

CESNET Technical Report 7/2013

Designing a Card for 100 Gb/s Network Monitoring

ŠTĚPÁN FRIEDL, VIKTOR PUŠ, JIŘÍ MATOUŠEK, MARTIN ŠPINLER

Received 18. 12. 2013

Abstract

This technical report describes the design of hardware accelerator for 100G Ethernet network security monitoring. The hardware is a PCI Express gen3 X16 board with a single 100G optical Ethernet interface and uses the Virtex-7 FPGA. In addition to the hardware, the report also presents some important blocks of the FPGA firmware: 100G Ethernet block and the NetCOPE Development Platform. The hardware, together with some additional firmware and software is intended to be used for the CESNET2 network border lines security monitoring.

Keywords: 100GE, Ethernet, FPGA, monitoring, security, NetCOPE, SDM

1 Introduction

The amount of data transferred by the CESNET2 network continually grows. This growth follows the overall trends in the Internet traffic development and is caused by the variety and growing bandwidth demands of individual Internet applications. The demand of bandwidth includes the local area networks, therefore the development of the latest Ethernet standard IEEE 802.3ba [3] was started in 2007. The standard was ratified in June 2010, and it includes two link rates – 40 and 100 Gb/s. To keep pace with this evolution, it is necessary to upgrade the core elements of the network: routers, switches, optical network lines and others. But in addition to these basic building blocks, it is highly desired to upgrade the security infrastructure as well.

CESNET has a long history of developing custom boards for hardware acceleration of Ethernet networking tasks. These accelerators have shaped to the form of network security monitoring devices in recent years. The CESNET2 network is equipped with a hardware-accelerated flow monitoring probe at every external network line. These probes measure and report unsampled flow statistics for the complete foreign traffic of the network. Since the peak traffic of the most heavily utilized external line can currently easily reach 15 Gb/s during the 5 minute interval (the line uses 4x10G Etherchannel split), it is clear that the upgrade of the monitoring infrastructure will be required.

In 2010 we introduced [1] the first generation of the hardware accelerator intended for the 40/100 Gb Ethernet. This hardware fell within the ComboV2 card family: the base was the ComboHXT card with the ComboI-100G1 interface. However this architecture exhibited a lot of limitations, for example the PCI Express throughput of 12 Gb/s, insufficient FPGA resources and only one possible type of Ethernet interface (100GBASE-SR10).

This paper reports our progress in developing a new generation of the hardware accelerator with a single 100 Gb/s Ethernet interface, based on recent technologies: Xilinx Virtex-7 FPGA with 25 Gb/s serial transceivers, versatile CFP2 optical interface and PCI Express generation 3 interface. We describe the overall architecture (Section 2.1), the hardware itself (Section 2) and some important and unique firmware modules: the NetCOPE Development Platform (Section 3) including the 100G Ethernet blocks and PCIe core with the DMA module.

2 Card Design

2.1 Architecture requirements

Our aim is to propose a modern cost-effective programmable card, particularly for 100 Gb Ethernet network monitoring applications. The card should also enable the implementation of other related networking applications: filtering, store and replay, packet generation and low-latency communication. Therefore the basic requirement is the programmability and the wire speed packet processing ability. The only technology offering such qualities is the FPGA (Field-Programmable Gate Array). The recent FPGAs are high-performance devices supporting many communication protocols and interfaces, enabling the almost single-chip design of such applications, which simplifies the board layout and reduces overall cost.

The next requirements result from our Software Defined Monitoring[4] (SDM) concept, which is the basis of our metering applications. SDM is a concept of network data processing which combines a control part running in software with a powerful hardware formed by the FPGA firmware. The key idea is to enable different types of hardware accelerated preprocessing and aggregation of network traffic based on actual needs of software applications. In order to achieve this, SDM firmware must be able to store relatively large amount of aggregation records and rules controlling the preprocessing for different kinds of traffic. Therefore, an external memory with storage capacity of hundreds of Mb is required. The external memory for aggregation records should also have low access latency, because updates of these records are done in read and write-back manner. The FPGA must also enable the reception of full 100 Gb/s of network data from the link and also the sending of full 100 Gb/s of data through the PCI Express to the host memory.

The SDM is built on the NetCOPE platform. NetCOPE is a hardware abstraction layer destined for rapid development of networking applications. The hardware is supposed to be a programmable card for the PC, with at least one network interface. NetCOPE is composed of a software part - drivers and supporting tools, and a firmware running in the FPGA, hiding the implementation of network interface, PCIe interface, DMA controller and other auxiliary modules. The NetCOPE platform is already implemented for many cards developed by CESNET, starting from the Combo6X cards, including the ComboV2 family, ending with the new card being described here.

2.2 Detailed architecture

With respect to the NetCOPE and SDM requirements we proposed a card, with block schematic shown on Figure 1. The design abandons the "sandwich" concept of a powerful base card and specialized interface card, as known from previous Combo families. The reasons are largely technological: it is not feasible to connect two cards using a serial data links of necessary throughput. Also the cooling is a difficult problem, because it is not possible to guarantee the adequate airflow inside the sandwich. There are also economical reasons: it is much cheaper to develop and fabricate only one card instead of two, containing expensive interconnect.

The heart of the card is the FPGA, serving for firmware implementation. The FPGA is connected to one network interface - a connector with a shielding cage for plugging an optical module. The FPGA is also connected to the PCI Express connector, static and dynamic memories, clock source, a coaxial connector for synchronization pulse reception and other support circuits.

