Design and Implementation of a TCP/IP packet filter and classifier IP block through High Level Synthesis

Benjamín Vega del Pino, Pedro P. Carballo and Antonio Nuñez
IUMA, Institute for Applied Microelectronics, University of Las Palmas Gran Canaria, Spain
{bvega,carballo,nunez}@iuma.ulpgc.es

***Abstract*—This paper presents the work done to design a packet classifier implemented on a FPGA. The target of this IP block is to accelerate the decision process of a network security system. This block decides whether a packet is sent either to a DPI block or Ethernet interface. The decision is made by checking the values of different fields within the headers of the Ethernet and IP layers.**

Keywords-component; TCP/IP; classifier; packet; inspection; hardware; DPI; Zynq; IP block;

I. Introduction

During the recent years the amount of cyber-attacks has increased drastically. The targets are not only large companies but increasingly ordinary Internet users, public figures and celebrities. The wide range of victims remarks the need of implementing security management systems within networks. This is the reason why security institutions, in both the public and private sector, are investing in research and development of more efficient DPI systems.

DPI systems offer a higher security level than ordinary firewalls and anti-malware applications and are located in the fog. The DPI systems are based on algorithms to detect certain patterns in the payload of TCP/IP packet. These systems have the goal of detecting intrusions, management of the bandwidth and another security issues.

However, the extensive use of the network and the high speed rates of the Internet connection require DPI systems to work with high throughput communication lines. The analysis should be performed within short critical periods of time in order to not produce bottle necks on the supervised network. In order to satisfy these timing requirements these systems should be designed on hardware instead of using the traditional software application approach, taking advantage of the hardware concurrency features [1][2].

The implementations of hardware accelerators in DPI systems normally follow a CPU-DMA based architecture (CDBA) in which the communication between the processor and the accelerator is based on DMA transactions. This architecture presents bottlenecks in the RAM transactions and CPU processing time that limit the performance of the whole system. [3]

This article presents an approach to overcome the limits of the CDBA by developing an IP block that allows the DPI system to perform the analysis without any processing core. The IP block assumes the role of a pre-filter of the incoming traffic to redirect the potential malicious packet to a DPI system or forward the packet to the output Ethernet interface.

II. Main characteristics of the IP block

Performing a filtering on hardware according to the values of the header fields reduces the latency of the process since the packet does not need to be transferred to RAM and be inspected on the CPU. This approach is defined as Hardware-IP base architecture (HIBA). By using this block the processing system does not need a TCP/IP Stack library to manage the incoming packets. That produces a reduction in the power consumption and increases the throughput of the system. The differences between the CDBA and HIBA are shown in the Fig1.

In order to achieve these goals during the design of the IP block some strategies have been considered. The strategies are based on making a shadow copy of the relevant flits while the packet is stored flit by flit in a FIFO. An important feature is that the analysis takes place once the shadow copying has finished. This implies that the result of the analysis is obtained before the packet is completely received.

