Development of a Virtual Platform for a FPGA based hardware accelerator for DPI

Sonia Raquel León Martín, Pedro Pérez Carballo and Antonio Nuñez Ordóñez
IUMA, Institute for Applied Microelectronics, University of Las Palmas Gran Canaria, Spain
{slmartin,carballo,nunez}@iuma.ulpgc.es

***Abstract*—. Virtual platforms provide a full hardware/software platform. The software can be developed without the physical device. This paper describes the development process of a virtual platform for Deep Packet Inspection accelerator using transaction level modeling (TLM). We have used Mentor Vista platform that provides an environment to create virtual platform and perform simulation, debugging and analysis hardware/software.**

Keywords—virtual platform; TLM; DPI; ESL; Mentor Vista

I. INTRODUCTION

Nowadays, the growing complexity of the embedded designs has produced that RTL modelling and verification techniques are not effective to design complex device architectures. Therefore, it is necessary to create fast and accurate simulation models to reduce the time-to-market for new products. Design flows will inevitably shift toward other methodologies that raise the level of abstraction above RTL. The electronic system level (ESL) is a methodology that focusses on the global architecture of the design. It is used for architectural exploration and hardware/software partitioning as well as for initial verification of the design [1].

Raising the level of abstraction requires a mechanism that models communication and functionality at higher levels. This mechanism is called transaction-level modelling (TLM) provides compact descriptions without detail signal-level because the communication is performed using function calls. TLM allows designers to quickly design, modelling and validation. The introduction of TLM and ESL methodologies allow modeling an entire system facilitating the virtual platform development.

A virtual platform is an executable software model of a hardware electronic system based on SystemC transaction level modeling (TLM). Virtual platforms let software developers start coding early and concurrently with the hardware development. This reduces the overall software development cycle and shorts the time-to-market [2].

This paper presents the development of a virtual platform based on Xilinx Zynq-7000 Series SoC FPGA device for DPI applications. We used Mentor Graphics Vista environment to create the virtual platform. It provides all necessary resources, including library of models, IDE and simulation and debugging tools. The main goal for creating this virtual platform is to explore the design space.

II. RELATED WORK

In this section, we will give a brief introduction to DPI, SystemC, TLM-2.0 and QEMU.

A. Deep Packet Inspection

Deep packet inspection (DPI) is a technique that is employed to analyze TCP/IP traffic in real time and considers all information in the packets to be subject to inspection, including the payload [3]. DPI uses a search engine to detect patterns in the payload of TCP/IP packet. These systems are composed by a TCP/IP packet capture and filtering block that analyzes packet's header and determines if the packet payload will be analyzed, a block to eliminate the headers of Ethernet packets, and multiple analysis blocks. The analysis blocks can be based on pattern search algorithms (Boyer-Moore [4] or Wu-Manber [5]) or other more complex, like machine learning [6].

Because of the complexity of DPI architecture, we propose to develop a virtual platform which allow architecture design space exploration to accomplish the best solution for different use cases (Figure 1).

B. SystemC

SystemC is a system modeling language and C++ class library allowing hardware/software co-design and co-verification. It was developed by Open SystemC Initiative (OSCI) in 1999 and become in standard IEEE 1666-2011 [7]. SystemC has a high level of abstraction and can simulate concurrency, events and signals of a hardware, adding simulation time to C++. Models based on SystemC are composed hierarchically from other modules. A module can communicate to other through ports/exports and channels. The functionality is described on SystemC processes which explicitly synchronize on events (notify/wait). SystemC also offers an event-based simulation kernel that is compiled and linked with the model to create a compiled simulation engine.

C. TLM-2.0

OSCI committee, responsible for SystemC, helped define the TLM standard (TLM 1.0 and 2.0) that provides rules for a set of interfaces to be defined for a TLM component. This standard makes models reusable and interoperable [8]. TLM-2.0 was adopted to develop virtual platform because of using a memory-mapped bus model can connect to an ISS.

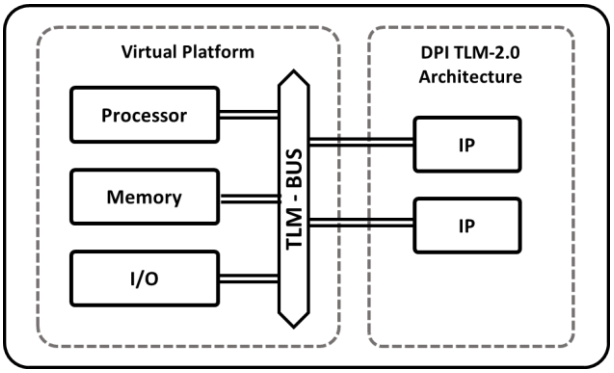

Figure 1. Virtual platform for DPI

TLM provides abstractions for the communication protocol and interfaces between the SystemC modules. Communication between modules is done at transaction-level using sockets and is represented as function calls using two different methods: blocking calls and non-blocking calls. In addition, it provides different levels of time accuracy: untimed, simulation can be completed in zero virtual time; loosely timed (LT), only two-timing points with each transaction; and approximately-timed (AT), allowing a non-specified number of timing point per transaction.

