] Chen Yingjie, Wang Haixin, Bai Guoqiang, and Chen Hongyi, “A VLSI IP Module Design for Implementing Multi-hash Function,” Microelectronics and Computer, val.27, No.4, pp. 89-94, April. 2010.
[14] AMBA Specification, ARM. http://www.arm.com, 2001
[15] Nick McKeown, “The iSLIP Scheduling Algorithm for Input-Queued Switches,” IEEE/ACM Transactions on Networking, vol.7, No.2, pp.188-201, April 1999. “doi: 10.1109/90.769767”
[16] John D. Pape, “Implementation of an On-chip Interconnect Using the i-SLIP Scheduling Algorithm,” M.S. Thesis, The University of Texas at Austin, 2006.
[17] Gupta, P. and McKeown, N. “Designing and implementing a fast crossbar scheduler,” Micro, IEEE, vol.19, No.1, pp.20-28, Jan/Feb 1999. “doi: 10.1109/40.748793”
[18] Open SystemC Initiative, http://www.systemc.org/, 2012.

Yun Niu received the M.S. degree in electronic engineering from Beijing University of Technology, Beijing, China, in 2002. From 2008, she has been a Ph.D. student with the Institute of Micro-electronics, at Tsinghua University, Beijing, China.

From Aug. 2002 to Aug. 2008, she worked in the Beijing electronic technology research institute, as an IC design engineer. Her research interests focus on the information security and the System-on-Chip (SoC) design methodology.

Liji Wu received the B.S., M.S., and Ph.D. degrees in electronic engineering from Tsinghua University, Beijing, China, in 1988, 1991 and 1997, respectively.

From April 1997 to May 2000, he worked in the Center for Advanced Technology in Telecommunications, Polytechnic Institute of New York University (NYU-Poly), Brooklyn, New York, as a Postdoctoral Fellow, worked on design and implementation of high-speed control circuits and systems utilized in WDM ATM Multicast optical switching systems sponsored by DARPA. Then he worked in high-tech industry in the U.S. for more than 4 years, including TyCom Laboratories (former AT&T Bell Labs on Undersea Optical Fiber Communications), Eatontown, New Jersey, as a Senior Member of Technical Staff.

Prof. Wu received Tsinghua University Outstanding Graduate Award and Medal in 1988, Beijing, China. He joined Tsinghua University, Beijing, China as a full time faculty since 2005. He is a board member of Shanghai Pudong Science&Technology Association, Shanghai, China since 2006.

Xiangmin Zhang was born in Beijing, P.R. China in 1966. He received the B.S. degree in microelectronics from Peking University, Beijing, China in 1988, and M.S. degree in electronic engineering from Tsinghua University, Beijing, China in 1991.

Since then, he joined the Institute of Microelectronics, Tsinghua University. His main research interests are in information security and automotive electronics.