Journal of Zhejiang University-SCIENCE C (Computers & Electronics)
ISSN 1869-1951 (Print); ISSN 1869-196X (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

# A 10 Gbps in-line network security processor based on configurable hetero-multi-cores*

Yun NIU†1,2, Li-ji WU†‡1,2, Yang LIU1,2, Xiang-min ZHANG1,2, Hong-yi CHEN1,2
(1National Laboratory for Information Science and Technology, Tsinghua University, Beijing 100084, China)
(2Institute of Microelectronics, Tsinghua University, Beijing 100084, China)
†E-mail: niuy08@mails.tsinghua.edu.cn; lijiwu@mail.tsinghua.edu.cn

Received Dec. 22, 2012; Revision accepted Apr. 22, 2013; Crosschecked July 12, 2013

**Abstract:** This paper deals with an in-line network security processor (NSP) design that implements the Internet Protocol Security (IPSec) protocol processing for the 10 Gbps Ethernet. The 10 Gbps high speed data transfer, the IPSec processing including the crypto-operation, the database query, and IPSec header processing are integrated in the design. The in-line NSP is implemented using 65 nm CMOS technology and the layout area is 2.5 mm×3 mm with 360 million gates. A configurable crossbar data transfer skeleton implementing an iSLIP scheduling algorithm is proposed, which enables simultaneous data transfer between the heterogeneous multiple cores. There are, in addition, a high speed input/output data buffering mechanism and design of high performance hardware structures for modules, wherein the transfer efficiency and the resource utilization are maximized and the IPSec protocol processing achieves 10 Gbps line speed. A high speed and low power hardware look-up method is proposed, which effectively reduces the area and power dissipation. The post simulation results demonstrate that the design gives a peak throughput for the Authentication Header (AH) transport mode of 10.06 Gbps with the average test packet length of 512 bytes under the clock rate of 250 MHz, and power dissipation less than 1 W is obtained. An FPGA prototype is constructed to verify the function of the design. A test bench is being set up for performance and function verification.

**Key words:** 10 Gbps Ethernet, Network security processor (NSP), Internet Protocol Security (IPSec), Crossbar
**doi:**10.1631/jzus.C1200370 **Document code:** A **CLC number:** TN918

## 1 Introduction

Performance and security are the two key factors for the Internet. The ever-increasing e-commerce and e-government have made Internet security more important than before. Nowadays the line card interface data throughput in a backbone network has already reached 40 Gbps or above (Chen, 2011). The high performance Internet security devices, however, are far behind because the data handling for security includes not only the packet header and payload data checking, but also the cryptographic operation, which has been proved to be computationally intensive (Ferrante *et al.*, 2005). Thus, it is difficult for security devices to achieve equal performance. As the most popular protocol in local area network (LAN), Ethernet has been deployed extensively by the governments, institutions, and corporations, for all of whom the issue of security is more important. With the 10 Gbps Ethernet becoming more popular, and the proposal of the 40/100 Gbps Ethernet standard (IEEE Std 802.3-2012), study of the Internet security devices at 10 Gbps and higher speeds has become more necessary and urgent than ever.

At present, network security is always achieved by firewall or deep packet inspection (DPI) in the application layer or transport layer. These methods cannot directly and effectively protect the network data. The Internet Protocol Security (IPSec) protocol

‡ Corresponding author

* Project (No. 2011ZX01034-002-002-003) supported by the National Science and Technology Major Projects of the Ministry of Industry and Information Technology, China

suite (RFC2401:1998) developed by the Internet Engineering Task Force (IETF) is a popular solution to the protection of the Internet data at the Internet Protocol (IP) layer. The IPSec protocol provides access control, connectionless integrity, data origin authentication, protection against replays, and confidentiality. It includes two main protocols, the Authentication Header (AH) protocol and the Encapsulating Security Payload (ESP) protocol. AH provides data origin authentication and connectionless integrity by the authentication algorithms. ESP allows data encryption and authentication by the cryptographic algorithms. IPSec supports two modes of operation, transport mode and tunnel mode. In transport mode, only the upper-layer protocol data segment of the IP packet is authenticated or encrypted and it is used typically 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 the security gateways to create a virtual private network (VPN).

Nowadays the IPSec protocol is usually implemented by embedding into TCP/IP protocol stack via software in the operating system, such as Linux. Although the main frequency of the general CPU has achieved several GHz, software-implemented IPSec still cannot meet the high speed network security demand because the IPSec includes a significant amount of protocol processing such as looking up security policy databases (SPDs) and executing crypto-operation.

Hardware implementation of the IPSec may be a good solution. The network security processor (NSP) is an appropriate Internet security hardware solution. It integrates data transfer, NSP, and crypto-operation on the chip, and supports large network bandwidths. Based on the placement of the NSP in Internet devices, the NSP can be divided into look-aside mode and flow-through mode. In look-aside mode, the NSP is placed outside the main data path, and needs the network processor (NP) or CPU to allocate packets to it. In flow-through mode, the NSP is placed on the main data path and receives the packet itself.