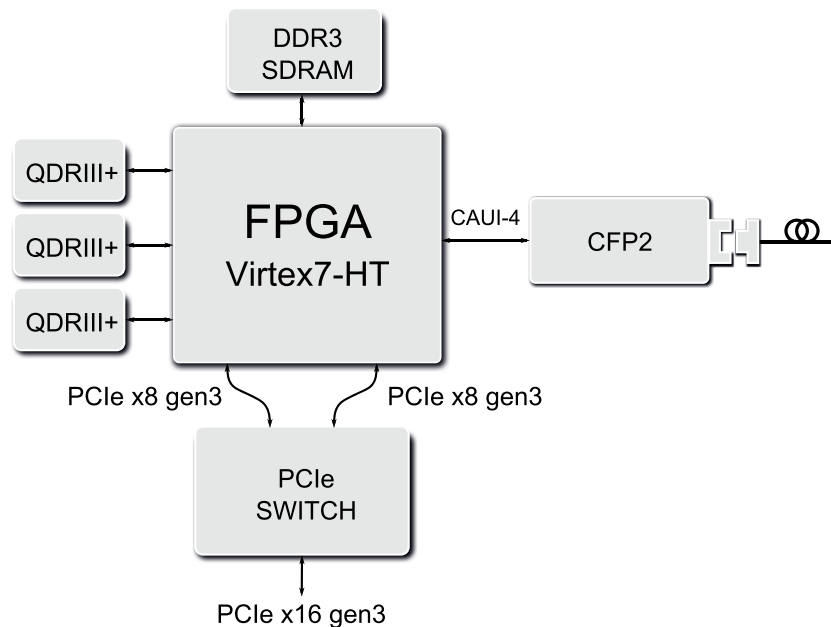


Figure 1. 100 GbE card block schematic

2.2.1 FPGA

The FPGA is a programmable logic structure – array of programmable logic elements with a flexible interconnect matrix. The basic logic element is a look-up table (LUT). The array of LUTs is supplemented by flip-flops (FF) and other more complex structures such as RAM blocks, DSP cells, clock management blocks and fast serial transceivers. The serial transceiver basis is a powerful serializer and deserializer with the clock data recovery (CDR) circuit and analog signal processing gear (serial line drivers and receivers, equalization).

To satisfy resource and throughput requirements of our firmware applications, the FPGA with the sufficient amount of logic resources, serial bandwidth and IO connectivity must be chosen. Because of our previous experiences with the platform, we focused on Xilinx products. As the right candidate we have found the

Virtex7-HT series FPGAs, which are the only FPGAs containing the GTZ type serial transceivers, supporting link rates up to 28 Gb/s. The four of them can form the CAUI-4 interface, thus the CFP2 or CFP4 module can be connected directly. Moreover the Virtex7-HT FPGAs also include up to 3 PCI Express endpoint blocks, which, in conjunction with the GTH transceivers, implement all of PCI Express protocol layers in generation 3 and width up to 8 lanes. The resources available in Virtex7-HT FPGAs are summarized in Table 1.

Table 1. Resources available in the Virtex7-HT FPGA devices

Device	LUTs	FFs	BlockRAMs	PCIe	GTH	GTZ	I/O
XC7VH580T	580480	725600	940	2	48	8	600
XC7VH870T	876160	1095200	1410	3	72	16	650

Our estimations shows that the complete SDM implementation will consume about 250 000 LUTs and FFs, 300 RAM blocks, 4 GTZ transceivers (Ethernet interface), 16 GTH transceivers (PCI Express) and 500 IOs (memories, clocking and others). Thus, the XC7VH580T FPGA will be adequate. If necessary, the package pinout makes possible to equip the board with the larger one FPGA as well.

2.2.2 Network interface and optical module

There are two interface types on optical modules: towards the MAC/PCS/PMA is the *electrical interface*. According to the signal conditioning requirement is this interface designated either CAUI or CPPI for the 100 GbE and either XLAUI or XLPPi for the 40 GbE. Towards the fiber the *optical interface* is in the form of LC/SC or MPO connector.

The first generation of 100 GbE, available from 2010, was based on well-proven 10G technologies. Mainly the electrical interface was specified only as 10×10 Gb, therefore modules using 25 Gb link rates contained a "gearbox" intended for 10 Gb and 25 Gb lane multiplexing and demultiplexing. For signal conditioning the clock-data recovery (CDR) circuit was common.

The main representative of the first generation optical module is the C-Form Pluggable (CFP). CFP supports all of optical media types: ten lane (10×10 Gb/s) as well as four lane (4×25 Gb/s, 4×10 Gb/s). The signaling rate of the electrical interface is limited to N×10 Gb/s, therefore the CFP modules include a 10:4 gearbox and the CDR circuit. In consequence these modules are relatively large (145mm×82mm×14mm) and high power consuming: the consumption can reach up to 32 W. The more energy saving and smaller modules are CXP (100GBASE-SR10) and QSFP (40GBASE-SR4), however these modules are not versatile and support only limited set of media types.

The second generation of 100 Gb optical modules uses 25 Gb signaling rates on the electrical interface (CAUI-4), which cancels the need of a gearbox. The simplification of module structure brings reduction of the number of sub-components, decrease in area and power and cost reduction. The successor of the CFP is the CFP2 module with dimensions of 108mm×42mm×12mm. The optical interface can be the LC/SC connector for long range WDM media types or the MPO/MTP for

parallel optic media. The peak power consumption is 12 W. In the lowest power class it is possible to operate even without a heatsink.

The near future successor of the CFP2 is the CFP4, where the further electronic improvements allow to continue with the process of integration. The dimensions are only 88mm×22mm×10mm and the power consumption should not exceed 6 W. All of the optical interface types available today are summarized in Table 1.

