

# C-GEP: 100 Gbit/s Capable, FPGA-based, Reconfigurable Networking Equipment

Pál Varga, László Kovács, Tamás Tóthfalusi  
AITIA International Inc.  
Telecommunication Division  
Budapest, Hungary  
Email: {pvarga, lkovacs, tothfalusi}@aitia.ai

Péter Orosz  
Faculty of Informatics  
University of Debrecen  
Debrecen, Hungary  
Email: orospz@unideb.hu

**Abstract**—Programmable networking platforms are in the spotlight since the advent of SDN (Software Defined Networking). It is a great challenge to create such a platform - especially with reconfigurable hardware and line-rate capabilities reaching and exceeding 100 Gbit/s. These requirements together put FPGA (Field Programmable Gate Array) technology into the focus of high performance networking. In this paper, we introduce a highly flexible, programmable, multi-purpose networking platform, which is capable of hosting multiple 1 and 10 Gbit/s Ethernet interfaces - beside their 40 or 100 Gbit/s interface. The hardware of the introduced C-GEP platform is reconfigurable, even on-the-fly; due to the FPGA technology. C-GEP can host a wide range of high-speed network specific applications – including monitoring, switching and media conversion –, and it is aligned with the SDN principles. The system consists of two main building blocks: a high performance FPGA-based custom specific hardware platform and the firmware tailored to the actual task. The architecture is briefly introduced by its hardware and firmware setup, then some of the core functionalities, such as packet processing, filtering, and switching are presented.

## I. INTRODUCTION

Networking at ever-growing rates constantly generates challenges for researchers and engineers to develop equipment that handles the demands of networking services. The increasing rate is not the only concern – other transmission properties should be kept under tight control, sometimes even flow-level control is necessary. The principles of SDN (Software Defined Networking) allow high flexibility for control – although the underlying firmware and hardware should support such flexibility as well.

Networking hardware can be made *flexible* and *fast*, down to the bare metal, silicon, and fiber, by the application of FPGAs (Field Programmable Gate Arrays).

In this paper, we describe the architecture of a high performance, 100 Gbit/s-capable networking platform: C-GEP. The reconfigurable property of the FPGA chip enables to turn its hardware platform into a high performance networking device, e.g., network monitor, switch, router, firewall or intrusion detection system. The general packet processing capabilities of the system is described in this paper, together with showing its host *network monitoring* and *switching* functionalities.

As part of a network monitoring system, C-GEP supports distributed and lossless packet level monitoring of Ethernet links up to 100 Gbit/s. Distributed monitoring implies multiple

synchronized instances of the C-GEP device that we call probes in the context of network measurement. Reconfiguration of any part of a hardware-based network monitor got more focus by the emergence of SDN. The requirement for high throughput and high level of reconfiguration together ended up in the design and implementation of the presented system.

Regarding switching functionality, this paper presents how C-GEP enables the implementation of various lookup algorithms, including a specialized, buffer feedback weighted round robin (WRR) method. Furthermore, design and implementation considerations of VLAN tunneling, and lossless traffic compression/decompression are also briefly described.

## II. RELATED WORK

The C-GEP network processing platform is a direct successor to the C-Board architecture [1]. C-Board aimed for multi-FPGA network traffic distribution (by four Virtex-5 chips) at 10 Gbit/s, with an additional aim to achieve better power scaling through smart traffic routing to different FPGAs that are connected in a ring-like architecture [2]. C-GEP aims for a simpler but more powerful, single-FPGA packet processing platform, handling 100 Gbit/s traffic – using a Virtex-6 FPGA.

Regarding FPGA-based switching platforms, Asif Khan *et al.* [3] proposed a NetFPGA-based [4] OpenFlow switch. Since the implementation is ported to the NetFPGA-10G board, it is able to operate at 10 Gbit/s link speed. Using the 4 SFP+ transceivers the switch is able to operate at 160 MHz core frequency, combined with a 64-bit wide internal data path.