Most of the previous work on NSP aimed at look-aside mode to improve the performance of the crypto-operation needed by the IPSec, which can be seen in Ha *et al*. (2004) and Wang *et al.* (2008), ignoring the protocol processing which has also been shown to be time consuming (Potlapally *et al*., 2006). In Wang *et al*. (2006) and Nishida *et al.* (2010), the IPSec protocol processing was presented, but the performance cannot satisfy the 10 Gbps data transfer. A kind of system on chip (SoC) architecture proposed by Ferrante and Piuri (2007) can achieve high IPSec processing performance, but it is simulated only at the function level. A new multi-core NP developed by some famous network equipment companies integrates security modules in the NP and can achieve very high speeds, but the cost is too high. A commercial product based on the flow-through architecture has been developed by Hifn (2008), allowing for 1 Gbps full duplex IPSec traffic processing and a processing capability of up to one million of packets per second. However, the conventional shared bus data transfer topology is used, with the disadvantage and bottleneck of tremendous arbitrations and only one data transfer path at a time. Thus, it cannot satisfy the requirement of 10 Gbps data transfer.

This paper deals with in-line NSP design of a 10 Gbps Ethernet. By adopting a configurable hetero-multi-core architecture, the whole IPSec protocol can be implemented and 10 Gbps data processing line speed realized. In the design, two different 64-bit configurable crossbar data transfer skeletons combined with the optimized high speed input/output data buffering mechanism maximize the homogenous and heterogeneous computation resources. A high speed low power hardware look-up method is proposed, which effectively reduces the layout area and power dissipation. New hardware structures of the function modules are dedicatedly designed and optimized to achieve high performance.

## 2 Overview of 10 Gbps in-line NSP architecture

The architecture of the 10 Gbps in-line NSP is illustrated in Fig. 1. It consists of a 10 Gbps serial port (SerDes), a framer, a configurable hetero-multi-core IPSec processor, a packet processor, and a packet analyzer, all of which together implement the whole IPSec protocol processing including crypto-operation, database query functionalities, and IPSec header processing.

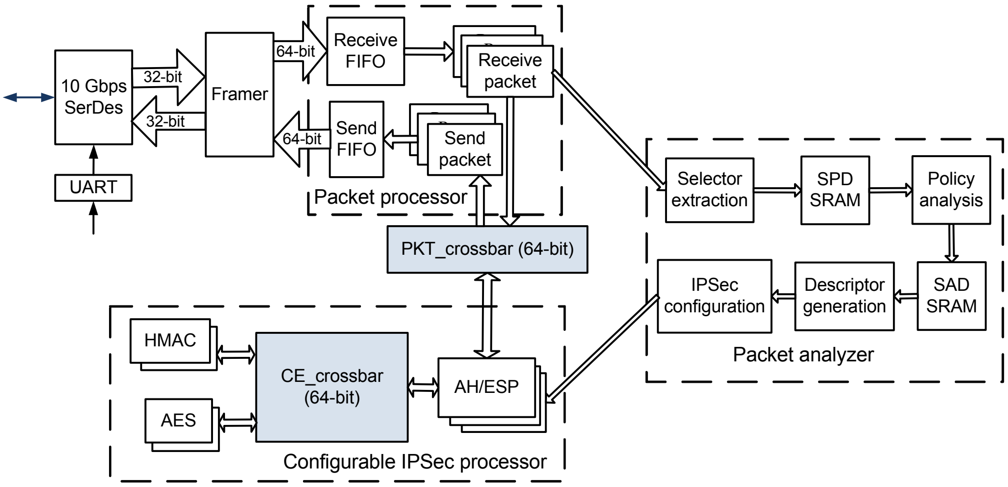


**Fig. 1 Architecture and data flow of 10 Gbps in-line network security processor (NSP)**

The look-aside method requires a packet to traverse the memory four times, which greatly degrades the performance of the NSP. The flow-through mode receives the packet and executes the packet security processing by itself, eliminating the bottleneck of communication between the NP and the NSP required in look-aside mode (Khan *et al.*, 2003). To achieve a 10 Gbps processing line speed, flow-through mode is adopted in the architecture.

Two different configurable 64-bit crossbar switch structures, which enable simultaneous data transfer between the heterogeneous multiple cores on the data paths, are proposed to maximize transfer efficiency and resource utilization. The modified hardware look-up module achieves high speed database look-up, small chip area, and low power dissipation of the chip.

By using a reasonable handshaking protocol between modules, and optimizing the frame input and output buffer mechanism, the size and number of on-chip FIFOs are greatly reduced, which significantly reduces the layout area and power dissipation.

The data flow of the design is described as follows: the high speed serial data from 10 Gbps Ethernet is received by the 10 Gbps SerDes and converted into 32-bit parallel data. The framer encapsulates the received data into frames based on the 10 Gbps Ethernet data link layer protocol, and then buffers them into the packet processor. In the packet processor, frame headers are removed and stored in the registers, and an interrupt signal will be generated. The packet analyzer then reads the IP header and extracts the selector, and looks up the on-chip SPD SRAM, which will decide the action to the packet. If the IP packet needs to perform IPSec processing, a security association database (SAD) is queried and a task descriptor, which includes operation mode, IP packet address, key address, and other information, is generated. The packet analyzer reads and analyzes the descriptor, inquires the status of the IPSec processor, and dispatches tasks to the IPSec processor. The IPSec processor reads in the IP packet that needs to be processed from the packet processor and invokes the security algorithm modules to do crypto-operation. After the IPSec processor finishes packet processing, the packet processor encapsulates the packet with the frame header stored previously and sends the processed frame out through the framer and the SerDes to the 10 Gbps Ethernet.