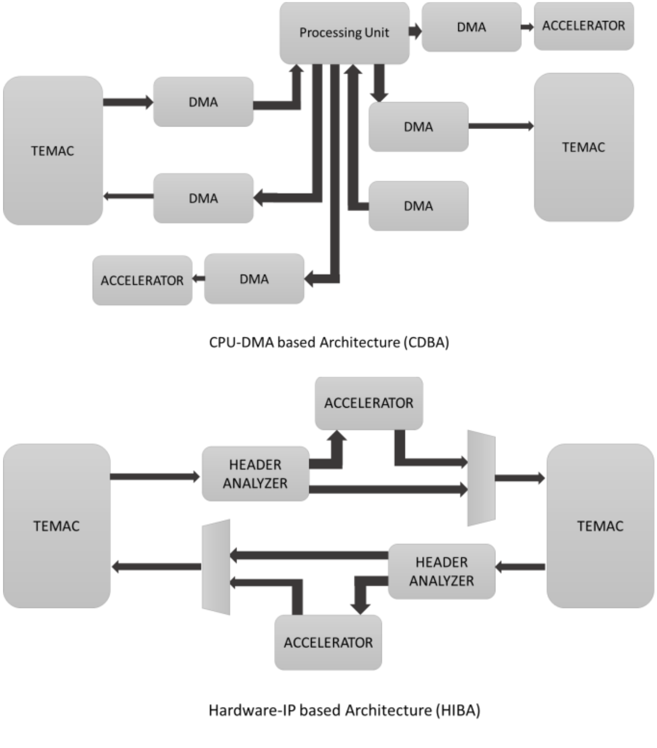

Figure 2. Architecture Diagrams

III. METHODOLOGY

The quality of the design derives from the degree of control that the designer considers during the modelling, synthesis and implementation phases of the IP block. In order to achieve a high degree of control over the synthesis results, the followed design methodology is comprised of a heterogeneous set of tools, where every tool has been chosen specifically to yield the best results in each phase of the design flow.

The IP block has been modeled in SystemC. This decision has been made taking into account the key features that SystemC offers as a hardware description oriented language. [4][5]. SystemC has been used in the modeling phase because of the intrinsic cycle accurate characteristics of the language. Otherwise, C/C++ language does not allow the designer to model the behavior of communication interfaces at CABA level. Taking the advantage of timing control, the IP block has been structured in such a way that the different processes that take place in the block are organized following a pipeline strategy. This structure allows the IP block to work as a dataflow model.

The **C-to-Silicon** (**CtoS**) HLS environment has been used to perform the RTL synthesis. This environment presents a degree of schedule control that is higher than other environments used in previous experiences.

The logic synthesis and the optimization has been done using Synopsys **Synplify**. Synplify gets better results that other tools used such as Vivado. This tool is also useful to analyze the critical paths of the design in order to perform changes in SystemC model to increase the maximum frequency of the block.

IV. THE IP BLOCK

The development of this block has been approached by dividing its functionality in different processes. This technique presents better synthesis results and also eases the verification phase.

The block is comprised by three processes and an internal module. The Fig2 presents a diagram of the block structure and the numeration of each process that compose it. The first process manages the bus protocol of the AXI4-Stream [6] input channel and stores the incoming data. The second process performs the analysis once the first nine flits of each incoming packet arrive. The third process takes the decision to forward the stored packet through either, the Ethernet interface or the DPI system, according to the analysis result. Finally, the AXI4-Lite [6] is included as a module with several processes that handle not only the bus signals of the protocol but also update the configuration registers that are used in the analysis process.

The results of the logic synthesis given by Synplify determine that the maximum frequency at which the block can operate is 460 MHz. It means that the block may operate with 32-bit network interfaces up to 14Gbps such as OC-256 (Optical Carrier) per network interface.

V. DESIGNING THE PLATFORM

The platform is the hardware support for the system that uses the designed IP block. This hardware is based on two Tri-Ethernet-MAC controllers connected each other through two instances of the designed IP block. This way the incoming traffic of each interface will be forwarded to the opposite interface or the hardware system to run the DPI process.

The tasks done by the embedded software serve initialization and configuration purposes of the IP block and the DPI system. The software latencies that occur in the system are reduced remarkably since the CPU does not affect to the capturing, analysis and forwarding processes of the packets.

VI. RESULTS

1) *Architecture Comparisson*

The results of the platform that includes the designed IP (HIBA) have been compared with a platform that follows the CDBA with a hardware-software approach. Both platforms have been implemented in a ZC706 with a CPU clock frequency of 800MHz and a clock frequency of the FPGA of 200MHZ.

The measurements of the latencies have been taken using a hardware counter in the Processing System and an Integrated Logic Analyzer core. These measurements determine the latencies form the transmission of the packet from the TEMAC core until either the forwarding to the opposite TEMAC core or the DPI core (TABLE I).