As a NoC (Network-on-Chip) realization, Ronny Pau *et al.* [5] proposed a router design on an Altera Stratix FPGA chip, supporting five network topologies. The architecture consists of a control logic and five bidirectional ports, where the switching is performed by two 3x3 crossbars. The dual-crossbar solution claims less resources within the FPGA chip.

Yutian Huan *et al.* [6] proposed a Split-Merge switch architecture on a Virtex 6 FPGA. The low latency, pipelined architecture is implemented for both the DOR (Dimension Ordered Routing) and the WSF (West-Side First) routing algorithms. Supporting high-performance network dependencies, the design is capable of running at 300 MHz frequency.

Ye Lu *et al.* [7] proposed a low-latency NoC router design, aimed for FPGA. The architecture is based on an optimized

pipeline organization, and it is scalable as well. The packet propagation latency of the proposed work is only two clock cycles long, including the routing and link traversal.

Nachiketa Das *et al.* [8] proposed a run time congestion aware router, implemented on a Virtex-II FPGA. The design estimates the congestion cost to determine the routing path. The routing algorithm uses a Java based API by Xilinx.

Yahia Salah *et al.* [9] proposed an FPGA-based NoC router. Its the design supports QoS, using a priority-based scheduler.

### III. ARCHITECTURE

C-GEP is a multi-purpose, programmable, FPGA-based Gigabit Ethernet Platform. The main purpose of the architecture is to provide means for handling 1, 10, 40 and 100 Gbit/s traffic in a flexible manner. It allows the implementation of various applications: SDN devices, media gateways, traffic generators, DPI (Deep Packet Inspection) support, and more.

Beside providing sufficient resources for switching and routing at 100 Gbit/s, the design includes some special, network monitoring-related requirements, namely

- lossless packet capture,
- 64-bit timestamping with 3.2 ns resolution,
- header-only capture: configurable depth of decoding,
- on-the-fly packet parsing by hardware,
- parameterized packet/flow generator for mass testing.

#### A. Hardware

The modular view of the C-GEP hardware is shown in Figure 1. The heart of the platform is a high capacity, Virtex 6 FPGA chip. The COM Express PC mainly hosts management functionalities; moreover, it can also cover various, Software Defined Networking (SDN) control functionalities, when the FPGA acts as a pure forwarding plane entity. The 2 x DDR3 slots support buffering and temporal data storage.

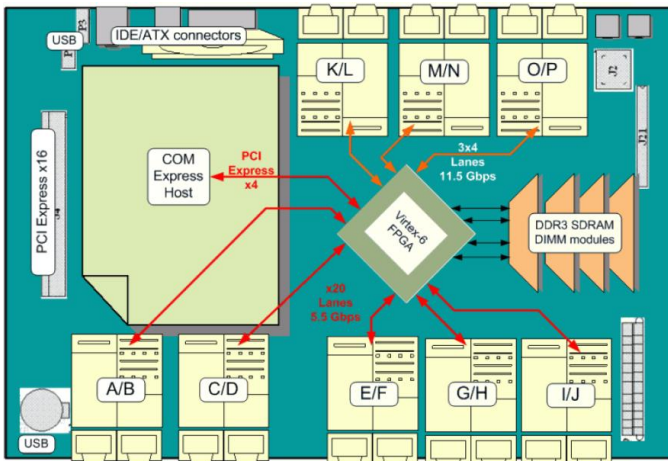


Fig. 1. Hardware Architecture of C-GEP

- Depending on the environmental requirements,
- K-P interfaces can realize 1x100 Gbit/s, or 3x40 Gbit/s, or 12x10 Gbit/s configurations, whereas

- A-J interfaces can realize 20x1 Gbit/s, or 5x10 Gbit/s configurations – or variations of these.

This flexibility allows the platform to cover various needs, such as

- 1x100 Gbit/s and 5x10 Gbit/s,
- 3x40 Gbit/s and 5x10 Gbit/s,
- 12x10 Gbit/s and 20x1 Gbit/s,
- and many other configurations.

#### B. General Firmware

The firmware architecture is highly modular, separated into two main module types by design:

- interface module (dependent on the architecture of the board), and
- user application module (connecting/controlling interface modules; or realizing various application functions).

