

# IMPLEMENTATION OF CRYPTOGRAPHIC ALGORITHM AS A NETWORK PROCESSOR ELEMENT

Prof A.I Trivedi, Prof S.K. Shah

Kiran Parmar, Mihir Shah,

Electrical Engineering Dept.

Faculty of Technology & Engineering, M.S.University,Baroda

[krparmar@indiatimes.com](mailto:krparmar@indiatimes.com) [shah\\_mv@indiatimes.com](mailto:shah_mv@indiatimes.com)

## ABSTRACT

Network processors are specialized CPUs optimized to support the implementation of network protocols at wire line speed and will become critical component of next-generation networking equipment. The vulnerability of Internet from attacks by intruders and hackers, as well as expansion of private networks over the Internet has made the security aspects as a key element in designing Network Processor architecture. Thus combinations of cryptographic applications and communication applications have become important issues of network processor design.

In this paper, we discuss the architectural issues of network processor in general and the same for cryptographic applications in particular. The most important criteria of selection of cryptographic algorithm and its implementation depend on the speed, security of the packets to be processed by the NPU (Network Processor Unit).

## 1. INTRODUCTION

Existing designs have added security to the network through either a co-processor or an inline security processor. As data rates go up, the co-processor solution becomes less and less practical [6]. Inline security processors can actually scale to the higher data rates but must perform many of the same functions as the network processor. Currently available NPU & Cryptographic processors use DES, 3DES & AES algorithms.

An Architectural analysis of different cryptographic algorithms suggest that IDEA (International Data Encryption Algorithm) is one of the best & more secure block ciphers available today [4]. In this paper we suggest to implement IDEA as a core algorithm in the inline security engine of the NPU.

A collection of cryptography applications such as secure IP (IPsec) and Virtual Private Networks (VPNs) has been widely deployed in both routers and end systems. Security related applications are all computational intensive applications that can consume as much as 95 percent of an application server's processing capacity [4]. As the demands for secure communication grow, cryptographic processing may become a bottleneck to the system performance.

Network traffic, consists of packets from thousands of different flows that each requires relatively little processing. Speed of a Network Processor is a problem because the bandwidth of Optical fiber is growing at an even faster rate than the speed of silicon. It becomes evident from the following points:

- DWDM is only just begun to open up the WAN backbone and MAN trunks.
- DSL, Cable Modems, B-band wireless, and cheaper fiber-based access solutions are finally opening up WAN access, which has been the toughest bottleneck of all.
- In 5 years processing power will increase 8x (doubling 3 times).
- At the present growth rate (doubling every four months), aggregate Internet bandwidth will increase 32,768x. [7]

Compared to custom designed ASICs, Network Processors will shorten development cycles and enable higher value features such as QoS and policy based networking. However, they are targeted not only for packet processing applications. As demands for communication security grow, cryptographic processing becomes another type of application domain.

The fastest security processor can achieve the speed of few Gbps, however the impact of encryption and decryption functions performed

on network processors is still an area of research. Little research has been conducted on the architectural requirements of cryptographic applications for network processor designs. This has motivated us to consider the architectural issues of network processor along with cryptographic application.

## 2. GENERAL ARCHITECTURAL ISSUES OF NPU

In such architecture, a core processor manages complex global tasks, while multiple low-level processors called *microengines* perform the packet-processing operations. The various architectural issues are as follows:

1. Creating the infrastructure that enables the microengines of the processor to communicate with each other.
2. Distributing the control and packet-handling responsibilities across a core processor and multiple micro engines in a manner that provides high performance.
3. The process of determining the *optimal hardware and software co-design* for such processors is faced with issues involving resource allocation and partitioning.
4. The architecture design should take into account different *packet processing* functions, task scheduling options, information about the packet forms, and the QoS guarantees; that the processor should be able to meet.
5. Efficient decomposability of tasks with well-defined and clean interaction for parallel processing.
6. Open interfaces for ease of networking hardware and software/OS porting (CSIX, PoS Phy, and CPIX)
7. Trade-offs between parallel engines, context switched multithreading, Multiple parallel instructions (VLIW) architectures.

### 2.1 NETWORK PROCESSOR FUNCTIONS

*Classification:* Looking inside the packet to determine destination and any special processing requirements.

*Modification:* Changing the contents of the packet, for example, doing encryption or security processing.