Table 1. 40/100GE optical PMD types and transceiver modules.

Type	Line rate	Reach	Fiber type	Fiber pairs	Lambdas per fiber	Signaling rate	Optical modules
100GBASE-LR4	100 Gb/s	10 km	SM	1	4	25.78125 Gb/s	CFP, CFP2, CFP4
100GBASE-ER4	100 Gb/s	40 km	SM	1	4	25.78125 Gb/s	CFP, CFP2
100GBASE-SR4	100 Gb/s	100m	MM	4	1	25.78125 Gb/s	CFP, CFP2, CFP4
100GBASE-SR10	100 Gb/s	100m	MM	10	1	10.3125 Gb/s	CXP, CFP, CFP2
10×10	100 Gb/s	2/10 km	SM	1	10	10.3125 Gb/s	CFP, CFP2
40GBASE-LR4	40 Gb/s	10 km	SM	1	4	10.3125 Gb/s	CFP, QSFP+
40GBASE-SR4	40 Gb/s	100 m	MM	4	1	10.3125 Gb/s	QSFP+
40GBASE-FR	40 Gb/s	2 km	SM	1	1	41.25 Gb/s	CFP

We chose the CFP2 module for our card - it enables almost all of 100 GbE media types to be supported by the card, moreover it is small enough to meet the PCI Express card dimensioning requirement. The interface also allows to be directly connected to the FPGA and the four-lane interface (8 differential pairs in total + management pins) simplifies the PCB design. Last but not least the CFP2 modules, connectors and cages are available today.

2.2.3 PCI Express

PCI Express is a system bus standard from 2004. It is based on a set of full-duplex serial lines working as differential signal pairs. The first generation had throughput 2.5 GT/s per each pair, while the 8/10 encoding yields the effective throughput of 2 Gb/s. The number of lines can be either 1, 4, 8 or 16, effectively multiplying the throughput. The most recent PCI Express standard is in its third generation. The signal speed is increased to 8 GT/s and in conjunction with more effective 128/130 link encoding the bus achieves throughput of almost 8 Gb/s per line. Therefore the bus scales theoretically up to 128 Gb/s when using 16 lines. That is the only combination that satisfies our requirements of transmitting whole 100 Gb/s traffic to the host RAM.

Virtex-7 FPGAs utilize up to three independent PCI Express endpoint blocks, each using up to 8 lines. To achieve full 100 Gb/s throughput, the card uses an external chip (PCI Express switch) to combine two x8 interfaces to one x16.

2.2.4 Memories

Our initial intent was to use an advanced memory with serial interfaces. These modern devices allow to simplify the PCB design by using significantly less wires. Some of them also offer an integrated ALU, which can perform read-modify-write