1) *Interface modules:* In order to utilize the actual hardware interfaces of C-GEP, various interface modules are provided within the firmware. These include 1, 10, 40 and 100 Gbit/s Ethernet device specific transceiver modules; DDR3 memory interfaces; MDIO/i2c/serial port handlers for management; PCI-express x4 endpoint block module; and the chipscope interface for debugging.

2) *User modules by application:* There are various user modules included in the firmware in order to support the network-on-chip functionalities. Among others, these include 1, 10, 40 and 100 Gbit/s MAC modules; the PCI-express controller; DDR3 memory controllers; TCP stream transmitter (to stream data between C-GEP and the data post-processors); time synchronization; and management modules.

Various applications then require other supported functionalities. As an example, the high-speed monitoring application consists of the following sub-modules:

- timestamping every frame upon reception;
- packet decoding from layer 2 up to the application layer;
- packet filtering with a reconfigurable rule-set to decide what we do with a given packet;
- packet chunking: packets can be truncated depending on the matching rule;
- packet distribution: to distribute packets by different criteria: IP flows, fragment steering, steering based on mobile core network parameters, etc.;
- packet headering: monitoring information is stored in a specified header format.

#### C. 100 Gbit/s Ethernet MAC

C-GEP has its own 100 Gbit/s PCS/PMA (Physical Coding Sublayer/Physical Medium Attachment) and MAC (Media Access Control) protocol stack implementation provided by AITIA. The implementation is fully compliant to IEEE 802.3ba-2010 [10]; and the Intellectual Property rights (IP-core) are available for others to use in binary or source-code format, under licensing conditions. The 100 Gbit/s modules are field-tested on C-GEP with CFP SR optical modules.

The MAC uses Xilinx Virtex 6 device-specific GTH (Gigabit-Transceiver H) transceivers, but can be easily ported

to other devices, as well. User interface has a 512 bit/320 bit Receive/Transmit MAC data-width running at 312.5 MHz clock frequency. It has an FPGA-optimized Ethernet CRC checksum calculation for Rx and Tx MAC interfaces.

All major building modules of the 100 Gbit/s Ethernet core are well separated, according to their functions. The modular setup (illustrated by Figure 2) resulted in a code that is easy to understand and develop.

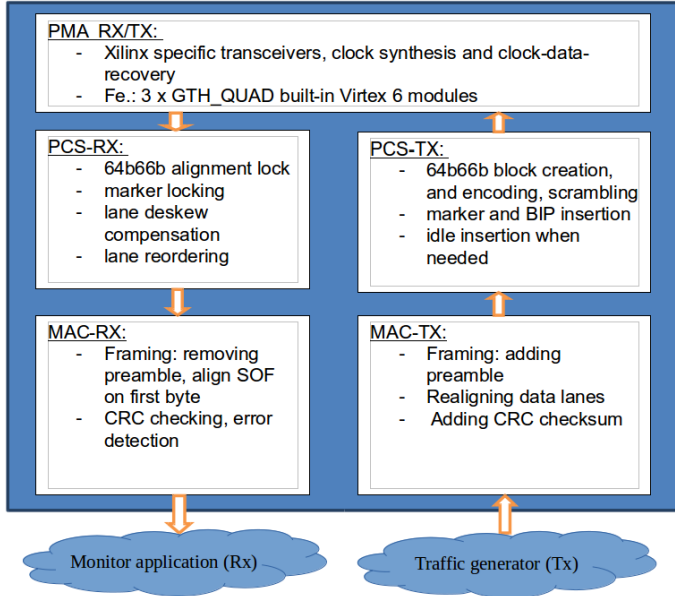


Fig. 2. C-GEP-100Gbit/s PCS/PMA and MAC block diagram

### 1) Receive modules of the 100G core:

- PMA\_RX\_100G (Rx path): is the device specific physical interface of the 100G core. In the case of C-GEP it is a Virtex 6 device – although if another FPGA device (e.g. an Altera chip) is used, this is the only module to be replaced – the other core parts are not dependent on the physical interface;
- PCS\_RX\_100G: implements the RX PCS/PMA functions according to [10];
- MAC\_RX\_100G: aligns frame data to always start on the first byte; cuts off Ethernet preamble; implements 320 bits to 512 bits data-width conversion to compensate speed differences; and calculates Ethernet frame checksum.