*Queuing:* Assigning the packet to a queue (specifying priority) for presentation to the fabric.

*Management and Control:* Managing the process, dealing with exceptions, talking to control elements of the switch.

## 3. ISSUES FOR INCORPORATING CRYPTOGRAPHIC APPLICATION FOR NPU

- When designing a network security product, one must consider both the packet-processing requirements and the security requirements.
- Adding security functionality into the silicon area of network processor while maintaining wire rate and minimizing new silicon area.
- The securing of network traffic is portioned into two partitions: protocol processing and cryptographic algorithm processing. [1]
  - a) Protocol processing would include Encapsulating Security Protocol (ESP), Authentication Header (AH), Secure Sockets Layer (SSL), Transport Layer Security (TLS), and other non-security protocols such as Transmission Control Protocol (TCP) and the Internet Protocol (IP) processing.
  - b) Cryptographic algorithm processing includes the data manipulation that would need to be done on the entirety of the payloads, such as confidentiality and integrity.
- The cryptography unit may be comprised of several algorithms that in conjunction must provide data confidentiality and data integrity. Each algorithm has its own set of trade-offs and challenges, in terms of silicon area, parallelism, and symmetry.
- The added security functionality may support the Data Encryption Standard (DES), 3DES, and the Advanced Encryption Standard (AES), IDEA algorithms along with the Secure Hash Algorithm (SHA-1) for data authentication directly in hardware.

## 4. IMPLEMENTATION OF IDEA

We have implemented encryption of IDEA (International Data Encryption Algorithm) using ALTERA's Development tools.

IDEA is a block cipher that uses a 128-bit key to encrypt data in blocks of 64 bits. By contrast, DES also uses 64-bit blocks but a 56-bit key.

*Implementation Considerations:* IDEA is designed to facilitate both software and hardware implementation. Hardware implementation, typically in VLSI, is designed to achieve high speed. Design principles for hardware implementation are as follows:

- (i) Similarity of encryption and decryption: Encryption and decryption should differ only in the way of using the key so that the same device can be used for both encryption and decryption. IDEA has a structure that can satisfy this requirement.
- (ii) Regular structure: The cipher should have a regular modular structure to facilitate VLSI implementation. IDEA is constructed from two basic modular building blocks repeated multiple times.

IDEA encryption: The single round of IDEA encryption is shown in the *Figure 1*. There are two inputs to the encryption function: I ) the plaintext to be encrypted II) the key. Plaintext is 64 bits in length and the key is 128 bits in length in this particular case. The 64 bit data block is divided into four 16-bit sub blocks: X1, X2, X3 and X4. These four sub blocks become input to the first round of algorithm.

The IDEA algorithm consists of eight rounds followed by a final transformation. In each round, the sequence of events is as follows:

1. Multiply X1 and the first subkey
2. Add X2 and the second subkey
3. Add X3 and the third subkey
4. Multiply X4 and the fourth subkey
5. XOR the results of steps (1) and (3)
6. XOR the results of steps (2) and (4)
7. Multiply the results of step (5) with the fifth subkey
8. Add the results of steps (6) and (7)
9. Multiply the results of step (8) with the sixth subkey
10. Add the results of steps (7) and (9)
11. XOR the results of steps (1) and (9)
12. XOR the results of steps (3) and (9)
13. XOR the results of steps (2) and (10)
14. XOR the results of steps (4) and (10)

The output of the round is the four subblocks that are results of steps (11) to (14). Swap the two inner blocks (except for the last round) and that's the input to the next round. After the eighth round, there is final output transformation as shown in figure 2 :

- (i) Multiply W81 and the first subkey Z49
- (ii) Add W82 and the second subkey Z50
- (iii) Add W83 and the third subkey Z51
- (iv) Multiply W84 and the fourth subkey Z52

Finally, the four sub blocks are reattached to produce cipher text.

Creating the subkeys is easier. The algorithm uses 52 of them (six for each of the eight rounds and four more for the output transformation.) First, the 128-bit key is divided into eight 16-bit subkeys. These are first eight subkeys for the algorithm. Then the key is rotated 25 bits to the left and again divided into eight subkeys. The first four are used in round 2; the last four are used in round 3. The key is rotated another 25 bits to the left for the next eight subkeys, and so on until the end of the algorithm.

