AUTOMATED GENERATING OF PROCESSING ELEMENTS FOR FPGA

Ondřej Lengál
Bachelor Degree Programme (3), FIT BUT
E-mail: xlenga00@stud.fit.vutbr.cz

Supervised by: Martin Žádník
E-mail: izadnik@fit.vutbr.cz

ABSTRACT

Some information processing applications, such as computer network monitoring, need to continuously perform processing of rapidly incoming data. As the speed of the incoming data increases, it is desirable to perform the processing in the hardware. In this paper, we propose a configuration system that generates a VHDL specification of a hardware data processing circuit based on a user provided definition of data and computation operations. The system focuses on network traffic monitoring in multi-gigabit computer networks.

1 INTRODUCTION

As the volume and speed of electronic communication increase, the demand for processing of communication at top speed occurs. Current speed of communication disables its processing in the computer processor, due to the limited system bus throughput of data to computer memory and processing of the data by a generic processor. A large number of *application-specific integrated circuits* (ASIC) that enables data processing at high speeds exist; however, they suffer from such disadvantages as very restricted configuration options and high price when only a small amount of chips is produced.

Nevertheless, in practice there are applications that require both high data processing speed and high degree of configurability. An example of such an applications is IPFIX-based [1] network monitoring. *Field-programmable gate arrays* (FPGA) appear to be a convenient platform for these applications. The present paper proposes a system for automated generation of synthesizable VHDL description of processing elements according to the user's definition. The work primarily focuses on the hardware design of the *Flow Processing Unit* for Flexible FlowMon [2] network flow monitoring probe.

2 ARCHITECTURE

A rough outline of the Flexible FlowMon probe can be seen in Figure 1. The captured packet is processed in the preprocessing block, where the packet headers are extracted and combined into the *unified header* [3]. The FlowContext [4] stores the context for every flow (for the definition

of flow, see [1]) and binds proper context to the packet. When packet data are received by the FlowContext, the data (unified header, payload and packet flow context) are sent to the FlowContext endpoint, where they are accessed by the Flow Processing Unit (FlowPU). A detailed description of the Flexible FlowMon probe can be found in [2].

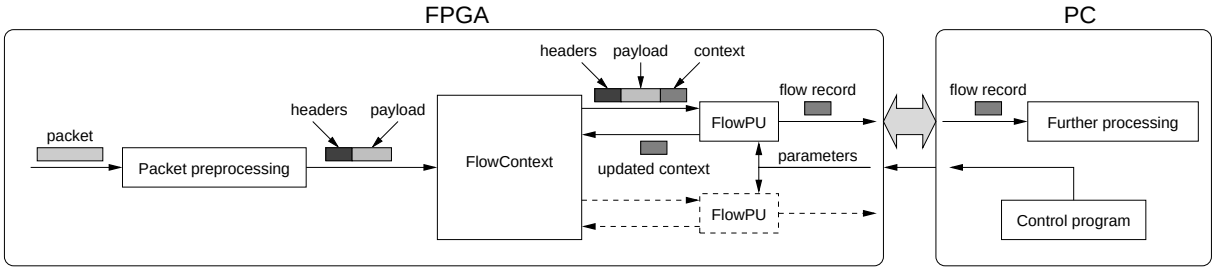

Figure 1: Architecture of the probe

The purpose of the Flow Processing Unit is to update flow context fields according to the user definition of the monitoring process. The update operations are carried out using a tailor-made computation engine generated by a core generator program. Expired flow contexts are sent to the PC in the form of flow records for further processing.
The architecture of the Flow Processing Unit is divided into two blocks with specific functions:

- **The preprocessing block** decodes the command that is sent together with the packet information and determines how the packet will be processed. This block also checks for collisions, i.e. it ensures that the context bound to the packet is correct. In case it is incorrect, the context is exported to the PC and a new one is created using default values.
- **The context update block** contains the computation pipeline that is configured to perform the user-defined update and control operations. The result of the update is also checked in this block and if a control condition is violated, the result context is exported for further processing to the PC.

3 PROCESSING UNIT CONFIGURATION

The user configuration of the processing unit is provided in the form of XML structured data file. The configuration file consists of the definition of the *unified header*, *payload*, *parameters*, *flow* and *controls* structures. The definition includes description of update operations (e.g. sum of packet lengths) and description of control operations (e.g. "export the flow context if the duration of the flow exceeds a certain time"); these are defined using C-like expressions.
The core-generator parses the input XML and operations definition, and constructs the data flow graph (Fig. 2) of the processing chain, with the input being the unified header and payload of the captured packet, parameters (set by the control program) and current flow context fields; the output is the updated context of the flow. The structure of the unified header and the context is then determined in the way that consumes minimal resources on the chip while achieving maximum throughput. The throughput of the unit is crucial because of the anticipated use of the network probe in 10 and 40 Gigabit Ethernet networks. The demand for such high-speed processing requires the application of pipelined architecture of the unit, which makes correct timing more complex.

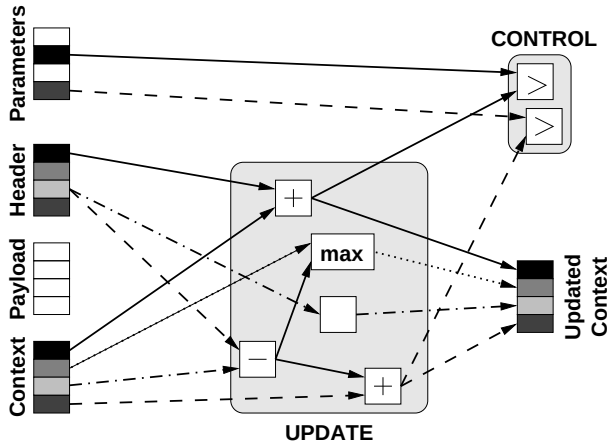

Figure 2: Example of the data flow graph.

4 CONCLUSION

The paper proposes an architecture and a configuration system for a high-speed user configurable processing unit. The primary purpose of the unit is to be used as a part of the Flexible FlowMon flow monitoring network probe. Moreover, the architecture is designed in a modular way, which means that the computation unit can be used as a part of different systems, provided a proper preprocessing block is used. The use of XML and C-like expressions for the definition of the computation process enables high level of configurability and employment of semantic aware optimization algorithms.

ACKNOWLEDGEMENT

This project was supported by Institutional research plan MSM6383917201 in the frame of research activity Programmable hardware of CESNET, association of legal entities.

REFERENCES

[1] J. Quittek, T. Zseby, B. Claise, and S. Zander, "RFC 3917: Requirements for IP Flow Information Export (IPFIX)," Oct. 2004.

[2] M. Žádník, P. Špringl, and P. Čeleda, "Flexible FlowMon," Tech. Rep. 36, CESNET, Oct. 2007.

[3] T. Dedek, T. Marek, and T. Martínek, "High Level Abstraction Language as an Alternative to Embeded Processors for Internet Packet Processing in FPGA," in *2007 International Conference on Field Programmable Logic and Applications*, pp. 648–651, IEEE Computer Society, 2007.

[4] M. Košek and J. Kořenek, "FlowContext: Flexible Platform for Multigigabit Stateful Packet Processing," in *2007 International Conference on Field Programmable Logic and Applications*, pp. 804–807, IEEE Computer Society, 2007.