INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2010, VOL. 56, NO. 2, PP. 107–110
Manuscript received January 21, 2010; revised June, 2010.
10.2478/v10177-010-0013-4

Large Data Stream Processing – Embedded Systems Design Challenges

Adam Handzlik and Andrzej Jabłonski

***Abstract*—The following paper describes an application of reconfigurable hardware architectures for processing of huge data streams. Radar, sonar and high speed internet networks are typical sources of data that require extreme computing power and resources to enable real time acquisition, processing and management. An approach to monitoring of real time multi-gigabit internet network has been described as a practical application of FPGA based board, designed for fast data processing.**

***Keywords*—Reconfigurable hardware, system on chip, digital signal processing, embedded systems.**

I. Introduction

CONTEMPORARY LIFE gets more and more convenient as we surround ourselves with numerous devices either for pleasure or to speed up our work. Electronics is installed in almost all machines and equipment, used in different domains of life. Industry manufacturers develop and manufacture hardware control platforms every day. They have to be still in motion by improving their control systems to be competitive on European and world markets, always a step ahead towards their rivals. Hundreds of different solutions, platforms, architectures and configurations offered on the market seem to fulfill wide range of requirements and needs. However, no matter which solution designers might choose, it soon becomes too slow for the next generation of the product family.

Whatever the application, somehow day by day we need more power, more performance, more resources, more, more and more…

II. Directions

Although there are many approaches to design of highly effective processing architectures, the Field Programmable Gate Array (FPGA) technology seems to be the most versatile and convenient. It is an integrated circuit that contains many identical logic cells that can be viewed as standard components. Each logic cell can independently take on any one of a limited set of personalities. The individual cells are interconnected by a matrix of wires and programmable switches. A user's design is implemented by specifying the simple logic function for each cell and selectively closing the switches in the interconnect matrix. Complex designs are created by combining these basic blocks to create the desired circuit.

A. Handzlik is with Microtech International S.A., Wołoska 20, 51-116 Wrocław, Poland (e-mail: a.handzlik@microtech.com.pl).

A. Jabłoński is with the Institute of Computer Engineering, Automation and Robotics, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland (e-mail: andrzej.jablonski@pwr.wroc.pl).

The FPGA has three main configurable elements: configurable logic blocks (CLBs), input/output blocks, and interconnects. The CLBs provide the functional elements for constructing user's logic. The IOBs provide the interface between the package pins and internal signal lines. The programmable interconnect resources provide routing paths to connect the inputs and outputs of the CLBs and IOBs onto the appropriate networks.

As the programming tools become more and more advanced, the FPGA chips can be configured as complex System-On-Chip architectures employing many types of peripherals and interfaces. Still, programming tools are only one of many aspects of efficient system development.

One might think that clocking of the digital system faster is the simplest way to increase the processing rate. However the last three or four years of work by world leaders in integrated circuit production proved that running the silicon chip itself at substantially higher clock frequencies is simply not achievable.

Internal chip clocking is therefore limited to 500...600MHz but even these frequencies are not describing particular system computing power properly. First, the implementation of particular application or algorithm usually requires multiple clock cycles to complete processing step, i.e. slows down the processing rate several times. Second, most of the system components interfaced with the FPGA chip work at lower clock frequencies. All types of memories are the most critical examples of these components, slowing down system clock frequency.

Power consumption might be a little bit neglected at this point, computing power is the goal here, after all. Power dissipation is far more important in this case and a proper cooling approach is essential at the development stage.

Finally, truly critical part of the fast system design is power regulation and distribution within the system board, and underestimation of that part of the design leads to painful disappointments at new system setting up stage.

III. Methodology

Maximizing system performance is far from an easy task and must take into account several aspects. The most important are: application type and implementation, interfaces, FPGA chip constraints, PCB design limitations and programming tools.

A. Application type and implementation

The application type, algorithms to be implemented define required system components, computing power and resources. In all cases the application should be decomposed into tasks which can be completed in the least possible clock cycles, performed in parallel (where possible) and executed in pipeline mode. These basic rules should be used during

application decomposition process and tasks implementation. They also mean that internal resources of the chip should be huge, preferably large enough to create one chip solution. Unfortunately it rarely happens and the application requires additional, external memory in easier cases or several FPGA chips interconnected in a specific ways, each using multiple memory ports. These more difficult architectures are truly efficient if chip-to-chip interfaces and application partitioning are done very well.