D. QEMU

QEMU is a generic open source processor emulator based on a portable dynamic binary translator (DBT) that executes unmodified standard operating system and application programs of a target architecture on a host. Supports multiple hosts and target CPUs like ARM, CRIS, x86, x86-64, MicroBlaze, Sparc, etc. Each instruction of the target CPU is translated into a set of micro-operations that are implemented by C functions for the host machine. These functions are compiled to obtain a dynamic code generator. The generator is invoked at run time to obtain the corresponding code to be executed on the host machine. QEMU offer high simulation speed and has been widely used as a virtual platform. It has been released as Open Source and therefore there are many implementations available, including libraries to interface QEMU with SystemC TLM [9].

III. VIRTUAL PLATFORM

Virtual platform is an executable software model of a hardware system that runs on a host computer and must be binary compatible to physical hardware (instruction set, memory map, registers). It runs the full software stack from boot-code and provides the same debugging utilities as the physical hardware [10].

A. Advantages

Also, virtual platform provides several advantages over the physical one. Firstly, software development can start before a physical prototype is available, thus verifying firmware available before a physical prototype, reducing hardware bring-up time, run test special conditions without damaging hardware.

Other benefits are efficient management of design complexity; accurately and efficiently exploration of different solutions, balancing design functionality, performance, power, etc.; refining communication architecture separately from functionality using more abstract models of functional units among others (Figure 2).

B. Vista Virtual Prototyping

Mentor Vista platform [12] provides an early and abstract functional model of the hardware to debug complex software/hardware design and to optimize the software. Vista offer a TLM-2.0 modeling methodology based on layers to separate communication, functionality and power/timing (Figure 3). The tool provides two-simulation modes: loosely-timed software execution (LT mode) and approximately-timed simulation for hardware verification, performance and power analysis (AT mode).

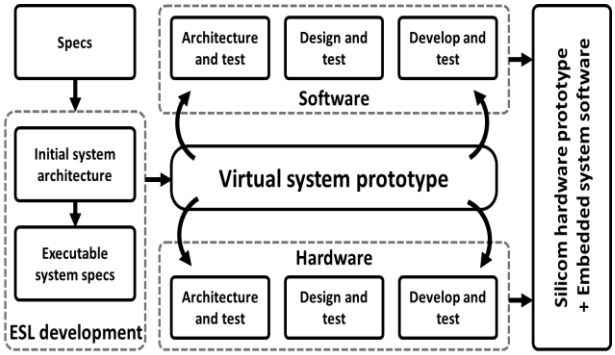

Figure 2. Virtual platform approach [11]

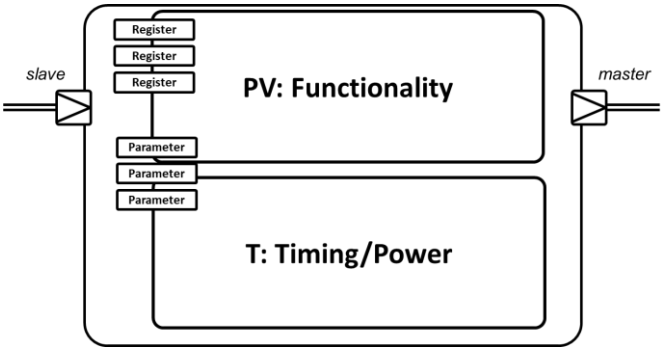

Figure 3. PVT model structure

IV. CREATING A VIRTUAL PLATFORM

Vista provides a library of pre-defined transaction-level models to create a virtual platform at TLM level. Users can customize key attributes, create new model or import external models into the Vista model library.