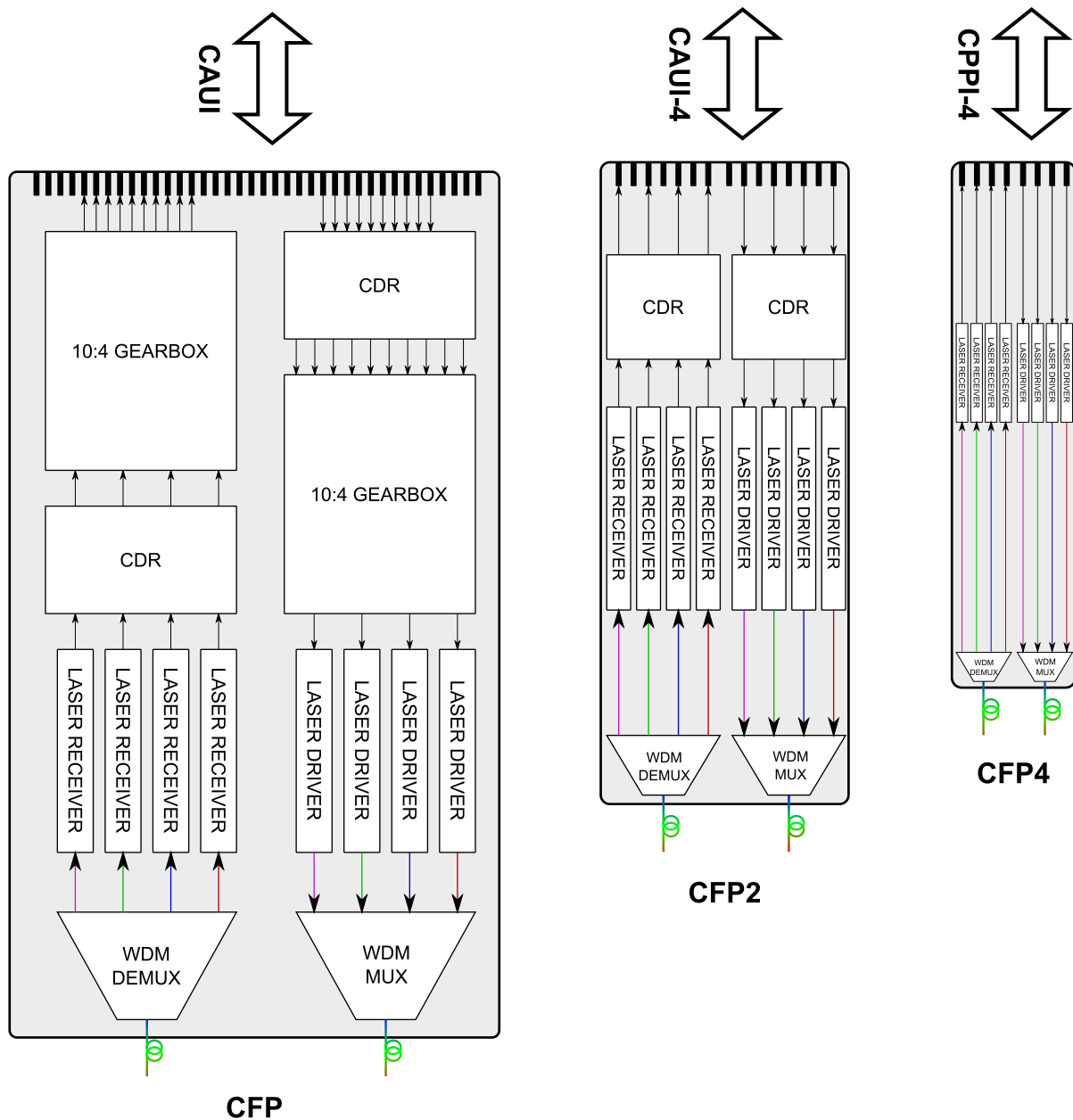


Figure 2. Comparison of 100 GbE CFP modules

operations atomically within the memory. This feature could be very useful in our projected SDM firmware.

However, our study shows that these memories are not mature enough yet to be used in our cards. The greatest issue is the complexity of communication. The memory controller has to implement three layers of communication to ensure reliable transfer of data from and to the memory. Integration of such complex system into our card would require significant time, FPGA resources and would impose additional risks to our projects. That's why we decided to use some proven technology.

We found QDR-III+ to be a viable alternative. Their communication protocol is very simple and similar to what we have done before. QDRs have two independent ports, one for reads and the other for writes. Each port can transfer data at

the frequency up to 700 MHz DDR. Combined with the data width of 36 bits, the throughput of single QDR-III+ module exceeds 50 Gb/s. To achieve very high throughput, the card utilizes three independent QDR memories. The storage capacity of the single QDR chip is 72 Mb, giving the 216 Mb in total. Due to their low latency and truly random access, these memories will be used to store state data, counters, tables, tree structures etc.

Another memory type on the board is DDR3 SDRAM. Using 800 MHz clock at 8 bits of data width, single module of this memory supports up to 12.8 Gb/s of bandwidth. Using 8 of these chips yields the throughput over 100 Gb/s. Due to higher latency, large capacity (up to 1 Gb per chip) and not completely random access pattern, these memories will be used mainly as large packet buffers.

3 FPGA firmware design

Similar to previous generations, we have designed the NetCOPE development framework for this card. The framework serves us to abstract from card-specific hardware features and allows us to focus on the application functionality. Basic building blocks are:

- Networking module containing 40/100 Gb Ethernet interfaces (MAC, PCS, PMA) for receiving and transmitting the network data
- PCI Express interface
- DMA transfer engine for very fast data transfers between the card and host memory
- Memory controllers
- Supporting modules - firmware and card identification, temperature and voltage monitors, configuration memory access, precise timestamp generation and others

3.1 100G Ethernet PMA/PCS

The Ethernet Physical Coding Sublayer (PCS) is responsible for link coding and decoding, scrambling, lane distribution and alignment. To support various physical link widths, the PCS internally uses 4 (40 GbE) or 20 (100 GbE) lane architecture. The transmitter periodically inserts an alignment marker block to all lanes, so that the receiver can detect and compensate lane to lane skew and ordering. The Physical Medium Attachment (PMA) performs bit level multiplexing and demultiplexing of PCS and physical lanes, data serialization/deserialization, clock data recovery and provides XLAUI/CAUI interface for chip to optical module communication.

GTZ transceivers included with the Virtex7-HT FPGAs greatly simplify the 100 GbE physical layers implementation and save FPGA logic resources. The GTZs comprise the whole PMA layer including the CAUI-4 interface - analog processing of the serial signal, data serialization and deserialization and 5:1 gearbox for multiplexing and demultiplexing the 5 PCS lanes at 5 Gb/s into one physical lane at 25 Gb/s. Moreover it includes the PCS gearbox (barrel shifter) for block synchronization of individual PCS lanes. Such implementation saves about 20% of logic resources in comparison with the implementation presented in [1]. Moreover, the

