A New Embedded Platform for Rapid Development of Network Applications

Jan Korenek, Pavol Korcek, Vlastimil Kosar, Martin Zadnik, Jan Viktorin
Brno University of Technology
Faculty of Information Technology
IT4Innovations Centre of Excellence
Bozetechova 1/2,612 66 Brno, Czech Republic
{korenek, ikorcek, ikosar, izadnik}@fit.vutbr.cz, xvikto03@stud.fit.vutbr.cz

Categories and Subject Descriptors

C.3 [**Special-purpose and application-based systems**]: Real-time and embedded systems

General Terms

Design, Embedded, Networking

Keywords

FPGA, Zynq, Embedded, Networking

1. INTRODUCTION

NetFPGA-1G [2] has shown its potential in enabling fast traffic processing while introducing no packet loss and minimal delay. Now it is time to scale down in order to enable a massive deployment of the FPGA solutions in networking. We propose and build a low-cost and low-power platform which is be capable of hosting embedded applications with FPGA support. Such a platform might enable faster deployment of new ideas in networking and might prove useful for large-scale experiments (stacks of platforms) as its size, power consumption and cost are expected to be ten times lower of the NetFPGA-cube. It is recognized that the FPGA coupled with the host processor comprises a powerful platform for network traffic processing. The logic of such a solution is clear. Computational intensive tasks are handled by the FPGA logic whereas more complex tasks by the processor. The proposed platform aims at such a design in which the FPGA and the processor are even more tightly coupled together on a single die SoC solution.

As a proof of this concept, we build a platform called uG4-150. Its FPGA hosts a synthesized softcore processor (e.g. Xilinx MicroBlaze). But our final goal is to utilize the hardcore processor (e.g. ARM-based) integrated with the FPGA such as Xilinx Zynq [1].

2. SYSTEM CONCEPT

The concept of the whole system is based on a modular and layered design (see Fig. 1) to minimize the efforts when porting it to a new platform or when new components are introduced. The concept comprises hardware platform,

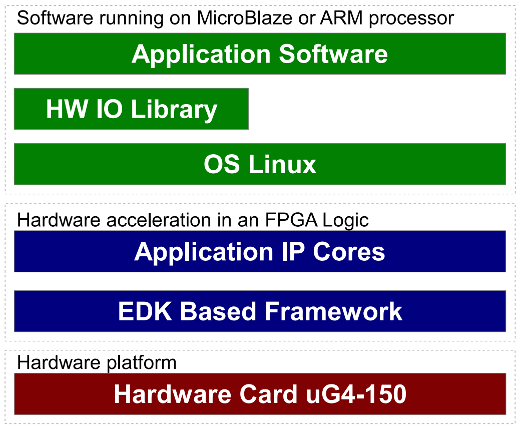

Figure 1: Concept of the system.

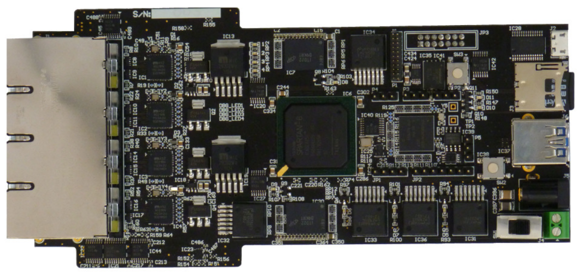

Figure 2: uG4-150 hardware platform.

EDK-based framework, firmware, application IP cores, operating system and application software libraries.

The first layer represents the hardware platform itself. In our case it is uG4-150 platform (depicted in Fig. 2) with its main processing element Xilinx Spartan-6 FPGA but other similar platforms may fit as well. Based on the analysis of FPGA components (i. e. Ethernet cores, DMA cores, MicroBlaze soft processor core, AXI interconnection system and packet processing system) we have equipped uG4-150 with the high-end Spartan-6 (XC6SLX150-3FGG484C).

uG4-150 also hosts additional components such as:

- 2x 256MB of DDR3 for the processor and the FPGA processing system,
- 4x 1 Gbps Ethernet over metallic RJ45 connectors,

- USB 3.0 (type A) connector to a fast data storage,
- 16 MB 4-bit serial flash memory (FPGA boot),
- 512 kB I^2C PROM (USB3.0 controller boot),
- slot for microSD memory cards, etc.

The EDK-based framework abstracts application IP cores from specifics of the given platform by providing controllers to the various peripheral components. The most important modules are: AXI Ethernet (transforming RGMII interface to AXI-Stream), SDRAM controller (AXI S6 DDRX controls DDR3 memories), SD card controller and others.