### 2) Transmit modules of the 100G core:

- PMA\_TX\_100G (Tx path): is the transmit path of the physical interface;
- PCS\_TX\_100G: implements TX PCS/PMA functions [10];
- MAC\_TX\_100G: is responsible for assembling the Ethernet frame from the raw input data – adding preamble, and checksum; working with either 320 bits or 512 bits data-width, depending on the application.

3) *Checksum calculation and verification:* FPGA-based CRC implementations get more complicated, and harder to implement as the operating frequency and bus width gets

higher and higher. At 312.5 MHz and 512 bit data width traditional parallelized LFSR is nearly impossible to realize, because of the high input-feedback bit dependency count. In this realization, we had to mathematically optimize the working principle to meet timing and placement needs.

### D. High Precision Time Handling

High resolution timestamping is one of the essential requirements that a monitoring platform has to meet, in order to detect and reconstruct network events correctly. Monitoring on multiple links of a network device needs precise synchronization between the monitoring probes. The probes always have to be prepared for the worst case and therefore apply external clock sources to maintain high time accuracy. The C-GEP platform enables several clock synchronization solutions (e.g., NTP, PTPv1, GPS, atomic clock). It is able to operate as a PTPv1 slave device, this implementation is fully compliant with the IEEE 1588-2002 standard. The implemented architecture operates with an internal controller module for smooth clock adjustment and realizing precise time synchronization between the monitoring probes. The controller module is also able to cooperate with an external GPS device. Considering the worst case (e.g. synchronization signal loss), the platform can operate with an atomic clock source (e.g., SA45CSAC atomic clock [11]). The Physical Coding Sublayer of 100Gbit/s Ethernet dictates at least 6.4 ns resolution for the controller and the timestamping modules.

## IV. HOSTING NETWORKING FUNCTIONALITIES

The architecture of the C-GEP device provides a general platform for measuring network components. As the platform is a combination of high-speed interfaces, a reconfigurable hardware chip and an embedded computer, it can be used for various specific tasks.

Monitoring user behavior and dimensioning network links are extremely important functions for network operators. Configuring C-GEP as a monitoring platform provides an opportunity to measure traffic-mix and important network attributes. Once the traffic flow is identified (e.g., based on 5-tuple) various metrics can be calculated (packet loss, packet delay, jitter).

Besides the monitoring and analyzing various numerical properties, the device is able to operate as a traffic generator. Configuring the platform as an IP packet generator, it can generate multi-encapsulated (e.g., VLAN/MPLS/IP/UDP/GTP/IP/UDP) traffic internally to stress test routers, servers or firewalls by well controlled extreme network traffic load. Operating in generator mode, all interfaces can simultaneously utilize the maximum bandwidth. The content of the generated traffic is either stored in internal memory, or it can be calculated on the fly.

### A. Network Monitoring

In this section, we introduce a high-performance network monitoring system, operated on the C-GEP platform. Figure 3 represents the internal module architecture of the firmware,