B. Interfaces

Interfacing FPGA chips with other system components can be done in many ways but one thing is sure – nothing shall be executed as fast as the same operation performed on chip. Hence numerous interconnection schemes, electrical standards and special circuitry dedicated to sending and receiving data off the chip. Even the simplest use of external memory requires precisely designed memory controller consuming FPGA pins and logical resources.

High performance systems utilize DDR2(3) dynamic RAM, RLDRAM and QDR memories. The best of them can be run at clock frequencies up to 300/333MHz. This limitation forced designers to application of multiple data ports for transmitting large data streams. Although it sounds quite natural, achieving above mentioned speeds in practice requires use of special software tools for memory interface creation and tuning.

C. FPGA chip constraints

The best solution any designer can imagine is one chip solution. It is true in many aspects. It could be run at top achievable clock frequencies and the PCB design would be far easier too. The most powerful FPGA integrated circuits offer several Mbits of configurable, internal memory, running at high frequencies. This is not enough for most applications and external memories have to be utilized. Multiple independent FPGA ports have to be interconnected with the memories to assure parallel access. In case of SODIMM module, a single, physical memory interface may use 288 pins. Considering that the system may need many independent memory ports and data transmission to the outside world may be performed by differential interfaces, the chip of over 1700 pins seems to be another limitation. Moreover, although FPGA pins and functions are configurable, interfacing to different peripherals requires the use of specific pin sets to obtain good results.

D. PCB design limitations

Every IC manufacturer publishes recommendations concerning chip interconnection (interfacing) and PCB design. It refers specifically to large programmable chips which can serve multiple functions and interface numerous other chips. High performance digital systems demand advanced tools to assure controllable crosstalk, power distribution and signal integrity.

E. Programming tools

Programming of FPGA circuits requires special, particular and responsible approach form designers, which is described in Reuse Methodology Manual. Reuse Methodology Manual for System-on-a-Chip Designs outlines a set of best practices for creating reusable designs for use in an SoC design methodology. These practices are based on the authors' experience in developing reusable designs, as well as the experience of design teams in many companies around the world. Silicon and tool technologies move so quickly that many of the details of design-for-reuse will undoubtedly continue to evolve over time. But the fundamental aspects of the methodology described in this book have become widely adopted and are likely to form the foundation of chip design for some time to come.

IV. PRACTICAL DESIGN

A practical approach to the design of high speed architecture was initiated as a direct answer to current problems of large companies and data communication services providers that relate to issues of implementing security methods both for data transmission and for IT systems. Special effort has been put into development of a layer architecture integrating results of various advanced analyses of network traffic, including the ones based on pattern recognition, on changes in packet traffic and filter functions. In addition, the solution will provide transmission analysis at speeds (in range of 10Gbps) that are by far higher than supported by the current hardware.

The main technical objective of the project is design of a hardware solution that would allow parallel operation of reactive and pro-active security algorithms, as well as designing contextual algorithms for detection models based on the packet behavior and pattern recognition. Another aim of the project is creation of more advanced and correlated collections of rules for effective alarm management and threat causes analysis.

At the functional level, identification of threat sources (identification of network switch, geographic location, service provider), which enable a global support for threat detection in the whole network and providing information to higher-level applications or people managing the network for further actions, such as blocking threat sources and alerting service providers, is expected.

As a result of the design process a new-generation security management architecture based on multiple agent system to model and implement smart intruder detection system should be created. The purpose is detecting both known and unknown network attack types, as well as minimizing false alarms occurring due to the use of one detection model (method), i.e. the approach that does not utilize the correlation between threat detection by anomaly analysis algorithms and data packets signatures.

Achieving a high level of operational correlation (and thus reducing the number of false or non-deterministic alarms) by combining the models analyzing anomalies, behavior of packets and data packet signatures with a selection of multiple security functions (such as pattern scanning or intruder detection), as well as combining the results of the above functions and algorithms to obtain the correct information in the context of specific location, network topology and data transmission in a specific session is further expected.

The new architecture should enable execution of varied types of functions, at transmission speeds higher than currently used, using FPGA structures and using multiple re-

programming capabilities, and defining the best way to design functions that examine the correlations in such a way as to facilitate the process activities upon identifying a threat. This operation will feed-back the optimization of real-time data processing speed, as well as optimization of parallel execution for individual and correlation post-process tasks.

Researching the possibilities of parallel algorithms processing to establish whether the performance that allows scaling the FPGA-based solution in the direction of analyzing data transmission at 40 Gbit/s or more can be obtained. The target speed of operation for the proposed solution is closely related to enabling the security functions in the actual transmission network. These functions will be key to a multi-layer security architecture connected with external desktop.