A five-stage pipeline is implemented through the data flow path to increase the data throughput.

## 3 High performance data communication mechanism

To achieve 10 Gbps line speed data processing, multiple heterogeneous and homogenous modules must be paralleled in the design. Time consumed on processing the packets by these modules varies with packet length. Thus, data exchange and transfer among these heterogeneous and homogenous modules become sophisticated.

The bottleneck of the conventional shared bus transfer structure and a dual one-way 64-bit data shared bus structure proposed by Wang HX *et al.* (2010) is that only one data transfer path can be set up at the same time; hence, the source utilization and data transfer efficiency become very low, and the performance is severely degraded. A high performance data communication mechanism is required to maximize the use of computing resources in the design to achieve 10 Gbps line speed data processing.

The crossbar switch uses a scheduler to allocate resources and provides multiple independent paths for concurrent data transfer between the heterogeneous and homogenous modules, which greatly improves the transfer efficiency. Also, the number of ports can be configured to meet the requirements of different applications. Thus, the crossbar switch structure is adopted for multi-core data transfer in the design. Two 64-bit crossbars, PKT_crossbar and CE_crossbar, are used (Fig. 1). PKT_crossbar is connected between the packet processor and the IPSec processor, and CE_crossbar is used in the IPSec processor between AH/ESP cores and AES/HMAC-SHA-1 cores. Based on their data flow characteristics, two different crossbar structures and scheduling mechanisms are proposed.

One side of PKT_crossbar interfaces the high speed frame receiving and transmitting modules, and the other side connects the IPSec processor, which consumes much more time to process packets. For this condition, packet loss will occur at 10 Gbps line speed. A two-stage buffer mechanism (Fig. 2) is implemented to solve the problem.

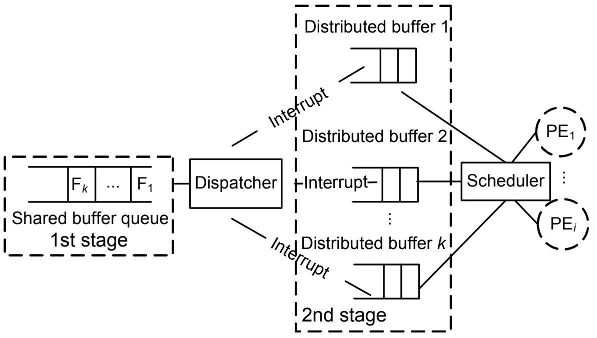


**Fig. 2 PKT_crossbar structure**

In the first stage, a shared buffer serves as a big pool to buffer the high speed data from the 10 Gbps Ethernet. In the second stage, buffers are distributed to each PKT_crossbar port, which means multiple packets (the packet number depends on the port number) in the 1st-stage shared buffer can be handled simultaneously. A dynamic load allocation algorithm based on interrupt and a round-robin scheduling algorithm are used in the dispatcher and the scheduler, respectively, to balance the load distribution. The dispatcher controls the packet allocation using the dynamic load allocation algorithm: when the packet is processed and transmitted out by the IPSec processing engine (PE$_i$ in Fig. 2) through the scheduler, the corresponding distributed buffer generates an interrupt signal; the dispatcher can read the next packet from the 1st-stage shared buffer based on this interrupt signal. The distributed buffers can store only one longest packet permitted by the IP layer protocol. The size of the shared buffer is chosen considering the worst case; that is, all the PEs are busy. So, the size of the 1st-stage buffer is the product of the distributed buffer and the maximum port number of the PKT_crossbar. With this interrupt mechanism, the design has the capacity to process the variable size packets. By simulation, if the PKT_crossbar is configured to 6×16, the zero packet loss ratio is obtained.

For the CE_crossbar, the load balancing and scheduling efficiency are the key factors for maximizing computing resources, because multiple heterogeneous functional modules must be paralleled to achieve 10 Gbps data processing line speed. The iSLIP (McKeown, 1999) scheduling algorithm, which includes three phases—request, grant, and accept, is adopted in the design. The delay between the grant and accept arbiters directly affects the speed of data transfer. Fig. 3a illustrates an arbiter design (Pape, 2006) chosen for this chip implementation. The arbiter is based on a simple round-robin arbiter. It includes an Update_enable signal to allow the iSLIP algorithm to update the priority only under certain circumstances.

The priority process engine (PPE) shown in Fig. 3b combines two simple process engines (PEs). The input signal P_enc to Simple_PE_thermo is masked by a thermometer encoding (Gupta and McKeown, 1999), in which an $n$-bit-wide vector $\boldsymbol{y}$ satisfies $y[i]=1$ if and only if $i<X$ for all $0\leq i<n$, where $X$ is the decimal value of a $\log_2 n$-bit vector $\boldsymbol{x}$ and $0\leq X<n$. The non-masked Simple_PE does not take

any priority, and determines the output when the Simple_PE_thermo does not find any 1-input between the programmed priority and the input.