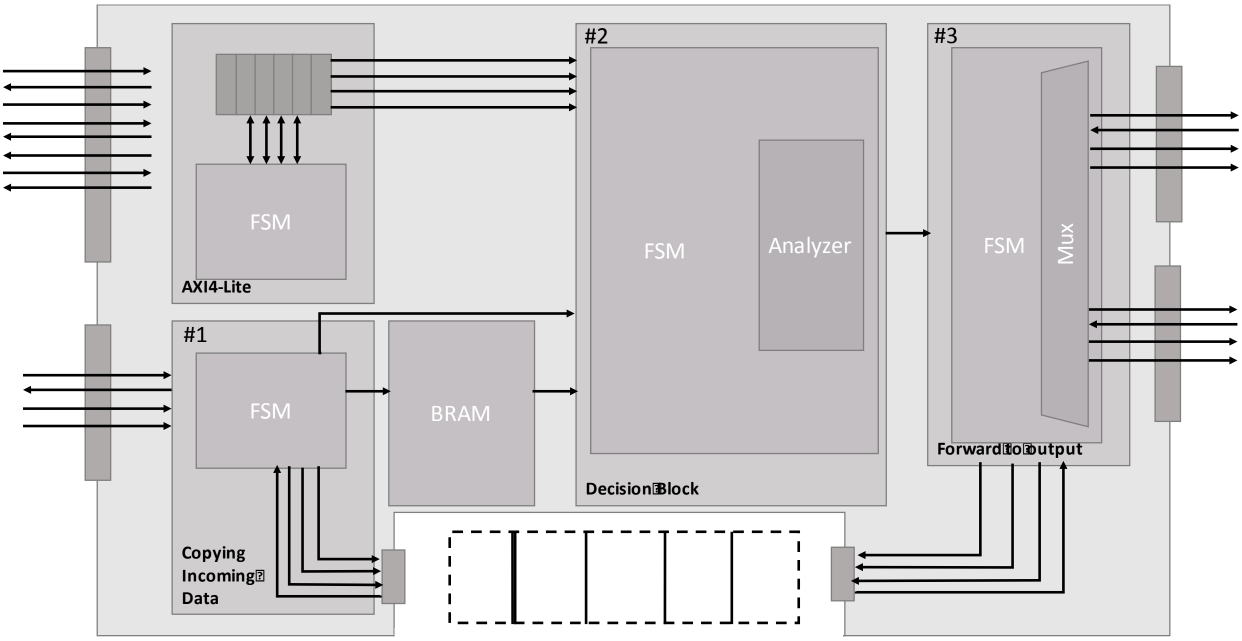

Figure 2. IP block structure

These measurements are the assessment of evidences for the acceleration that presents the developed IP block. The HIBA solution reaches an acceleration **x106** times faster than the CDBA solution.

The developed IP block also presents a significant reduction of the required hardware resources in order to perform the equivalent task in the classic architecture. This also leads to a reduction of the power consumption in the FPGA and the whole system (TABLE II).

This solution also has benefits in the power consumption of the system. As one may observe in the TABLE III, the HIBA offers a reduction of power consumption of 7 times over the CDBA solution.

2) *Solution comparisson*

In 2015 a research group designed a block whose goal is quite close to the developed IP block's one. This block is called DPFEE. The cited block works in systems that handle up to 25.71Gbps and it is implemented on a Virtex-7 [7]. Therefore, in order to compare the developed and the cited block, the developed block was synthetized for Virtex-7 as well.

TABLE I. MEASUREMENTS OF THE LATENCY

Param.	**Definition**	**CDBA**	**HIBA**	**CDBA/HIBA**
$\mathbf{t}_{ad}$	Time from the packet arrive until the process of making the decision starts.	13,6 µs	0,04 µs	340
$\mathbf{t}_{td}$	Time taken to make the decision.	4µs	0,2µs	20
$\mathbf{t}_{dt}$	Time taken to the system to deliver the packet to the accelerator once the decision is made.	9µs	0,01µs	900
$\mathbf{T}_{tot}$	Total time.	26,6µs	0,25µs	**106**

