

# Configuration Tool and FPGA-Prototype of a Hardware Packet Processing System

Stephan Kubisch, Harald Widiger, Peter Danielis,  
Dirk Timmermann, Christian Lange, Oliver Roewer  
University of Rostock, Institute of Applied Microelectronics and CE  
18051 Rostock, Germany  
E-mail to: stephan.kubisch@uni-rostock.de

Daniel Duchow, Thomas Bahls  
Siemens Networks GmbH & Co. KG  
Siemensallee 1, 17489 Greifswald, Germany  
E-mail to: daniel.duchow.ext@siemens.com

**Abstract**—The packet processing system MATMUNI is presented. It is implemented on an FPGA board and configurable at runtime via a graphical configuration tool. Two application scenarios are demonstrated.

## I. INTRODUCTION

In modern network applications and especially in access networks, the demands towards functionality and throughput as well as on security and availability are rising permanently. Currently, important driving forces in the Internet are new transmission technologies [1], the growing number of Internet users, and oversubscription of transmission lines. Furthermore, telecommunication carriers have different and changing requirements concerning the network equipment. Hence, only hardware solutions provide sufficient performance for packet classification, manipulation, and forwarding. Due to their flexibility, FPGAs are widely used as target platform. We developed a working FPGA prototype for a packet processing system (PPS).

Section II presents our PPS. In Section III, we introduce the configuration tool. Section IV briefly sketches two demonstration scenarios before the paper concludes in Section V.

## II. MATMUNI – THE PACKET PROCESSING SYSTEM

The PPS is called MATMUNI. It is currently designed for a Xilinx Virtex-4 FX20 platform FPGA. MATMUNI's functional submodules offer mechanisms for medium access controller address translation (MAT), traffic management (TM), and a multi protocol label switching-user network interface (MPLS-UNI).

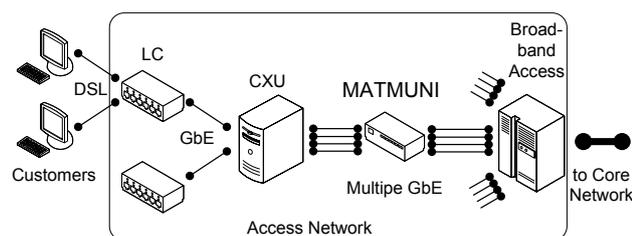


Fig. 1. Structure of an access network and logical location of MATMUNI

### A. General Functionality

As sketched in Figure 1, MATMUNI is located in the access network behind the line cards (LC) and the central switching unit (CXU). The CXU aggregates several thousand digital subscriber lines (DSLs). On independent Gigabit-Ethernet (GbE) channels, traffic of multiple Gbit/s must be handled. MATMUNI covers packet processing tasks that prepare and preprocess traffic for the core network.

**MAT** MAT targets scalability and security issues by performing a flexible n:1 translation of data link layer addresses—Ethernet

addresses of the medium access controller (MAC) in our scenario. Untrustworthy customer MAC addresses are translated into distinct, trustworthy MAC addresses of the provider.

**TM** The TM functionality meters traffic on a per customer basis to provide the level of QoS a customer has subscribed for. Each frame is color-marked either green, yellow, or red according to the customers' current bandwidth utilization. With policing mechanisms, the subscribed bandwidth is ensured as long as possible for each customer.

**MPLS-UNI** The MPLS-UNI encapsulates complete frames and inserts MPLS label stacks. Forwarding decisions of core routers solely depend on the labels' information. Usually, a full-blown label edge router (LER) is required. But in the case of MATMUNI, only a subset of LER functionality is necessary. Therefore, an adapted, compact MPLS module was realized in hardware.

For detailed information on MATMUNI's functionality, the interested reader is referred to [2], [3], [4].