With the data transfer mechanism described above, the data transfer efficiency reaches 86.6% in the design.

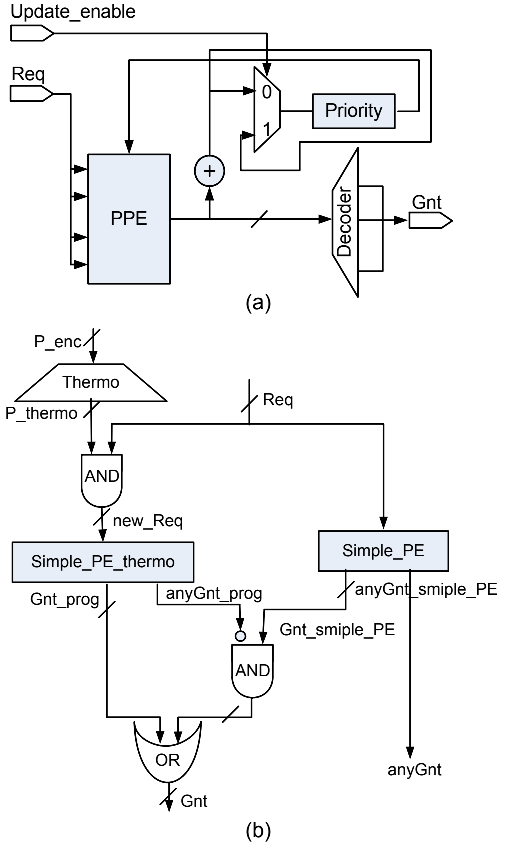


**Fig. 3 Arbiter diagram (a) and priority process engine (PPE) structure (b)**

## 4 High speed database look-up design

In IPSec processing, the SPD database query is performed on every packet from the 10 Gbps Ethernet to obtain the security policy, and the SAD is looked up if the packet needs to be IPSec processed. Thus, the speed of database look-up directly affects the performance of the whole design.

The SPD look-up can be seen as a special pattern matching to decide how to handle the IP packets by matching a selector extracted from some fields in the IP header. Typically, the selector is a 144-bit string composed of five fields in the IP header, which are the IP destination address, IP source address, IP protocol, destination port, and source port. Choosing a proper pattern matching algorithm is a key step. The hash-based algorithm (Cho and Mangione-Smith, 2005) and the ternary content addressable memories (TCAM) based algorithm (Liu *et al.*, 2010) are the existing hardware approaches. The throughput of the first algorithm cannot easily satisfy 10 Gbps look-up. Using TCAMs can achieve 10 Gbps throughput, but some issues such as high power consumption (about 2 to 3 times more than SRAMs (Fang *et al.*, 2008)) and large area make TCAM not a good choice. A cache plus off-chip memory mechanism was adopted in Ferrante *et al.* (2007) to achieve high throughput, but the caches used are also the CAMs.

In this design, we assume that all the packets from the same IP address and the TCP/UDP port apply the same security policy. For this condition, we choose a coarser 80-bit selector, which includes only three fields of the IP header: the IP source address, the IP destination address, and the source port in the allowable scope of the IPSec protocol. We improve the hash algorithm as follows:

1. The 80-bit selector is compressed to an *n*-bit vector (*n*<<80) by a given hash function.

2. The *n*-bit hash result is directly taken as the address input of the SPD SRAM.

3. The output of the SPD SRAM is the security policy for the packet.

4. If the packet needs to do IPsec processing, the SAD SRAM input address is read directly from the security policy.

5. The output of the SAD SRAM is the security association (SA) of the packet.

6. Analyze the SA and generate a descriptor for the IPSec processor.

The improved algorithm uses only two SRAMs on-chip, which reduces the read access time from off-chip memory (Ferrante *et al.*, 2007) and improves the throughput of the SPD look-up. Thus, the cache mechanism on chip is not needed any more. This method also reduces the power and area of the chip. From step 2, we can infer that the maximum number of security policies is $2^n$. Thus, the security policy number can be configured by choosing $n$, and this improves the flexibility of the design.

Cyclic redundancy check (CRC), which has proved a good hash function (Jain, 1992), is used in this design. By using CRC, the width of the compression can be chosen. And, CRC performance is high by parallel hardware implementation.

We also redesign the data format of SPD. The new data structure includes not only the strategies for the packet, but also the SAD address. If the packet needs to do IPSec, the SAD address is read directly from the SPD entry. By simulation, although the throughput of the new algorithm is slightly lower than that of the TCAM algorithm, it still fulfills the need of 10 Gbps look-up. We verify the improved algorithm on FPGA and compare it with the other two algorithms (Table 1).

**Table 1 Comparison of the three algorithms**

| Hardware algorithm | Throughput (Gbps) | Slice LUTs used (%) | BlockRAMs used (%) | Estimated power (W) |
|---|---|---|---|---|
| Hash based algorithm | 2 | NA | NA | NA |
| TCAM based algorithm | 15 | 84 | 58 | 1.680 |
| This work | 11.9 | 82 | 25 | 0.688 |

As can be seen, the number of BlockRAMs used in FPGA and power dissipation are greatly reduced. There is, however, a potential problem that SPD and SAD must be initialized when the NSP system starts up and cannot be updated while the system is working.