TABLE II. RESOURCES UTILIZATION

	Slice LUTs	**Slice Regist.**	**LUTs as Mem.**	**LUTs as Logic**	**LUTS as FF**	**BRAMs**
CDBA						
- DMA TX	3033	5254	371	2662	4777	4
- DMA RX	3033	5254	371	2662	4777	4
Total	6066	10508	742	5324	9554	8
HIBA						
- IP Block	578	980	44	534	878	0
- FIFO	43	49	0	43	57	1
Total	621	1029	44	577	935	1

TABLE III. POWER CONSUMPTION

	CDBA	**HIBA**	**CDBA/HIBA**
Power [W]	0.056	0.008	**7**

As the IP block cited also analyzes the header fields of transport and application layer, the approach for the comparison is to form a system called MLA, which uses three blocks as the one developed in this project. This system is presented in the Fig3. The performance and resource comparison between each of the blocks is based on the synthesis results.

The developed IP block presents a better result taken as figure of merit an expression that considers the throughput of the block and the resources utilization (TABLE IV).

$$FoM = \frac{Throughput\ [Gbps]}{Resources\ utilization\ [\%]}$$

For this expression, the higher result the better is the solution approached. For this comparison were used the liming factor of this design that is the number of LUTs available.

TABLE IV. COMPARISON WITH DPFEE

	LUTs	**Throughput**	**FoM**
DPFEE	6.00 %	25.71 Gbps	4.16
MLA	3.66 %	16.00 Gbps	4.37

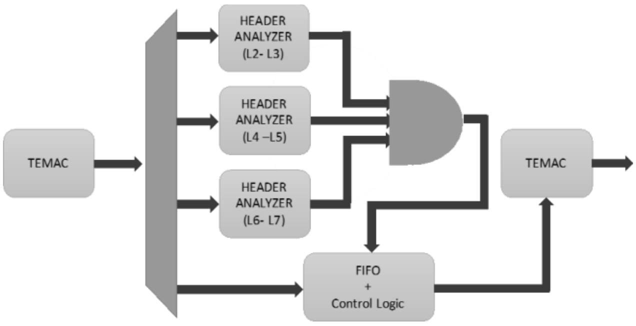

Figure 3. Results depending on the number of searches

VII. CONCLUSION

This document describes the characteristics of an IP block developed with the goal of designing a TCP/IP packet classifier according to the packet header field. This functionality was developed on an IP block, taking advantage of hardware design over software solution. This way the block can be used as a first stage analysis of a more complex DPI system.

The advantages of performing this filtering on hardware produces acceleration rate of x106 times faster than using a CDBA. The HIBA has produces a power reduction of x7 times of the total power consumption estimated for the CDBA.

REFERENCES

[1] M.Attig and G. Bredner, "400 Gb/s programmable packet parsing on a single FPGA," Proc. -2011 7th ACM/IEEE Symp. Archit. Commun. Syst. ANCS 2011, vol. 400, pp. 12-23, 2011.

[2] R. Dobai and L. Sekanina, "Towards evolvable systems based on the Xilinx Zynq platform," in 2013 IEEE International Conference on Evolvable Systems (ICES), 2013, pp. 89–95.

[3] B. Vega, P. P. Carballo, P. Hernández-Fernández, A. Domínguez, and A. Núñez, "TCP/IP Packet Analyzer on a Zynq Platform," in DSD 2015 -- Euromicro Digital Systems Design 2015 (WIP), 2015, p. 2

[4] A. Takach and M. Meredith, "SystemC Synthesis Working Group." [Online]. Available: http://accellera.org/activities/working-groups/systemc-synthesis.

[5] J. Verhaegh and A. Su, "SystemC Synthesizable Subset," Synthesis (Stuttg)., 2009

[6] A. X. I. Reference, V. Axi, and R. Guide, "Vivado Design," vol. 1037, 2015

[7] V. Jyothi, S. K. Addepalli, and R. Karri, "Deep Packet Field Extraction Engine (DPFEE): A Pre-processor for Network Intrusion Detection and Denial-of-Service Detection Systems," pp. 266–272, 2015.