The IDEA algorithm was implemented as shown in *Figure 3*, using ALTERA's MAX PLUS software for VLSI synthesis. The major blocks implemented are

- (i) 16 bit XOR: Bit – by--Bit exclusive-OR, denoted as  $\oplus$ .
- (ii) 16-bit adder: Addition of integers modulo  $2^{16}$  (modulo 65536), with inputs and outputs treated as unsigned 16-bit integers. This operation is denoted as  $\boxplus$ .
- (iii) Modulo 16 multiplier: Multiplication of integers modulo  $2^{16}+1$  (modulo 65537), with inputs and outputs treated as unsigned 16-bit integers, except that a block of all zeros is treated as representing  $2^{16}$ .

This operation is denoted as  $\odot$ .

Initially, one round was simulated and tested with schematic waveform files on Altera .We plan to implement all eight rounds for which the source code is already written.

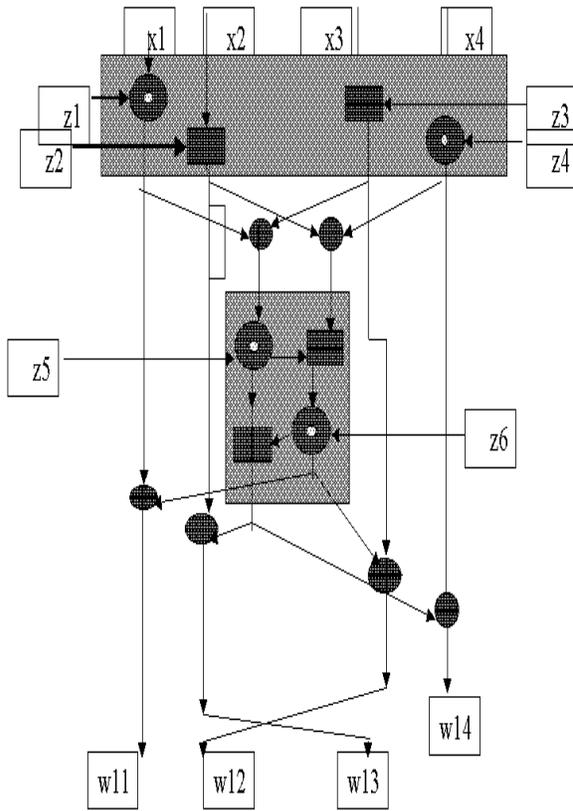


Figure 1: Single Round of IDEA

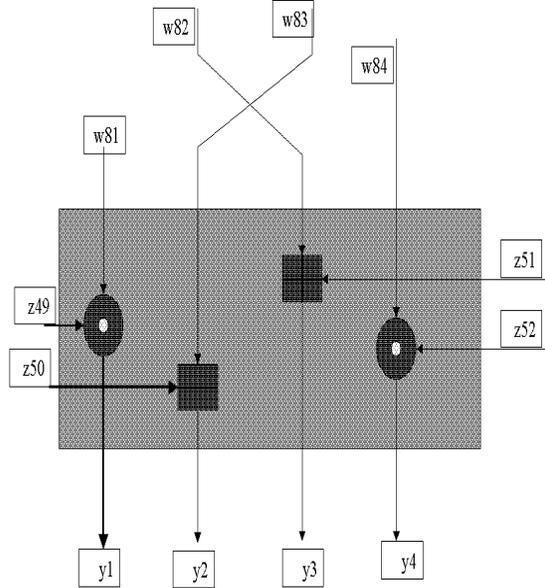


Figure 2: Output Transformation stage of IDEA

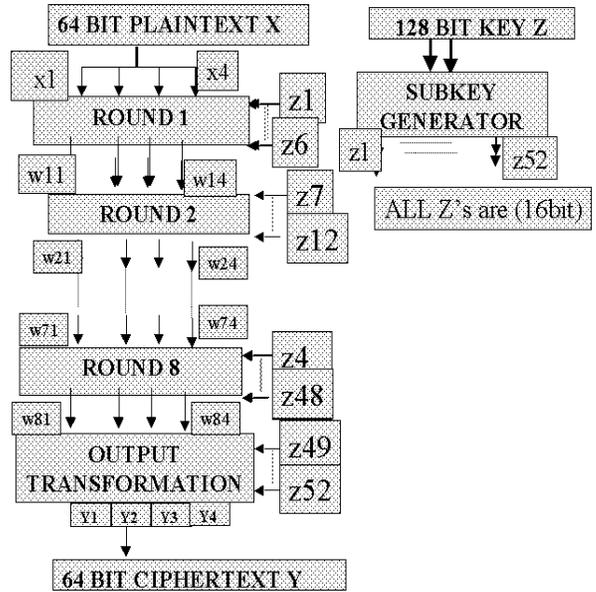


Figure 3: Overall IDEA Structure

The results for one round are as follows:

**Device: EPF10K130EFC484-1**

Total dedicated input pins used:	6/6	(100%)
Total I/O pins used:	219/363	(60%)
Total logic cells used:	5180/6656	(77%)
Total embedded cells used:	0/256	(0%)
Total EABs used:	0/16	(0%)
Average fan-in:	3.31/4	(82%)
Total fan-in:	17151/26624	(64%)
Total input pins required:	161	
Total input I/O cell registers required:	0	
Total output pins required:	64	
Synthesized logic cells:	188/6656	(2%)

**5. INTERFACING CONSIDERATION**

This block, can be interfaced with NPU in three different ways:

1. Security Co-processor coupled with NPU using Look aside Interface.
2. Designing Inline security processor to achieve high data rates.
3. Adding encryption circuitry into the same silicon as the network processor,

As a first phase in designing of security engine, we would like to implement this as a coprocessor working on IDEA algorithm and interfaced with NPU using the Look aside Interface.

The Hardware Working group of Network Processor Forum will be continuously validating this Look-Aside Interface during the development of the Open Interface standard. [8] This encryption co-processor operates using a request/response model and the LA-1 specification. Then based on the system performance and evaluation we may go for an inline implementation of the same.

## 6. RELATED WORK

Both Software and Hardware implementations of IDEA have been done as per following details:

Implementation	Performance	Reference
Software on 33 MHZ 386 machine	880 kbps	2
Software on 66 MHZ 486 machine	2400 kbps	2
Software on Pentium II 450mhz	23 Mbps	5
Software on Sun enterprise E4500	147 Mbps	5
VLSI on 25 MHZ chip	177Mbps	2
VLSI on 0.8 $\mu$ m CMOS ( 1995)	355 Mbps	5
VLSI on 0.7 $\mu$ m CMOS ( 1998)	424 Mbps	5
Ascom IDEA crypt coprocessor	300 Mbps	5

## 7. CONCLUSIONS

The performance of cryptographic processing is very crucial for a good network design. VLSI implementation of IDEA algorithm is suitable because of regular structures used for the construction of arithmetic/logical blocks. Security related function requires much more processing power from the processor than packet processing functions. So the Interface issue of a network processor with the security engine is a major one to be drawn focus.

## 8. ACKNOWLEDGEMENT

The authors wish to thank Mr. S.Sendil kumar, PG Student, L.D.College of Engg, Ahmedabad.

## 9.REFERENCES

- [1] Feghali, W., Burres, B., Wolrich, G., Carrigan, D., "Security: Adding Protection to the Network via the Network Processor." Intel Technology Journal. <http://www.intel.com/technology/itj/2002/volum e06issue03/> (August 2002).
- [2] Bruce Schneier, *Applied Cryptography, Second Edition Protocols, Algorithms and Source code in C* John Wiley & Sons Inc., Newyork ,2001
- [3] William Stalling: *Cryptography and Network security, Second Edition*, Prentice Hall, New Jersey,1999
- [4] Haiyong Xie, Li Zhou, and Laxmi Bhuyan "Architectural analysis of Cryptographic applications for Network Processors ," Workshop on Network Processors, pp 42-52 February 2002.
- [5] M.P.Leong, O.Y.H. Cheung, K.H.Tsoi, P.H.W. Leong " A Bit serial Implementation of the International Data Encryption Algorithm" ,FCCM ,April 2000
- [6] Wajdi Feghali, Gilbert Wolrich,,Douglas Carrigan, Intel Communications Group "High integration makes IPsec fly" An article in EE Times October 14, 2002
- [7] John Freeman, Fearless Group "Changing the Game Winners, Losers, and Opportunities Resulting From the Adoption of the Network Processor Unit (NPU)" Network+interop Las Vegas 2000.
- [8] "Hardware Working Group " [www.npforum.org/techinfo/charters.shtml](http://www.npforum.org/techinfo/charters.shtml)