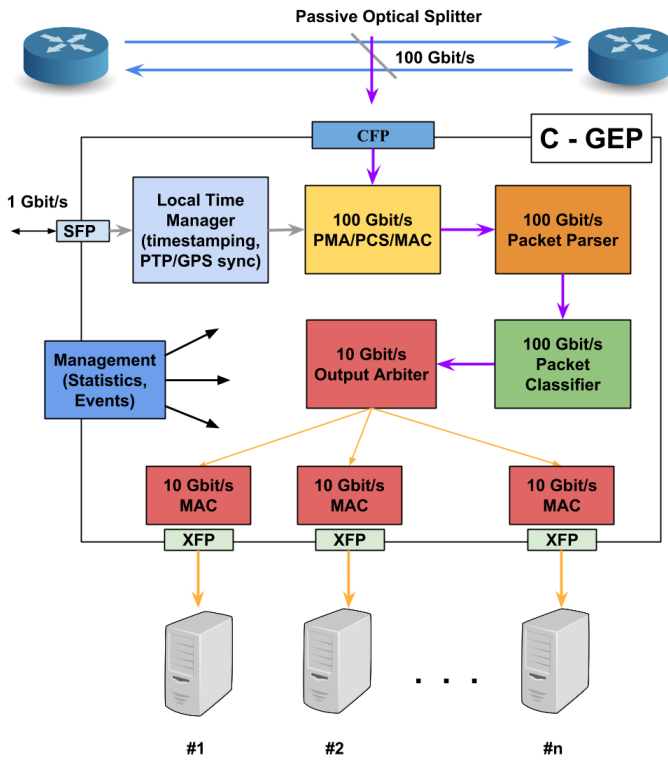


Fig. 3. Internal architecture of the monitoring platform

implemented within the FPGA chip. Each new packet arrived on the 100 Gbit/s interface receives a precise timestamp from the Local Time Manager module immediately. The timestamp is stored in a temporal storage element, and propagated through the whole processing phase as an extension header of the packet. After passing the MAC layer modules, the packet is handled by the parser and the classifier modules for protocol decoding and filtering. The monitoring firmware is able to decode even 14 multi-encapsulated protocol headers (e.g., QinQ, 2-level MPLS, GTP-U, IPinIP). Using as a preprocessing engine, it can effectively support malfunction detection or deep packet inspection processes. The platform is able to classify in real-time and drop or propagate the incoming packets. Matching packets are arbitrated to 10 Gbit/s link speed interfaces.

### B. Packet Processing

The heart of the C-GEP device is a Virtex-6 FPGA chip, that handles the time-critical packet processing tasks (e.g., reception, parsing, classification). Packet Processing in hardware level can be divided up to three main phases: packet parsing, packet classification and output arbitration.

The main task of the packet parsing engine is to determine the protocol hierarchy and the embedded header structure. The implemented parser module operates on a parse graph, which is a predefined header embedding scheme for handling multi-encapsulated packets. The parser graph is re-programmable in compilation time, and supports scalability for new protocol

appearance. The actual parse graph implementation can handle even 14 multi-encapsulated protocol headers. Since our 100 Gbit/s MAC module implementation operates on a 512-bit wide data path, a new minimum-sized packet can arrive in every clock cycle. To apply a loss-less parsing engine, a hand optimized pipeline architecture was implemented. The pipeline stages operate on predefined header fields, and the architecture enables a scalable, 100 Gbit/s throughput capable platform.

Classifying the captured traffic is one of the main functions of a monitoring system. The classification engine is based on a reconfigurable rule set, but also expects input information from the parsing phase. The monitoring firmware of the C-GEP device is able to operate on a 14-tuple based rule set, where AND or NOT(AND) operations can be applied between metadata. The matching packets are forwarded or dropped, depending on the action defined by the matching rules.

A basic feature of the monitor firmware is the real-time rule set reconfiguration property, without the need to propagate the traffic into a default route, or dropping packets during the uploading process. To apply these functions, the rules are organized in a pipeline architecture, where each pipeline stage – referred to as rule-stage – corresponds to one rule.

Each pipeline stage can be divided into four substages, according to the internal operations of the rule fitting process: input buffering to achieve the timing constraint (S1), fitting the packet header informations to the current rule (S2), summarizing the results of the second substage (S3), and finally, making decision on the current and on the previous rule-stage results (S4). These substages (S1, S2, S3, S4) constitute an internal pipeline within one rule-stage, illustrated by Figure 4. Each substage can operate on a minimum-sized packet, in worst case.