On the programming level the project should deliver a fully parallel, hardware genetic programming (GP) model to build more effective detection rules, using the initial detection rules created in the basic knowledge base of known attacks as input data, as well as to use the advantages of alternative techniques such as genetic algorithms (for monitoring network, data servers and detection rules evolution), niche technique (for constructing various detection rules, sequential niche technique, deterministic rule multiplication) and "Immuno-Fuzzy" approach method (used for anomaly detection, using fuzzy rules instead of fixed ones to cover non-specific threat areas, for example by using fuzzy detectors). The alternative techniques described above will be used in the methodological evolutional programming to increase intruder detection continuously and to enable the solution self-adapting to the changing conditions and threats.

Using "evidence" combining method according to Smith-Waterman algorithm for detecting masked intruders (intruders assuming the form of an authorized user), as well as bio-IT tools, such as similarity matrix, philogenetic trees and protocol analysis arrangements. Design works may be extended to include the criminal analysis of the computer stations (searching for hidden data chains in hard disks), analysis of user behavior in a network, etc.

The system structure includes two major components: hardware computing platform (interface module and processing module) and management station (Fig. 1). The objective of the hardware platform is to receive the Ethernet data stream and to detect potential threats using a set of rules. The management station enables system configuration, its updating and supports creating new rules based on historic data on network behavior from various sources.

The input interface are two bi-directional fiber optics links with the throughput of 10Gbps each. The two data streams are initially verified (Ethernet frame validity check) and are sent to further processing. The only possible intervention in the data stream sent through the device is packet blocking. After receiving a full IP frame, the packets are classified and the connection identified. For each set (sender's address, sender's port, receiver's address, receiver's port) an identification number (FlowID) is calculated and used as an address in the context table. If a given address set occurred previously, the relevant data structure is updated and made available for the further processing stages. If the address set occurs for the first time, the relevant structure is created. If the IP frame is at the same time a TCP frame status engine that allows tracking the TCP connection status is created. The context table automatically stores the basic connection information (addresses, protocol, packet statistics) and the information used by the user in the threat detection process.

The packet data and context record are then transmitted to the analysis block. The pre-compiled set of rules in the management station provides a basis for the threat detection system. The processing routine uses the data incoming from the secured network (analysis object) and the pattern database. Patterns are sets of searched signatures and include both simple texts and regular expressions. As a result of rules processing, actions are taken – context updating, packet blocking or alert report. If a threat is reported, the information is provided to the module communicating with the management station, which formats the message according to the IDMEF standards and sends it via an additional Ethernet link.

Selected algorithms in FPGA structures were chosen to complete these tasks as, to be able to utilize the advantages of parallel and pipelined processing. Defining the effective sharing of the tasks between the software and hardware platform enabled algorithms optimization to obtain real-time data stream processing as well as parallel execution of individual and correlation post-process tasks.

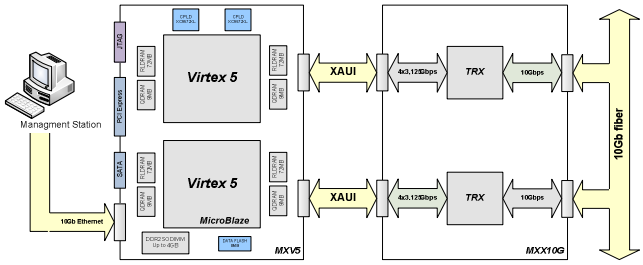

Fig. 1. Architecture of hardware platform of the device.

The management station serves as an interface between the hardware platform and system user. It includes the software that enables modification the security rules, their compilation and upgrading the hardware platform. This basic functionality is supplemented with a set of tools used for network security assurances using the information from various sources – system logs, historic data, network structure or other systems operating within an agent system. Combining the information from multiple sources, taking into account their quality and credibility, is aimed at enhancing threat detection [4]. Using correlation techniques [1] or analyzing time characteristics of network parameters provides a secondary source of higher-level information. Then, new rules may be based not only on the information directly available in the hardware platform, but also on the information from processed sources. A system based on genetic algorithms [2], [5] provides a help in creating new rules.