The application IP cores implement various tasks of basic packet processing. These tasks ranges from protocol analysis, packet classification, to filtration and queuing. In case of protocol analysis, the structure and encapsulation of protocols is described in XML and synthesized into VHDL. Despite that the extracted fields may be changed on-the-fly. Filter and classification components identify packets based on the match of its IP addresses and/or ports and protocol number against the given set of rules. The matching of prefix rules is implemented using Tree bitmap algorithm [6] whereas the matching of exact rules is implemented as Cuckoo hashing [5]. Further the packets may be queued in a fairly large buffer in DDR3 memories. The size of these memories provide capacity large enough to store more than a second of traffic at the wire-speed. Finally, there is a module implementing fast and secure packet encoding and secure delivery. All the cores may be assembled in a modular way to perform more complex network operations. The modularity has been achieved through utilization of standard AXI-Stream bus to pass data among modules and AXI-Lite for management purposes.

The Linux OS [3] provides standard ways to access the whole system. It allows to run a wide set of UNIX-based software such as ssh server, iptables or the Click Modular Router [4]. The current version of the operating system can be used on the Zynq platform without any significant modifications.

The access to the non-standard IP cores, which come without any standard kernel-space driver, is done over an abstract layer called HWIO. HWIO is a tiny library that utilizes *device-tree* provided by the Open Firmware kernel [1] to access memory mapped registers of application-specific components.

The application software initializes the HWIO and looks for a compatible core which performs required job. If such is found it is initialized and configured according to the needs of user.

3. USE CASES

The uG4-150 has been tested in application scenario aiming at legal network traffic interception. The system of the probe is built using the proposed system. It utilizes the IP cores to delay incoming packets, parse and extract their headers, filter incoming packets and send them to the collector. Tab. 1 shows synthesis results in ISE 14.1 xst tool for 32-bit wide processing pipeline and 128 filtering rules.

The platform may serve other applications as well. The striking one are specific routers or switches (OpenFlow), firewalls, various proxies, gateways and monitoring probes.

[1] www.openfirmware.org

Type	BRAMs	LUTs	FFs	Freq.
uG4-150	6(2%)	8887(9%)	2482(1%)	111 MHz
Zynq-7030	3(1%)	8046(7%)	2419(1%)	218 MHz

Table 1: Real and expected FPGA resource consumption on uG4-150 and Zynq platforms respectively.

4. CONCLUSIONS

This paper proposed a platform for rapid prototyping of high-speed and low-power embedded applications in networking. The concept utilizes the FPGA with the embedded processor to benefit from software flexibility and high performance of hardware processing. In comparison with the NetFPGA-cube, the proposed uG4-150 platform has significantly lower power consumption, cost and size.

As the processor is running OS Linux, the applications can use standard Linux tools and libraries to shorten software development. Moreover, hardware development is simplified by the prepared set of IP cores. These cores accelerate basic time-critical operations, provide AXI-Stream interface, and can be simply connected with Xilinx EDK into the processing pipeline. All IP cores are configurable from user space by the HWIO library.

Currently, the uG4-150 platform utilizes Spartan-6 FPGA with the MicroBlaze processor. But the aim is to utilize Xilinx Zynq with hardcore ARM processor. The uG4-150 is meant to be a functional testbed for upcoming development of platform with Zynq.

5. ACKNOWLEDGMENTS

This work has been partially supported by the Research Plan MSM 0021630528, IT4Innovations Centre of Excellence project CZ.1.05/1.1.00/02.0070, the grant BUT FIT-S-12-1 and Sec6net project VG20102015022.

6. REFERENCES

[1] Xilinx Inc., *Zynq-7000 EPP Overview*, Advance Product Specification, DS190 (v1.1.1) June 11, 2012 `http://www.xilinx.com/support/documentation/data_sheets/ds190-Zynq-7000-Overview.pdf`.

[2] NetFPGA Team, *NetFPGA-1G*, `http://www.netfpga.org/php/specs.php`.

[3] J. Viktorin, *MicroBlaze Simple Linux*, `https://github.com/jviki/mbsl`

[4] Click Modular Router documentation - `http://pdos.csail.mit.edu/click/doc/`

[5] Rasmus Pagh and Flemming Friche Rodler. Cuckoo hashing. *J. Algorithms*, 51(2):122-144, 2004.

[6] W. Eatherton and G. Varghese and Z. Dittia. Tree bitmap: hardware/software ip lookups with incremental updates. SIGCOMM Comput. Commun. Rev., 34:97-122, April 2004.