DPI TLM-2.0 models have been created using Vista Model Builder derived from the reference platform. DPI architecture includes the following blocks (Figure 4):

- *TrafficGenerator*, generates TCP/IP traffic to the system, mainly used for verification;
- *header_analyzer*, analyzes TCP/IP headers to decide if the packet will be sent to the search engines;

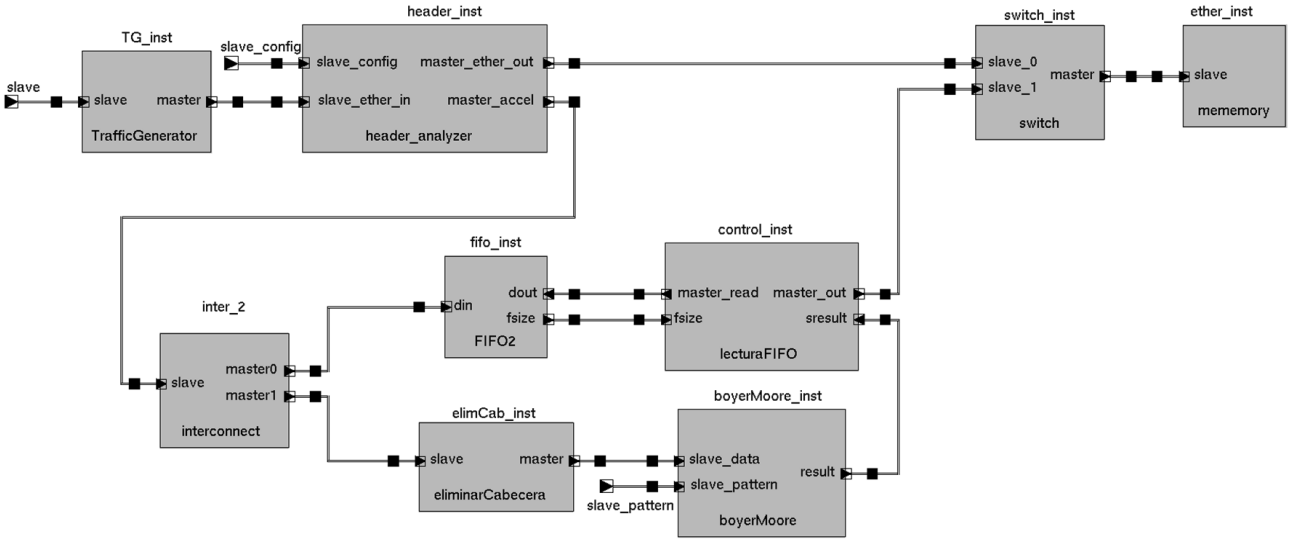

Figure 4. DPI Architecture on Vista

- *eliminarCabecera,* eliminates the headers of the packet to obtain the packed payload;
- *FIFO2 memory*, stores the incoming packets while the search engine takes a decision if the packed will be forwarded to the network or will be dropped;
- *boyerMoore search engine,* a search engine implemented using Boyer-Moore pattern search algorithm;
- *lecturaFIFO*, FIFO Memory controller;
- *interconnect* and *switch,* broadcaster and interconnection blocks, respectively.

Model Builder is a sub-set of Vista that facilitates the creation of models using transaction level TLM-2.0, where communication, functionality, and timing/power attributes are modeled independently.

SoC TLM models are included in Vista with Virtual Prototype Kits that provides configurable virtual prototype for Xilinx Zynq-7000. The embedded software design flow is integrated in Vista, validating and optimizing the software on an abstracted simulation model of FPGA prototype.

Once TLM platform design is build, the virtual platform can be generated. The virtual platform is a stand-alone executable created from the TLM platform which contains the default parameters of the design, the simulation executable and shared libraries.

Vista Virtual Prototyping provides a rich set of analysis capabilities for the embedded software and hardware under software control. The analysis results are provided through analysis viewing windows and analysis summary reports.

V. ANALISYS

The proposed DPI architecture was analyzed using the virtual platform created. The analysis was realized using this scenario:

- the simulation mode assigned is AT;
- The traffic generator transmits 42 packets, which 18 contained header parameters that forced a deep inspection by Boyer-Moore;
- The pattern assigned was “raquel leon” and 6 packets contain this pattern in their payload;

Table 1 show the number of transactions extracted from the socket tree of the model (**Table 1**).

Table 1. Socket throughput

Socket Name	Trans count	Trans/us	Bytes/us
TrafficGenerator.master	42	0.05091	72.9
HeaderAnalyzer.slave_ether_in	42	0.05091	72.9
HeaderAnalyzer.master_ether_out	24	0.02909	41.66
HeaderAnalyzer.master_accel	18	0.02182	31.24
HeaderRemover.slave	18	0.02182	31.24
HeaderRemover.master	18	0.02182	30.11
BoyerMoore.slave	18	0.02182	30.11
BoyerMoore.result	18	0.02182	0.1745
FIFO.din	18	0.02182	31.24
FIFO.dout	18	0.02182	31.24
Ether_out.slave	36	0.04364	62.49

Vista analyzer provides a set of graphic tools to display power, throughput, latency and bus bandwidth. The main power consumption in the platform is produce by CPU during initialization state reaching 100 mW (Figure 4). The DPI throughput is shown in Figure 5. The DPI latency with 42 packets is 724µs.