## 5 Configurable IPSec processor

The IPSec protocol processor includes many paralleled IPSec processing and crypto-operation modules. All these modules are connected to the CE_crossbar mentioned above, which forms a hetero-multi-core structure. The port number of the CE_crossbar can be configured and optimized to achieve 10 Gbps throughput. The interfaces of these modules are unified for reuse of the IP cores.

To improve data throughput, we study the feature of IPSec processing in the crypto and protocol modules and divide the whole data path into three stages: protocol pre-processing, data crypto-operation, and protocol post-processing. Two FIFOs are inserted before and after the crypto-operation stage, respectively, which forms a three-stage pipeline in the data path.

For configurability, the AH and ESP protocols are designed separately. The functions of the AH/ESP modules include: (1) modifying the original IP header and generating a new IP header, (2) generating and removing the AH/ESP header based on the operating mode, and (3) invoking the corresponding cryptographic algorithm core to encrypt, decrypt, or verify the packet. A four-stage pipeline is used to improve the IP header processing performance (Wang L *et al.*, 2010).

Meanwhile, the hardware performance and interface consistency are considered in the design and integration of the crypto-operation modules. In the IPSec protocol, the required security algorithms include the block cipher and authentication algorithm. In this design, we choose the Advanced Encryption Standard (AES) as the block cipher, and HMAC-SHA-1 as the message authentication algorithm, which are both standard algorithms required in RFC2401:1998. Other security algorithms such as SHA-2 can also be used because the IPSec processor has the configurable characteristics, but the area of the hardware implementation of SHA-2 is larger than that of SHA-1. Thus, SHA-1 is chosen in the design.

An ultra high throughput, low power consumption HMAC-SHA-1 hardware design was proposed by Liu *et al.* (2012) after analyzing the basic SHA-1 algorithm. By adopting a modified algorithm and a two-stage cascade hardware structure, the operation time for one authentication is reduced from 80 to 40 clock cycles and the hardware cost is only five 32-bit registers.

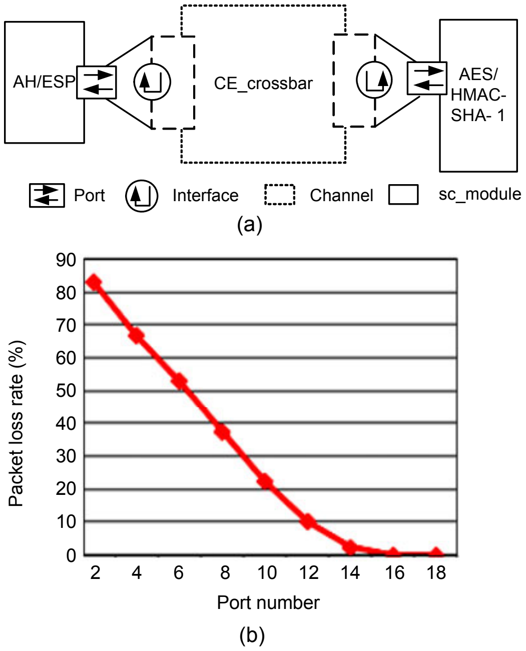
The AES crypto hardware design is based on our previous work (Wu *et al.*, 2009). The AES crypto-core has a block size of 128 bits and key length of 128 bits.

All the modules mentioned above are hardware implemented using Verilog Hardware Design Language. The simulation and synthesis results at 250 MHz clock frequency based on 65 nm CMOS technology are shown in Table 2.

A transaction level simulation model based on the SystemC language (Fig. 4a) is implemented. The basic components in the SystemC are module, channel, interface, and port. In this model, the functions of AH/ESP and AES/HMAC-SHA-1 are implemented through different modules: the CE_crossbar is implemented by the channel, the connections between the modules are realized through ports and interfaces, and the communication between the modules is accomplished by the channel. The number of modules can be configured to achieve optimum performance.

**Table 2 Simulation and synthesis results of the modules**

| Module | Gate count | Estimated power (mW) | Throughput (Gbps) |
|---|---|---|---|
| AH | 32 886 | 6.79 | 0.91 |
| ESP | 39 194 | 9.27 | 0.61 |
| HMAC-SHA-1 | 22 575 | 4.80 | 2.95 |
| AES-128 | 62 289 | 9.20 | 3.63 |



(a)

(b)