The system hardware platform was built based on MXV5 module designed and completed by Microtech. Block diagram of the module is presented in Fig. 2. Below is a set of module characteristics:

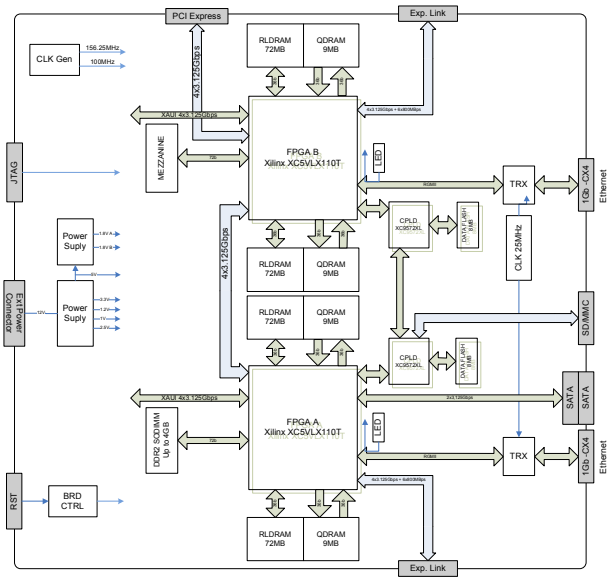

Fig. 2. Architecture of the processing board.

- two Xilinx Virtex 5 (XC5VLX110T) chips,
- two bi-directional 10Gbps interfaces on MXX10G board,
- two independent QDR II memory banks for each circuit (separate data write/read buses) – 9MB per each bank, total of 36 MB memory per each MXV5 board,
- two independent RLDRAM II memory banks for each circuit (reduced delays) – 72MB per each bank, total of 288MB memory per each MXV5 board,
- SODIMM connector for DDR II memory mod. max 4GB
- two 1Giga Ethernet interfaces, two eSATA interfaces for external storage,
- one PCIe interface (4 lines),
- Eurocard form factor with virtually unlimited capability of adding expansion boards (two expansion sockets with link sof 10Gbps + 6x800Mbps).

MXV5 module design enables construction of a parallel data processing system, focused on pipeline processing with multiple, simultaneous access to fast memory. A wide range of communication connectors provides for the option to configure the system for grid operation.

MXV5 module is supplemented with MXX10G board featuring 2 interfaces (based on Marvell 88E2010) and XFP module sockets, and enables connecting two 10Gbps fiber optics links. The connection between the two boards is a double XAUI interface. These two boards combined provide a basis for the first system prototype (Fig. 1).

V. Conclusion

The first hardware platform was prepared and the functional test stage was commenced.

At the hardware level, the design proved proper PCB routing, chip interfacing and system communication. Test tools used for hardware verification presented good memory interface speeds, full 10Gbit/s traffic handling by interface board and very good chip-to-chip and board-to-board transmission.

At the application level, basic functions have been implemented and tested. Processing speed achieved at the first stage enables full speed traffic analysis. However, more complex algorithms and functions shall require special implementation techniques to achieve multi gigabit, real time traffic management. Further actions assume optimization of the hardware implementation of the detection algorithms to obtain the assumed processing speed. It will also be necessary to carry out some research to allow the parallel operation of the system to be able to support networks of more than 10Gbps. The possible applications for the results within the project are presented below:

- control and security management devices for internet networks at very high (10Gbit/s and more) data transmission speeds, using correlated detection methods,
- implementation of various data recognition and processing algorithms in the packets transmitted via the Internet on the basis of the efficient hardware platform developed in the project with programmed 10Gbit Ethernet fiber optics interface,
- using the hardware platform for tasks that require parallel processing (meteorology, flow modeling, etc.).

References

[1] Z. Bankovic, D. Stepanovic, S. Bojanic, and O. Nieto-Taladriz, "Efficient application of genetic algorithm and a dimension reduction technique for intrusion detection", *in Proc. 2006 International Conference on Engineering and Mathematics (ENMA 2006)*, 2006, pp. 303-308.

[2] D. Barbara, N. Wu, and S. Jajodia, "Detecting novel network intrusions using Bayes estimators", *In Proc. the First SIAM Int. Conf. on Data Mining (SDM 2001)*, Chicago. Society for Industrial and Applied Mathematics (SIAM), 2001.

[3] W. Lu and I. Traore, "Detecting new forms of network intrusion using genetic programming, *Computational Intelligence*, vol. 20, No. 3, 2004.

[4] G. Shafer, *A Mathematical Theory of Evidence*, Princeton University Press, Princeton, 1976.

[5] A. Skotarczyk and A. Chorazyczewski, "FastMatch System – semantic integration of threat detection methods in networks at high transmission rates", *in Proc. Conference INTERNET2008*, Wroclaw, 2008, in press.