### B. System Architecture

As pictured in Figure 2, MATMUNI is implemented as a pipelined data path architecture. Communication between the modules bases upon special interfaces. Two main data paths exist within the structure; one upstream path from the customers to the core network and one downstream path. A frame entering the system through the FPGAs internal MACs is first stored in a synchronization buffer for reasons of clock domain crossing. The frame is then forwarded to a framebuffer module. A key parser extracts lookup information from the frame's protocol headers and generates a request to the memory arbiter. The memory arbiter administrates the memory, which contains rules for each key. When the memory lookup was successful, the frame is byte-serially transferred through the functional modules. The functional modules modify the frame according to their intended task using the rules, which are assigned to the appropriate functional module. The modules are serially linked. Finally, the last module forwards the frame to the egress MAC via another synchronization buffer. The basic data flow is the same for up- and downstream data path. Up to 4 parallel main data paths are supported in both directions. Therefore, the functional submodules are instantiated twice or fourfold. Figure 2 depicts the internal structure of the MATMUNI system without the synchronization buffers and MACs.

MATMUNI's current architecture is designed to handle GbE. A frequency of at least 125 MHz is required for non-blocking performance. To meet different demands, MATMUNI is highly configurable at synthesis time, e.g., the actual functional spectrum, the number of Gbit channels, and many parameters regarding the individual functionalities that cannot be configured at runtime.

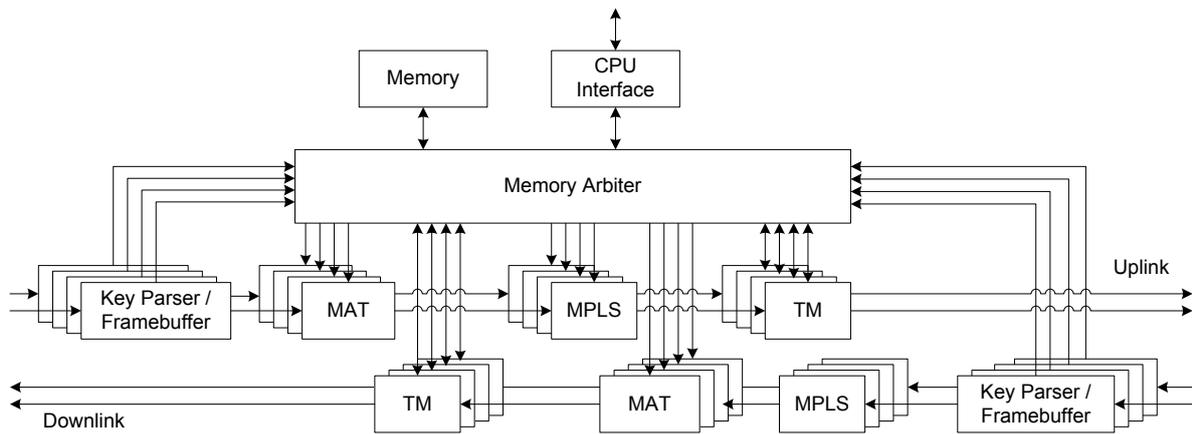


Fig. 2. Internal structure of the MATMUNI system with serially ordered functional modules

### III. THE CONFIGURATION TOOL

Additionally, the MATMUNI system contains an interface to a CPU for database management and system configuration (see Figure 2). The content of the memory—keys and their associated rules—and the system’s internal configuration parameters can be updated dynamically at runtime. Therefore, we developed a graphical user interface, which allows for an easy “push-button” configuration of the entire system. The screenshot in Figure 3 shows the graphical user interface (GUI) of the tool’s latest version. It is developed in C++ using the open source version of Trolltech’s Qt [5]. The first tab of the GUI allows for the configuration of global system parameters. Each of the other tabs is dedicated to the memory management of a certain functional module as MAT, TM, or MPLS.