VI. CONCLUSIONS

In this paper, we present a Virtual Platform for a DPI system to be implemented in a Zynq 7000 FPGA system. The

virtual platform allows the exploration of the design space, the debug and the analysis of the hardware/software solution. The platform has been modeled on SystemC TLM-2.0 that provides a rapid creation of complex design and can be interconnect with other models. The software has been development at the same time that the different blocks has been modeling, allowing a fast creation of the platform.

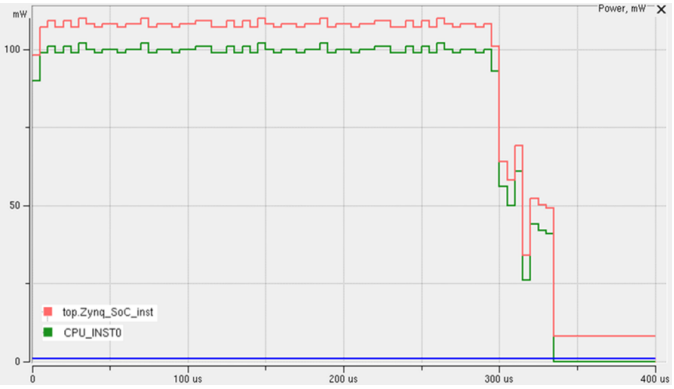

Figure 4. CPU Power consume during initialization platform

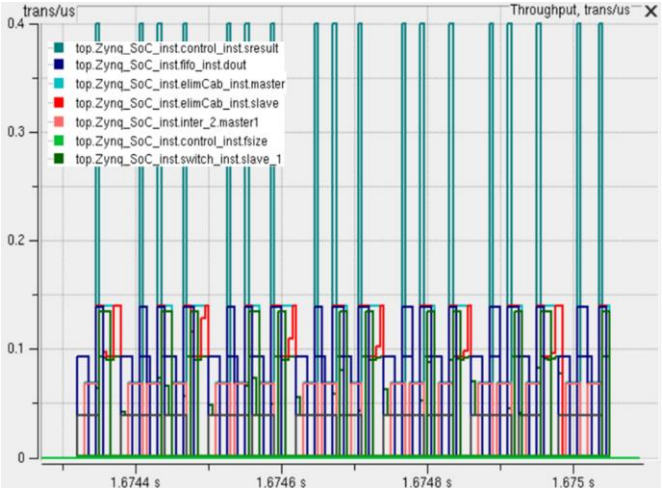

Figure 5. DPI throughput

REFERENCES

[1] J. Zhu y N. Dutt, «Chapter 5 - Electronic system-level design and high-level synthesis», en *Electronic Design Automation*, L.-T. Wang, Y.-W. Chang, y K.-T. (Tim) Cheng, Eds. Boston: Morgan Kaufmann, 2009, pp. 235-297.

[2] A. B. Mehta, «ESL (Electronic System Level) Verification Methodology», en *ASIC/SoC Functional Design Verification*, Cham: Springer International Publishing, 2018, pp. 221-241.

[3] M. Mueller, «DPI technology from the standpoint of Internet governance studies.», Nueva York, 2011.

[4] R. S. Boyer y J. S. Moore, «A Fast String Searching Algorithm», *Commun. ACM*, vol. 20, n.º 10, pp. 762-772, 1977.

[5] S. Wu, U. Manber, y others, «A fast algorithm for multi-pattern searching», University of Arizona. Department of Computer Science, Tucson, AZ (USA), may 1994.

[6] U. Trivedi y M. Patel, «A fully automated deep packet inspection verification system with machine learning», en *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, 2016, pp. 1-6.

[7] IEEE, «IEEE Standard for Standard SystemC Language Reference Manual», *IEEE Std 1666-2011 (Revision of IEEE Std 1666-2005)* -. pp. 1-638, 2012.

[8] OSCI, «OSCI TLM-2.0 LANGUAGE REFERENCE MANUAL». 2009.

[9] F. Bellard, «QEMU, a Fast and Portable Dynamic Translator», en *FREENIX Track: 2005 USENIX Annual Technical Conference*, 2005, pp. 41-46.

[10] A. Fasching, «Virtual Prototyping», 2016. [Online]. Disponible en: https://ti.tuwien.ac.at/ecs/teaching/courses/hwswcode_vu/hwsw-codesign-student-presentations/3-virtual-prototyping.pdf. [Accedido: 20-abr-2018]

[11] A. Lonardi y G. Pravadelli, «On the Co-simulation of SystemC with QEMU and OVP Virtual Platforms», en *VLSI-SoC: Internet of Things Foundations*, 2015, pp. 110-128.

[12] Mentor Graphics, «Vista™ User's Manual». Mentor Graphics Corporation, 2017.