**Fig. 4 Configurable IPSec processor modeling**
(a) Model diagram; (b) Simulation result

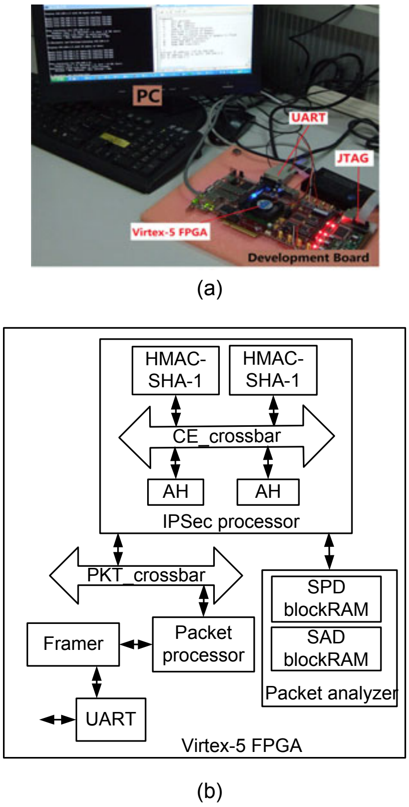
By configuring the port number of the CE_crossbar and simulation, the result is that, with 16 AH/ESP cores and 16 AES/HMAC-SHA-1 cores connected to the 16×16 CE_crossbar, the packet loss rate decreases to zero (Fig. 4b); under this condition, the IPSec processor is thought to achieve the capability of 10 Gbps in-line data processing.

## 6 Verification

Verification of the design includes FPGA prototyping verification and post simulation by electronic design automation (EDA) tools before the design is fabricated. A test bench is being set up to verify the function and performance of the 10 Gbps in-line NSP chip.

### 6.1 FPGA prototyping

To verify the proposed design, an FPGA prototyping platform is constructed (Fig. 5a). It includes a Xilinx development board and a PC. The PC acts as the terminal of data transmission. Data transmission between PC and the board is done using the UART port. The development board includes a Virtex-5 XC5VSX95T FPGA chip.



(a)

(b)

**Fig. 5 FPGA verification platform**
(a) Platform photography; (b) Design on the FPGA platform

To test the function correctness of the design, typical test data sizes of 64, 512, and 928 bytes are chosen. The test data is transferred and received by PC through the UART port and compared.

First, the configurable hetero-multi-core IPSec processor on the platform is verified. Due to the content limitation of the FPGA, the IPSec processor is configured first with four AH cores and four HMAC-

SHA-1 cores, and then with four ESP cores, two HMAC-SHA-1 cores, and two AES cores. The implementation results are demonstrated in Table 3.

**Table 3 Implementation results of configurable IPSec processor FPGA**

| Modules verified | Slice LUTs utilization | BlockRAMs utilization | Registers utilization |
|---|---|---|---|
| AH×4, HMAC-SHA-1×4 | 47% | 7% | 30% |
| ESP×4, AES×2, HMAC-SHA-1×2 | 52% | 16% | 47% |

Tests of AH transport mode and ESP transport and tunnel mode are performed separately at 100 MHz clock rate, and the correctness of the configurable IPSec processor is verified.

Then the whole proposed design is verified on the FPGA platform. As depicted in Fig. 5b, the IPSec processor is configured with two AH cores and two HMAC-SHA-1 cores connected to the 2×2 CE_ crossbar. The PK_crossbar is also configured to 2×2. The packet analyzer, the SPD and SAD memories, the framer, the packet processor, and the UART controller are included in the architecture. The 10 Gbps SerDes is not included because it is an analog module. Table 4 lists the FPGA implementation results.

**Table 4 Implementation results of 10 Gbps in-line NSP FPGA**

| FPGA source | Number | | Utilization |
|---|---|---|---|
| | Used | Available | |
| Slice LUTs | 47 597 | 58 880 | 81% |
| Slice registers | 18 727 | 58 880 | 32% |
| Logic | 34 133 | 58 880 | 58% |
| BlockRAMs | 61 | 244 | 25% |

The AH protocol in transport and tunnel mode is verified at 100 MHz clock frequency.

Based on the FPGA verification results, the proportion of main processing steps in AH processing is obtained (Table 5) and compared with Potlapally *et al.* (2006)'s design. It is indicated that after taking the methods and the architecture proposed above, the performances of IPSec protocol processing and SPD look-up greatly outperform those of Potlapally *et al.* (2006)'s design.

**Table 5 Function module performance comparison**

| Reference | Proportion of main processing steps in AH | | |
|---|---|---|---|
| | IPSec processing | SPD look-up | Crypto operation |
| Potlapally *et al.* (2006) | 33.7% | 6.8% | 52% |
| This work | 12% | 2% | 70% |

## 6.2 Post simulation

The 10 Gbps in-line NSP chip is implemented with 65 nm CMOS technology and taped out. To reduce the total power consumption, a low leakage low power 65 nm standard cell library is adopted. Fig. 6 shows the layout view of the 10 Gbps in-line NSP, the size of which is 2.5 mm×3 mm, including about 360 million equivalent logic gates.

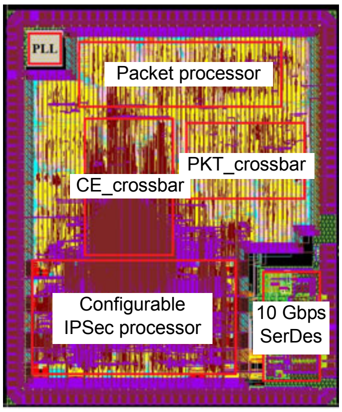


**Fig. 6 Layout view of the 10 Gbps in-line NSP**

The 10 Gbps SerDes module is included in the 10 Gbps in-line NSP, and the configurable IPSec processor includes 16 AH cores, 16 HMAC-SHA-1 cores, and a 16×16 CE_crossbar. The PKT_crossbar is configured to 6×16. The phase locking loop (PLL) module generates the clock needed in the design.