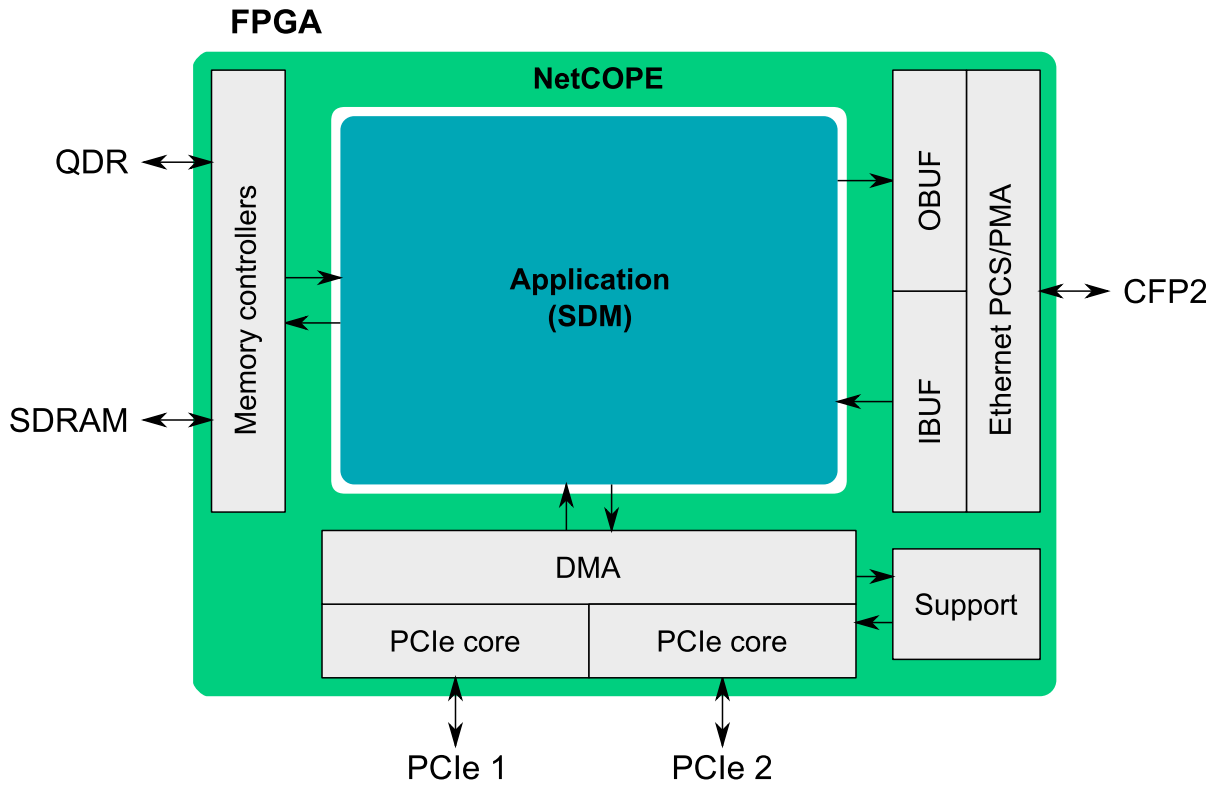


Figure 3. NetCOPE architecture on the 100 GbE card

software supplied by the FPGA vendor is able to generate the CAUI-4 configuration for GTZ transceivers automatically.

The remaining modules are implemented in logic resources inside the FPGA. The design is divided into two main blocks - the transmit path and the receive path. The transmit path block realizes the complete 100GBASE-R transmit operation as defined in 802.3ba [3], Clause 82. The path consist of a 64B/66B GBASE-R encoder, synchronization FIFO, data scrambler and alignment marker inserter.

64B/66B Encoder

performs the 100GBASE-R transmission encoding, described in detail in IEEE 802.3ba section 82.2.3.3.

FIFO

The asynchronous FIFO does the clock rate compensation between the MAC and the PMA interface. It also compensates for the data rate difference caused by the alignment marker insertion by deleting the idle control characters.

Scrambler

The payload of each 66-bit block is processed by a self-synchronizing scrambler. A parallel form with data width of 512-bit is implemented, which leads to a 15-level cascade of XOR gates and 58 D flip-flops.

Alignment Marker Inserter

periodically inserts an alignment marker to all PCS lanes, including the Bit Interleaved Parity (BIP) field - the result of a parity computation over all previous bits of a given lane, from the previous marker.

The receive path realizes the 100GBASE-R receive operation. The main building blocks are:

Lane Alignment and Reorder

is responsible for deskewing and reordering of the incoming data lanes. It looks for alignment marker blocks on all lanes. After the marker lock is obtained, it deskews the lanes to be mutually aligned and restores the lane order. The module also checks the BIP field providing the measure of bit-error ratio of a given lane. The design has been revised and optimized: the former version was based on an asynchronous approach and the implementation was therefore quite complex. The optimized version is completely synchronous and its implementation saves resources through the complexity reduction.

Descrambler

realizes the reverse operation to the scrambler in the receive path.

FIFO

The asynchronous FIFO performs the clock rate compensation between the PMA and MAC clock domains. It also compensates for data rate difference caused by the alignment marker deletion by inserting idle control characters.

Decoder

realizes the reverse operation to the encoder, i.e. converts the 100GBASE-R transmission encoding to CGMII.

The CGMII interface with a width of 512-bit and a clock frequency of 195.3125 MHz is used for the MAC layer connection. Total resources utilized by the implementation in the Virtex7 FPGA are 22508 flip-flops, 41663 LUTs and 16 BlockRAMs.

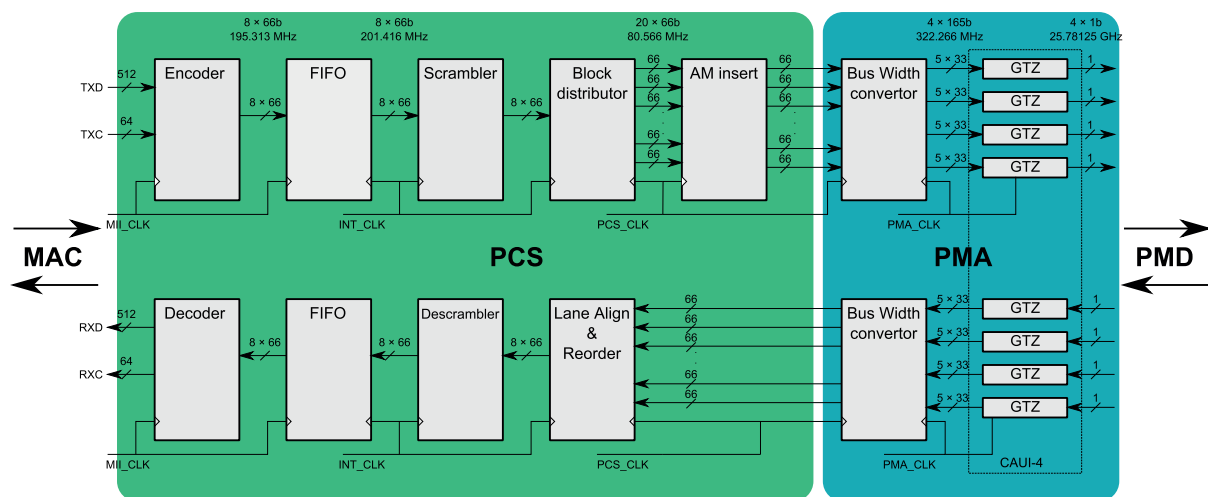


Figure 4. 100 GbE Ethernet PCS/PMA implementation

The design is supplemented by a management unit to allow the user to read and write the state and control registers according to the 803.3ba specification. The registers are holding line status, block synchronization status of individual lanes, alignment status, block- parity- and decode error counters and others.