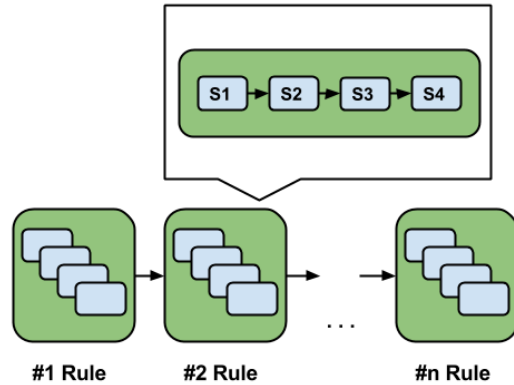


Fig. 4. Pipeline architecture of the classification engine

### C. Reconfiguration of Filtering Rules

The positive side effect of the classifier pipeline architecture is not only the scalability, but the lossless reconfiguration, as well. The monitor firmware dedicates multiple paths between the rule-stages for information propagation (e.g., rule reconfiguration, extra information, original data path). The reconfiguration-path is a dedicated path in the classification

pipeline. This allows uploading a new rule set without traffic loss and inconsistent states. While reconfiguration is in progress, traffic redirection to a default path is not necessary, and the capturing does not have to stop. To assist easier control over the C-GEP device, the platform contains a GUI for rule reconfiguration.

After receiving the final rule during reconfiguration, the whole rule set get read into the classification pipeline engine. Because of the substage operations within a rule stage, the upload procedure never results in an inconsistent state during the filter operation. When a packet enters a rule stage, the substages will operate on that rule content, which was active when the packet entered the substages. During reconfiguration, if a substage is operating on a packet, the next substages will operate on the previously active rule content. After the reconfiguration clock cycle, a new packet entering the rule stage will be classified according to the new rule. The reconfiguration process always overwrites the whole rule set with the actual content.

## V. REALIZING SWITCHING FUNCTIONALITIES

Packet switching is basically the process of assigning an incoming packet to an output interface. Incoming packets are assigned to output interfaces by predefined rules, and written into temporary buffers/FIFOs waiting to be transmitted (see Figure 5). In a wider sense packet tunneling, or changing protocol specific header content are considered as extended features of packet switching.

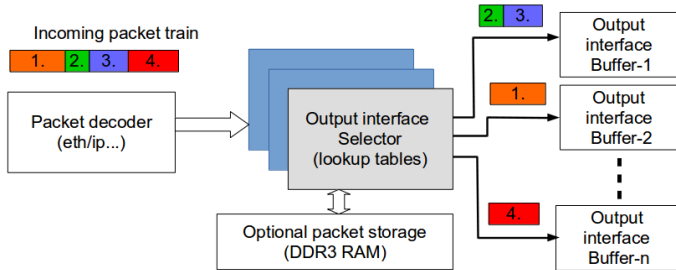


Fig. 5. A modular view of the generic packet switching function

Besides executing simple switching rules where the input-output mapping is pre-defined by specific protocol-header information, C-GEP implements some extra functionalities, tailored for network monitoring. In the following we describe some of these special, intelligent switching features.

### A. Traffic steering/switching for monitoring post-processing

The main purpose of traffic steering for post-processing is to evenly distribute incoming packets to output interfaces – with the requirement of some (related) packets sent to the same output. As Ethernet frame sizes range from 64 bytes to 1500 bytes, and even bigger if considering jumbo frames, a simple round-robin distribution may result in overloaded, and underloaded post-processors at the output.

The traffic distribution rule of packet scattering based on *buffer feedback* weighted round-robin (WRR) results in a quite

even output distribution. This means that the saturation of the output interface buffers determine where to route the actual packet. If all outputs have a similar buffer load, then this results in a round robin decision.

Main problem with evenly distributed traffic is, that most monitoring and traffic processing applications are independent, and have to receive related packets. This means, that related packets must be transmitted on the same interface, so usually a governing parameter of a specific protocol defines distribution.

Depending on the realized functionality, various parameters can define inter-relation of packets. Among others, the intelligent switching can be specified by

- 3-tuple: source/destination IP addresses, and layer3 protocol fields;
- GTP TEID tag (GPRS Tunneling Protocol, Temporary Equipment ID, for mobile network traffic analysis);
- VLAN/MPLS tag (for specific users/network segments);
- IP fragment ID (for fragmented IP packet reassembling);
- PPoE (Point-to-Point over Ethernet) port number; etc.