There are three clock domains in NSP: the 312.5 MHz clock from the 10 Gbps SerDes, the 156.25 MHz framer clock, and the 250 MHz core clock. Two cascaded registers are inserted between the clock domains on the chip to synchronize the processing data.

In the post-simulation, the 10 Gbps serial test vectors are generated by software, and the test packet length is 64, 512, or 928 bytes. The test data obeys the Poisson normal distribution. The throughput of the design is computed as follows:

$$\text{Throughput} = \frac{\text{Packetbits} \times f}{\text{CycleNumber}}, \tag{1}$$

where Packetbits is the number of all the valid packet bits sent to the design in CycleNumber cycles, and $f$ is the clock frequency. The post-simulation shows that the throughput of AH transport mode processing can achieve up to 10.06 Gbps with the average test packet length of 512 bytes under the clock frequency of 250 MHz. The power dissipation is about 0.868 W.

### 6.3 Test bench

To perform a whole test to the 10 Gbps in-line NSP in a real application environment, a test bench is designed (Fig. 7a). It includes a test board and a sandy bridge main board provided by the Inspur Co., Ltd. Fig. 7b is the test board, which includes the 10 Gbps in-line NSP test chip, the Quad Small Form-factor Pluggable (QSFP) port, the sub-miniature-A (SMA) port for clock, the UART port for connection with the PC to configure the 10 Gpbs in-line NSP chip, the power module, the clock oscillator, and the reset button. These two boards are connected by an optical fiber.

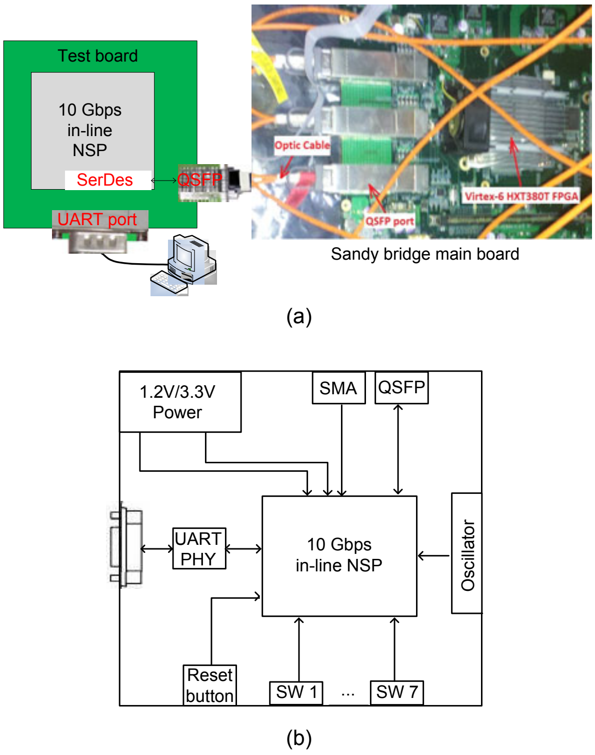


**Fig. 7 Diagrams of the test bench (a) and test board (b)**

The test frames of the 10 Gbps media access control (MAC) layer are generated using the data link layer (DLL) protocol, Aurora, supported by Xilinx and transmitted by the SerDes in the Virtex-6 HXT380T FPGA. The 10 Gbps serial data is transmitted through the QSFP port from the sandy bridge main board. On the test board, another QSFP port is mounted and connected to the 10 Gbps in-line NSP chip to receive the test data. The test data processed by the 10 Gbps in-line NSP is transmitted back to the main board and verified.

Currently, the test bench is being set up and the test vector generation program is coded.

## 7 Conclusions

A 10 Gbps in-line network security processor design used in the 10 Gbps Ethernet is described. The 10 Gbps high speed data transfer, the IPSec protocol processing, and crypto operations are accomplished in the design. The design is implemented with 65 nm CMOS technology and the layout size is 2.5 mm× 3 mm with 360 million logic gates.

Both the FPGA prototype verification and the post-simulation are performed to verify the proposed design. The functions of the configurable hetero-multi-core IPSec processor and the proposed design are verified on an FPGA platform. The post-simulation demonstrated that the design gives a peak throughput for the AH transport mode of 10.06 Gbps at an average packet length of 512 bytes under the clock rate of 250 MHz, with a power dissipation of 0.868 W. Thus, the design can be widely employed in VPN under 10 Gbps Ethernet environments. Also, the design can be used in the next generation network-based security equipment at 40/100 Gbps for its configurability.

The test bench is being set up and the test vector generation program is coded to test the 10 Gbps in-line NSP in a real application environment.

## References

Chen, Z.H., 2011. Research on Pattern Matching Algorithm in 40Gbps Application Awareness System. MS Thesis, PLA Information Engineering University, Zhengzhou, China (in Chinese).