All variants of the Ethernet PCS/PMA have been successfully verified in simulations. Moreover, the 40 GbE version of the core has been already proven in hardware on the development kit with 10 Gb GTH transceivers, whereas the operation and interoperability was checked using the Spirent TestCenter network tester. The hardware tests of the 100 GbE version are in progress and they should be done in next few months.

3.2 Ethernet MAC: IBUF and OBUF

A receive path of the Ethernet Media Access Control (MAC) sublayer of the Data Link layer is implemented within the NetCOPE framework using an Input Buffer (IBUF) module. A block schematic of IBUF's internal structure is shown in Figure 5. Functions of displayed submodules are as follows.

Decoder

The main aim of this submodule is to transform data received on the CGMII interface to the format utilized on a FrameLink Unaligned output of the IBUF. This operation consists of an input format correctness check, which is followed by removing the input frame's preamble and a start of frame delimiter (SFD).

Checker

Firstly, the Checker submodule counts the length of the input frame and computes its cyclic redundancy check (CRC) value. Subsequently, several checks are performed on the input frame

- destination MAC address check
- minimum frame length check
- maximum frame length check
- CRC check
- sampling

Moreover, the destination MAC address check can be performed in several modes

1. only MAC addresses stored in CAM memory are valid
2. mode 1 + broadcast MAC addresses are valid
3. mode 2 + multicast MAC addresses are valid
4. all MAC addresses are valid (promiscuous mode)

When all checks are done, the frame's CRC value can be optionally removed.

CAM

This submodule of the IBUF consists of content-addressable memory (CAM) for lookup of MAC addresses.

FIFO

The FIFO submodule contains a buffer for input frames with asynchronous read and write interfaces. The buffer implements store-and-forward functionality which is necessary for implementation of input frames discarding. Discarding is driven by the result of checks performed within the Checker submodule.

The result of checks are maskable (except the result of sampling). The FIFO submodule is also a place where control data for an input frame are added to the data path. These control data are acquired from a Packet Control Data Generator (PACODAG) and can optionally contain a timestamp with nanosecond resolution.

Statistics

The IBUF implements RFC 2819 [5] compliant set of counters. These counters can be found in the Statistics submodule.

MI32 Interface

This submodule connects software-accessible status and control registers and CAM memory to an MI32 bus of the NetCOPE framework. The connection utilizes FIFO buffer with asynchronous read and write interfaces.

The data path through the IBUF consists of 512-bit wide bus operating at 195.3125 MHz on the input and at user-defined frequency on the output. Third clock domain within the IBUF can be found within the MI32 Interface submodule.

The IBUF component utilizes 4780 flip-flops 8298 LUTs and 19 BlockRAMs. Its functionality has been verified in simulation and in functional verification as well.

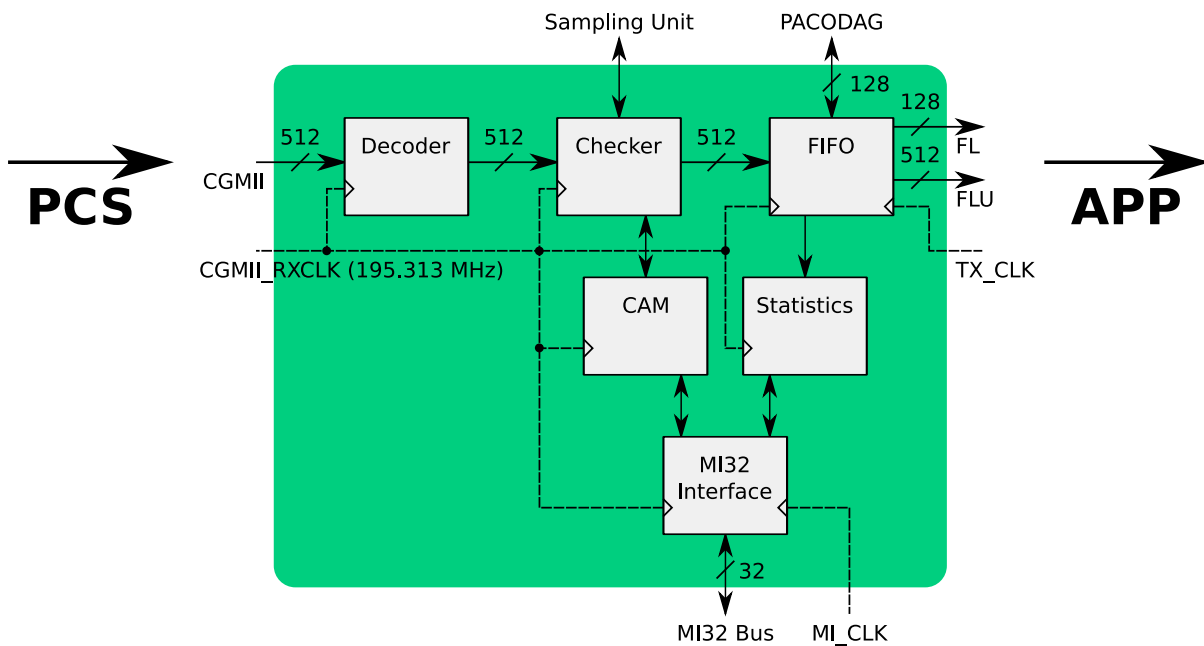


Figure 5. 100 GbE IBUF Internal Structure

The transmit path of the MAC sublayer is implemented using an Output Buffer (OBUF) module. A block schematic of OBUF's internal structure is shown in Figure 6. Functions of the OBUF's submodules are as follows.