These rules usually do not guarantee a smooth traffic distribution, so the WRR mechanism should be further optimized. Traffic with near-constant data rate is simple to handle as buffers are emptied more evenly in time. Since the inter-related IP-traffic (associated by 3-tuples) tend to have short bursts in their data transfers, there are large enough buffers needed to smooth the even output-distribution in the long term. This means, that beside the present buffer load, WRR should take into account the average and the deviation of buffer load, too.

### B. VLAN tunneling

VLAN (specified by IEEE 802.1q) is a layer 2 switching protocol, used to create logically segmented networks in one physical LAN domain, or in multiple LAN domains.

VLAN types can be grouped by membership:

- port-based: the physical port on the device defines the VLAN number;
- MAC address-based: the sender’s MAC address defines the VLAN number (meaning an 1v1 address mapping, or an address range);
- IP address-based: the sender’s IP address defines the VLAN number. The same rules apply as for the earlier point, but IP subnets allow even bigger groups of related hosts to be covered with less rules.

C-GEP can be realized with high variety of interface combination – as an example: with 4x10Gbps inputs there can be 1-4 VLAN groups formed by port number. Although 1v1 mapping is implemented, the lookup delay at 100 Gbit/s limits the number of such rules.

### C. C-GEP-100 generic tunneling implementation

The C-GEP FPGA-core supports one layer of static VLAN-header insertion (and removal) at the moment, from nx10 Gbit/s to 100 Gbit/s and 100 Gbit/s to nx10 Gbit/s directions.

The VLAN ruleset implementation and VLAN tagging is based on a simple procedure. It is done by examining the incoming IP packets source (or destination) address, looking

up the corresponding VLAN ID, and inserting it into the Ethernet frame. In order to keep the processing delay constant, multiple IP-to-VLAN-mapping instances are used.

Figure 6 depicts the modular diagram of the VLAN injector in the nx10 Gbit/s to 100 Gbit/s direction. VLAN header removal is implemented with similar considerations in the opposite direction, as well.

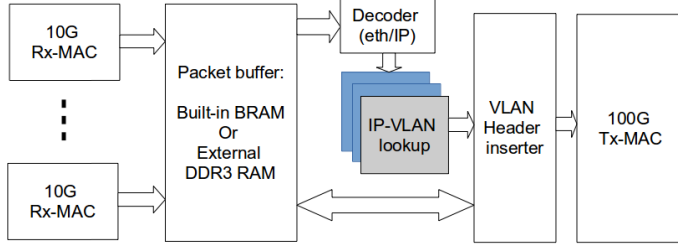


Fig. 6. VLAN injector block diagram (nx10Gbit/s to 100Gbit/s)

#### D. Lossless traffic compression/decompression

Data compression can be very important in processing data carried over 100 Gbit/s backbones. The main idea is, that the aggregated traffic from the 100 Gbit/s link is compressed on the fly, and distributed to 10 Gbit/s interfaces. The depth of compression can vary from layer 2 (everything is compressed) to the application levels. The 10 Gbit/s output streams are independent from each other, and use common LZ77 compression method. At the other side, another 10 Gbit/s-capable device decompresses the data on-the-fly. This results in highly bigger net payload traffic (approx. 1.4 to 2 times speed, depending on the type of data). Figure 7 depicts the modular diagram of the C-GEP implementation for this feature.

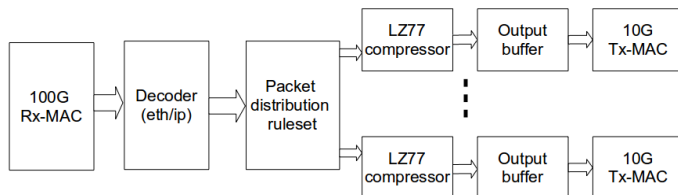


Fig. 7. Block diagram of the traffic compression system

## VI. CONCLUSION

C-GEP is a versatile programmable platform capable of handling 100 Gbit/s Ethernet traffic. It provides a base platform for various packet processing applications such as switching, routing, filtering, monitoring, etc. Its modular structure allows also the connection of additional post-processing devices in order to extend the application fields by using high-speed software processing.