Cho, Y.H., Mangione-Smith, W.H., 2005. Fast Reconfiguring Deep Packet for 1+ Gigabit Network. Proc. 13th Annual IEEE Symp. on Field Programmable Custom Computing Machine, p.215-224. [doi:10.1109/FCCM.2005.34]

Fang, Y.T., Huang, T.C., Wang, P.C., 2008. Ternary CAM Compaction for IP Address Lookup. 22nd Int. Conf. on Advanced Information Networking and Applications, p.1462-1467. [doi:10.1109/WAINA.2008.168]

Ferrante, A., Piuri, V., 2007. High-Level Architecture of an IPSec-Dedicated System on Chip. 3rd EuroNGI Conf. on Next Generation Internet Networks, p.159-166. [doi:10.1109/NGI.2007.371211]

Ferrante, A., Piuri, V., Owen, J., 2005. IPSec Hardware Resource Requirements Evaluation. Next Generation Internet Networks, p.240-246. [doi:10.1109/NGI.2005.1431672]

Ferrante, A., Satish, C., Piuri, V., 2007. IPSec Database Query Acceleration. 4th Int. Conf. on E-Business and Telecommunications, p.188-200.

Gupta, P., McKeown, N., 1999. Designing and implementing a fast crossbar scheduler. *IEEE Micro*, **19**(1):20-28. [doi:10.1109/40.748793]

Ha, C.S., Lee, J.H., Leem, D.S., 2004. ASIC Design of IPSec Hardware Accelerator for Network Security. IEEE Asia-Pacific Conf. on Advanced System Integrated Circuits, p.168-171.

Hifn, 2008. Flow Through Security Processor. Available from http://www.acaltechnology.com/_files/legacy_news/HifnPB-9150-5.pdf

IEEE Std 802.3-2012. IEEE Standard for Ethernet. IEEE Computer Society, NY, USA.

Jain, R., 1992. A comparison of hashing schemes for address lookup in computer networks. *IEEE Trans. Commun.*, **40**(10):1570-1573. [doi:10.1109/26.168785]

Khan, E., El-Kharashi, M.W., Rafiq, A.N.M.E., Gebali, F., Abd-El-Barr, M., 2003. Network Processors for Communication Security: a Review. IEEE Pacific Rim Conf. on Communications Computers and Signal Processing, p.173-176.

Liu, A.X., Meiners, C.R., Torng, E., 2010. TCAM razor: a systematic approach towards minimizing packet classifiers in TCAMs. *IEEE/ACM Trans. Network.*, **18**(2):490-500. [doi:10.1109/TNET.2009.2030188]

Liu, Y., Wu, L.J., Niu, Y., Zhang, X.M., Gao, Z.Q., 2012. A High-Speed SHA-1 IP Core for 10 Gbps Ethernet Security Processor. 8th Int. Conf. on Computational Intelligence and Security, p.237-241. [doi:10.1109/CIS.2012.60]

McKeown, N., 1999. iSLIP scheduling algorithm for input-queued switches. *IEEE/ACM Trans. Network.*, **7**(2):188-201. [doi:10.1109/90.769767]

Nishida, Y., Kawai, K., Koike, K., 2010. A 2Gbs Network Processor with a 24mW IPsec Offload for Residential Gateways. IEEE Int. Solid-State Circuits Conf., p.280-281. [doi:10.1109/ISSCC.2010.5433917]

Pape, J.D., 2006. Implementation of an On-Chip Interconnect Using the i-SLIP Scheduling Algorithm. MS Thesis, the University of Texas, Austin, USA.

Potlapally, N.R., Ravi, S., Raghunalhan, A., Lee, R.B., Jha, N.K., 2006. Impact of Configurability and Extensibility on IPSec Protocol Execution on Embedded Processors. 19th Int. Conf. on VLSI Design, p.299-304. [doi:10.1109/VLSID.2006.102]

RFC2401:1998. Security Architecture for the Internet Protocol. Internet Engineering Task Force (IETF), Washington D.C., USA.

Wang, C.H., Lo, C.Y., Lee, M.S., Yeh, J.C., Huang, C.T., Wu, C.W., Huang, S.Y., 2006. A Network Security Processor Design Based on an Integrated SOC Design and Test Platform. Proc. 43rd Annual Design Automation Conf., p.490-495. [doi:10.1145/1146909.1147039]

Wang, H.X., Bai, G.Q., Chen, H.Y., 2008. Zodiac: System Architecture Implementation for a High-Performance Network Security Processor. IEEE 19th Int. Conf. on Application-Specific Systems, Architectures and Processors, p.91-96. [doi:10.1109/ASAP.2008.4580160]

Wang, H.X., Bai, G.Q., Chen, H.Y., 2010. Design and implementation of a high performance network security processor. *Int. J. Electron.*, **97**(3):309-325. [doi:10.1080/00207210903289383]

Wang, L., Niu, Y., Wu, L.J., Zhang, X.M., 2010. Design of an IPSec IP-Core for 10 Gigabit Ethernet Security Processor. Proc. 10th IEEE Int. Conf. on Solid-State and Integrated Circuit Technology, p.539-541. [doi:10.1109/ICSICT.2010.5667343]

Wu, L.J., Ji, Y.J., Zhang, X.M., Li, X.Y., Yang, Y.S., 2009. Power analysis resistant AES crypto engine design for a network security co-processor. *J. Tsinghua Univ. (Sci. Tech.)*, **49**(S2):2097-2102 (in Chinese).