FIFO

The FIFO submodule contains buffer for output frames which implements store-and-forward functionality, thus ensuring a continuous data/idle stream

on an output CGMII interface of the OBUF. Similarly to the FIFO submodule of the IBUF, the buffer has asynchronous read and write interfaces. This submodule also contains counters of all transmitted frames and octets.

CRC Insert

The only aim of this module is to compute a CRC value for current output frame and to append this value to the end of the frame.

Encoder

This submodule of the OBUF is responsible for transformation of output frames to the format specified for the output CGMII interface. A preamble and a SFD are prepended to the output frame and, if necessary, the encoder inserts an inter-frame gap (IFG), determined by a deficit idle count (DIC) mechanism, between two consecutive output frames.

MI32 interface

Functionality of this submodule is almost the same as of the same submodule of the IBUF. The only exception is that this submodule does not connect CAM memory to the MI32 bus because there is no CAM memory within the OBUF.

Similarly to the IBUF, the data path of the OBUF consists of 512-bit wide bus. However, in the OBUF the input of the data path is operating at user-defined frequency and its output is operating at 195.3125 MHz. Third clock domain within the OBUF is again in the MI32 interface submodule.

The OBUF component utilizes 5429 flip-flops 10806 LUTs and 23 BlockRAMs and its functionality has been verified in simulation and in functional verification as well.

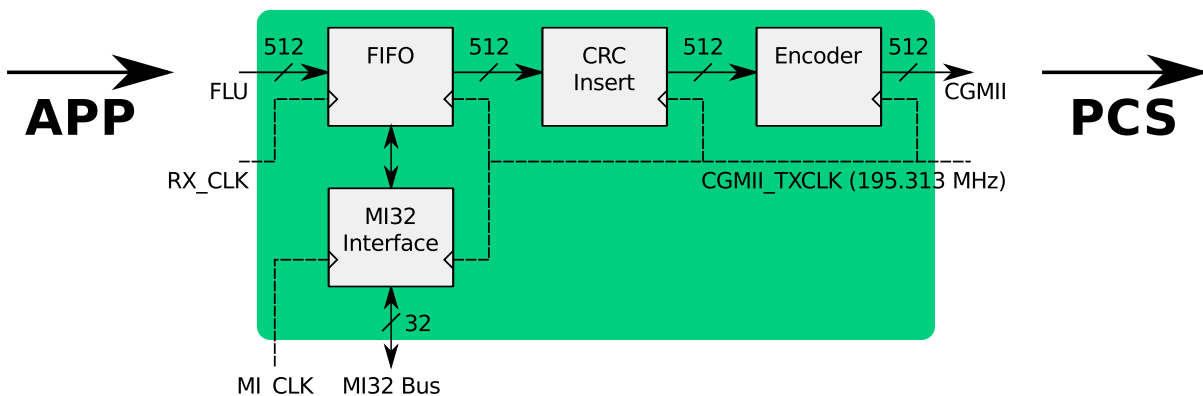


Figure 6. 100 GbE OBUF Internal Structure

3.3 DMA

DMA engine provides bidirectional transmission of data frames (network packets or any other data) between software applications and firmware blocks on the FPGA chip (typically connected to network interfaces). The whole DMA engine is in fact a highly effective parallel implementation of multi-threaded and synchronized data handover. The engine is composed of several software and hardware modules and is designed in order to achieve the maximum data throughput using the PCI Express

Gen3. It uses the principle of two circular buffers for each channel - one is located in the RAM of host system and the second is on FPGA. The hardware initiates data transfer between the buffers in both directions.

The first part of the software is a Linux driver, which configures the DMA controllers on the card, handles interrupts and manages the ring buffers and their locks (pointers). The second part is a software library named SZE that surrounds a circular buffer. This allows the software applications to operate at the level of individual frames of data.

The transfer itself is performed by the hardware part of the system. Hardware DMA module (Figure 7) is composed primarily of the buffers that store the data frames until they are passed to the application and the DMA controllers, that monitor the buffer status and initiate data transfers.

DMA module receives the data into several independent queues (channels). Subsequently, the controller tries to aggregate the data frames in each queue so that it can generate the longest possible transaction to PCI Express bus. This is necessary in order to achieve the desired throughput. After generating the request and data transaction, an interrupt is generated to inform the software driver that the transfer has completed.

DMA RX buffer

This buffer stores the packets from the network interface (more precisely, data frames from the HW application). First it checks whether there is enough space in the corresponding channel. Then it efficiently inserts the data into the memory and informs the New data process.

New data process

This process monitors the pointers, free space in the buffers and controls the states of DMA transfers. Accordingly to the current state it generates requests for transfers and forwards them to the Request process.

Request process

This process generates DMA transactions as requested by New data process. In the RX direction it reads from the hardware RX DMA buffer, appends the data header and forwards the result via the PCI Express bus to the RAM. In the TX direction it generates read requests to RAM and also receives completion transactions with data to be written to the TX hardware buffer.

Update and interrupt process

To allow the software to process the new data, this process regularly updates the ring buffer pointer in RAM.