The evolution of network infrastructures is driven by two factors: the increasing amount of user data and the emergence of new services. In order to give performance as well as flexibility, reconfigurable hardware may appear as the central building block of a high performance network management

system. In this paper, we introduced a multi-purpose, programmable, FPGA-based hardware platform that supports the mentioned new concept in many ways. Its main purpose is to provide means for handling 1, 10, 40 and 100 Gbit/s Ethernet traffic in a flexible manner. Beside introducing the architecture, we presented some features of C-GEP, including hardware-accelerated packet-capturing, -filtering, -classification, clock synchronization, and switching engines operated at 100 Gbit/s line rate.

The high-performance, programmable C-GEP platform can be widely used by operators, research engineers and application developers in order to tackle the challenges of effective traffic handling at high line rates.

## VII. ACKNOWLEDGEMENT

We would like to thank to all of our colleagues and students who contributed to the success of our project: Ákos Máté, Péter Mercse, Zoltán Nagy, István Pógár, Tamás Skopkó, Máté Varga, Péter Varga, János Végh from the University of Debrecen, György Horváth, Gábor Kródi, Péter Tatai, and the rest of the AITIA team.

This research was partially funded by the national project C-GEP, GOP-1.1.1-11-2012-0031, in Hungary.

The AITIA part of this research was partially funded by the national project DIANA, KMR\_12-1-2012-0207, in Hungary.

## REFERENCES

- [1] I. Moldovan and P. Varga, "A Flexible Switch-Router with Reconfigurable Forwarding and Linux-based Control Element," in *IEEE 10th International Symposium on Electronics and Telecommunications (ISETC)*, Timisoara, Romania, 2012, pp. 217–220.
- [2] P. Varga, I. Moldovan, D. Horvath, and S. Plosz, "A Low Power, Programmable Networking Platform and Development Environment," in *Network-Embedded Management and Applications (NEMA)*, Niagara Falls, Canada, 2010, pp. 19–36.
- [3] A. Khan and N. Dave, "Enabling Hardware Exploration in Software-Defined Networking: A Flexible, Portable OpenFlow Switch," *21st Annual International IEEE Symposium on Field-Programmable Custom Computing Machines*, pp. 145–148, April 2013.
- [4] NetFPGA, "NetFPGA-1G." [Online]. Available: <http://netfpga.org>
- [5] R. Pau and N. Manjikian, "Implementation of a Configurable Router for Embedded Network-on-Chip Support in FPGAs," *Joint 6th International IEEE Northeast Workshop on Circuits and Systems and TAISA Conference*, pp. 25–28, June 2008.
- [6] Y. Huan and A. DeHon, "FPGA Optimized Packet-Switched NoC using Split and Merge Primitives," *International Conference on Field-Programmable Technology*, pp. 47–52, December 2012.
- [7] Y. Lu, J. McCanny, and S. Sezer, "Generic Low-Latency NoC Router Architecture for FPGA Computing Systems," *International Conference on Field Programmable Logic and Applications*, pp. 82–89, September 2011.
- [8] N. Das, P. Roy, and H. Rahaman, "Runtime Congestion and Crosstalk Aware Router for FPGA Using Jbits3.0 for Partial Reconfiguration Application," *International Symposium on Electronic System Design*, pp. 146–151, December 2011.
- [9] Y. Salah and R. Tourki, "Design and FPGA Implementation of a QoS Router for Networks-on-Chip," *3rd International Conference on Next Generation Networks and Services*, pp. 84–89, December 2011.
- [10] *802.3ba-2010 – CSMA/CD Access Method and Physical Layer Specifications Amendment 4: Media Access Control Parameters, Physical Layers, and Management Parameters for 40 Gb/s and 100 Gb/s Operation*, IEEE Standard 802.3ba-2010.
- [11] Microsemi, "Quantum SA.45s Chip Scale Atomic Clock." [Online]. Available: <http://www.microsemi.com/products/timing-synchronization-systems/embedded-timing-solutions/components/sa-45s-chip-scale-atomic-clock>