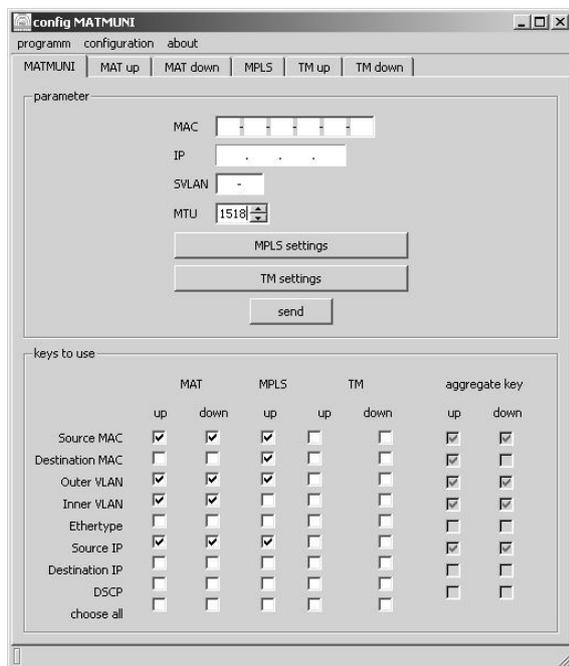


Fig. 3. MATMUNI configuration tool

### IV. SCENARIOS

MATMUNI is implemented on a Xilinx Virtex4 FX20 development board (ML-405). In the target application, which is

an IP digital subscriber line access multiplexer (IP-DSLAM), MATMUNI is implemented on a Virtex4-FX40 FPGA. Here, it reaches a throughput of 4 Gbit/s in both upstream and downstream. Due to the limited resources on the demonstration board, the prototype we are going to present provides a throughput of 1 Gbit/s in each direction. The typical, complex environment of MATMUNI on the IP-DSLAM cannot be rebuilt. Instead, two demonstrations have been prepared, which provide an insight into the operation modi of the different functional modules in the MATMUNI system.

**TM** This demonstration targets Quality-of-Service. Concurrent, independent media streams are metered and limited to a configurable bandwidth. Using the VLC media player, the Quality-of-Experience in streaming videos will be shown.

**MAT** This demonstration targets security. A MAC address flooding scenario is demonstrated with and without MAT functionality. With MAT functionality included in the data path, no MAC address flooding is possible.

### V. CONCLUSION

We presented the working prototype of the previously published packet processing system called MATMUNI. For comfortable system configuration, we also proposed a graphical configuration tool, which is used to configure the MATMUNI prototype on the FPGA development board at runtime. Furthermore, we briefly introduced two demonstration scenarios.

### ACKNOWLEDGMENT

We thank Siemens Networks GmbH & Co. KG, Greifswald, Germany, for support and cooperation in the MATMUNI project.

### REFERENCES

- [1] J. Cioffi et al., “Vectored DSLs with DSM: The Road to Ubiquitous Gigabit DSLs,” in *Proc. of the World Telecommunications Congress 2006*, Budapest, Hungary, April 30 - Mai 3 2006.
- [2] S. Kubisch, H. Widiger, D. Duchow, D. Timmermann, and T. Bahls, “Wirespeed MAC Address Translation and Traffic Management in Access Networks,” in *Proc. of the World Telecommunications Congress 2006*, Budapest, Hungary, April 30 - Mai 3 2006.
- [3] H. Widiger, S. Kubisch, D. Duchow, D. Timmermann, and T. Bahls, “A Simplified, Cost-Effective MPLS Labeling Architecture for Access Networks,” in *Proc. of the World Telecommunications Congress 2006*, Budapest, Hungary, April 30 - Mai 3 2006.
- [4] H. Widiger, S. Kubisch, D. Timmermann, and T. Bahls, “An Integrated Hardware Solution for MAT, MPLS-UNI, and TM in Access Networks,” in *Proc. of the 31st Annual IEEE Conf. on Local Computer Networks (LCN)*, Tampa, FL, USA, November 14-16 2006.
- [5] Qt by Trolltech, <http://www.trolltech.com>.