Descriptor manager

Because the software ring buffer may not be allocated as a continuous block of memory, it is necessary for the Request process to know the address of each 4 KiB page buffer. The Request process then uses it in the DMA transaction header as the target of the write or source of the read transaction.

DMA balancer

As already mentioned, for the throughput of 100 Gb/s it is required to use two PCIe endpoints on the FPGA chip. So it is necessary to spread the DMA requests to load both endpoints evenly for both directions of data transmission. For this purpose the balancer module monitors the current workload of PCIe endpoints and sends packets to the first one or to the second one accordingly.

DMA TX buffer

This buffer accepts writes from the Request process. Data is sent to the output interface as soon as it is properly written to the buffer. Immediately after the dispatch of the complete data frame the buffer informs the New data process of the freed space.

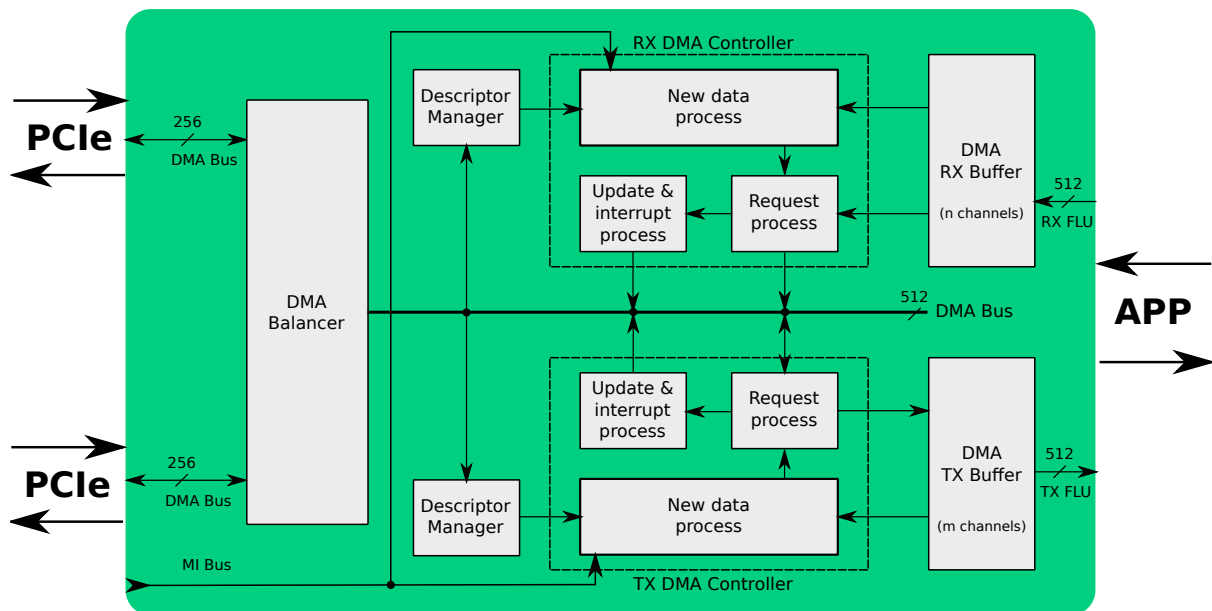


Figure 7. DMA module block schematic

3.4 Memory controllers and others

NetCOPE aims at providing simple, user-friendly interfaces to all card's memories. The QDR-III+ memory controller was delivered by the card vendor together with the card, while the DDR3 controller can be generated from the CoreGen tool provided by Xilinx. Other modules offered by NetCOPE include:

- Precise timestamp generator. This generator allows GPS synchronization to achieve high precision.
- Firmware identification module. In addition to holding some basic firmware identifiers, this module enables the access to some FPGA-specific features, such as temperature monitoring.
- Set of FrameLink building blocks. This comprises a rich set of various modules that allow the user to create custom pipelines that convey packets through the firmware. The FrameLink protocol is used by the modules delivering packets from and to the user application - Ethernet MAC and DMA module.

4 Conclusion

This technical report described our requirements and results in creating a high-speed programmable hardware accelerator of network monitoring and other tasks. While the complete firmware is still a work in progress, we can draw some preliminary conclusions. The hardware tests performed so far included complete 40 Gb/s Ethernet interface tests and PCI Express gen3 x8 interface tests. Both tests demonstrated that our firmware modules are standards-compliant and achieve the expected speeds.

Our future work on the card includes further testing of performance and conformance. Concurrently we work on the complete firmware and software integration, which will result in high-speed SDM firmware for 100 Gb/s network monitoring.

References

- [1] Štěpán F., Novotný J., Lhotka L. *Study of 40/100GE card implementation* CESNET technical report 22/2010 Available online¹
- [2] IEEE 802.3-2008. 26 December 2008 [cit. 2010-11-01]. ISBN 973-07381-5796-2. Available online²
- [3] IEEE 802.3ba-2010. 22 June 2010 [cit. 2010-11-01]. ISBN 978-0-7381-6322-2.
- [4] Viktor Puš, Lukáš Kekely *Software Defined Monitoring: A new approach to Network Traffic Monitoring* Available online³
- [5] S. Waldbusser *Remote Network Monitoring Management Information Base*. RFC 2819, May 2000 [cit. 2013-12-16]. Available online⁴.

¹ <http://archiv.cesnet.cz/doc/techzpravy/2010/100ge-study/>

² http://standards.ieee.org/getieee802/download/802.3-2008_section4.pdf

³ <https://tnc2013.terena.org/core/poster/5>

⁴ <http://www.ietf.org/rfc/rfc2819.txt>