An Implementation of Optical Transponder and Media Converter Unit for Speeds up to 2.5 Gbps*

Nikola Jovalekić, Predrag Mićović, and Jelena Popović-Božović, *Member, IEEE*

***Abstract* — The paper presents the implementation of protocol agnostic optical transponder and media converter unit for speeds up to 2.5 Gbps. Detailed depiction of the implementation is given, including hardware design issues, as well as the signal integrity solutions. In addition, the design implemented in a field-programmable gate array (FPGA) chip is discussed. On the other hand, embedded software implementation is explained, as well as the interface to the network management software that executes on a PC. Finally, strategy for the verification of the unit functionalities at the end of the development stage is given, and results are presented.**

***Keywords* — Media converter, optical transponder, optical transport systems, signal integrity.**

I. Introduction

THE backbone of modern communications is based on optical transport systems [1], [2]. Optical transport systems offer many advantages compared with any other transport system where a transmission medium is not a fiber. Some of the advantages when a fiber is deployed as a transmission medium are:

- Very high bandwidth
- Smaller dimensions and weight (in comparison with e.g. UTP cables paralleled to provide the same throughput)
- Galvanic isolation between transmitter and receiver
- Immunity to external electromagnetic aggressors
- Longer transmission paths
- Scalability

Optical transmission systems can provide the transmission capacity of a few terabytes per second using wavelength division multiplexing. This means theoretically that all voice conversations around the world at some point can be simultaneously transmitted over a single fiber [3].

Galvanic isolation between a transmitter and receiver is a very important feature because it has resolved the grounding problem, especially in local area networks (LAN). Furthermore, because there is no electrical connection between a transmitter and receiver, electromagnetic immunity has been drastically improved. That enabled deployment in very hostile electromagnetic environments.

Due to increased requirements for throughput, many network topologies and data transmission standards have been developed. As a consequence, increased effort for the purpose of providing interoperability has been invested. Besides that, various devices have been developed that enable connection of different networks. Some of the developed devices are optical transponders and media converters. The first ones are used when there is a need to transpose the wavelength, e.g. from 1310nm to 1550nm, or to convert into coarse wavelength division multiplexing (CWDM) or dense wavelength division multiplexing (DWDM) wavelengths [4]. Media converters are used when there is a need to switch between transmission media, for instance, from copper to fiber and vice versa [5].

As far as optical transponders are concerned, they can provide traffic protection combining an extra fiber path and corresponding logic for error detection and management. There is also a distinction between optical transponders according to functionalities they can provide in terms of signal conditioning. Optical transponders that can regenerate and re-amplify the inbound signal, but do not provide retiming are said to be the so-called 2R optical transponders [5]. On the other hand, transponders that enable retiming as well as regeneration and re-amplification of the inbound signal are said to be transponders with 3R functionalities. Finally, there are also transponders with 3R+ functionalities. They enable, besides the mentioned functionalities, mapping the client signals based on ITU-T G.709 recommendation with integrated forward error correction (FEC). In this case, traffic protection is based on key synchronous digital hierarchy (SDH) overhead bytes, such as B1 and B2 that allow fault isolation, as well as performance monitoring capabilities. In contrast, 2R performance monitoring is based on measured parameters provided by the small form factor pluggable (SFP) module, such as received and transmitted optical power, laser bias current, etc.

As far as media converters are concerned, they are widely used to enable connections of UTP copper-based Ethernet equipment over a fiber optic link. Besides that, converters are often used to extend links over greater

***The "Ilija Stojanović" award for the best scientific paper presented at Telecommunications Forum TELFOR 2012.**

The authors would like to thank the Ministry of Education, Science and Technological Development, Republic of Serbia, for partially funding the work on the project TR32007.

Nikola Jovalekić is with the Iritel A.D., Batajnički put 23, 11000 Belgrade, Serbia (phone: 381-60-0123-762; e-mail: n_jovalekic@iritel.com).

Predrag Mićović is with the Iritel A.D., Batajnički put 23, 11000 Belgrade, Serbia (phone: 381-11-3073-455, e-mail: micovic@iritel.com).

Jelena Popović-Božović is with the School of Electrical Engineering, University of Belgrade, Bul. kralja Aleksandra 73, 11120 Belgrade, Serbia (phone: 381-11-3218-310, email: jelena@etf.rs).

distances due to poor characteristics of copper-based transmission medium.

The main reason for developing the optical transponder and media converter unit was to make the optical transport platform ODS2G5 a fully rounded solution for optical transmission up to 2.5Gbps[6]. The unit provides media conversion – from copper to fiber and vice versa for Fast and Gigabit Ethernet. It can also be used as an optical protection unit in mesh networks due to 1+1 optical path protection. Moreover, it enables connection to CWDM/DWDM networks through the traffic which is not mapped in ODS2G5. Finally, the unit enables 2R optical transponder functionalities and is protocol agnostic.

II. Hardware Implementation

Hardware implementation required addressing various problems. The first one was to specify functionalities the board should have. The final decision was to implement 2R functionalities because it was not planned for the device to be realized as a standalone unit. The block diagram of the realized unit connected with the main board in the ODS2G5 system via backplane is given in Fig. 1.

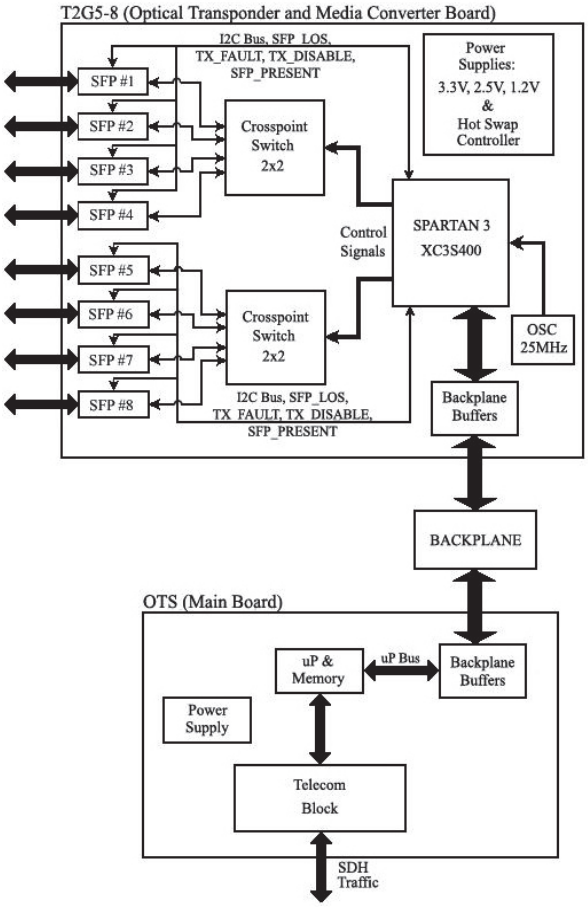

Fig. 1. Block diagram of the unit connected with the main board via backplane within ODS2G5 system.

The board is designed to be truly hot swappable and hot pluggable, which means that it can be de-attached from or attached to the system without interrupting the system, and without manually notifying the system that the board is attached or de-attached. That is implemented using a hot swap controller and adequate realization in software.

Chipset on the board requires three different power supplies: 1.2VDC, 2.5VDC, and 3.3VDC. Power supplies are realized by switching and low dropout voltage regulators. A buck DC-DC converter lowers the backplane voltage (-48VDC) to 3.3VDC which is delivered to SFP modules, crosspoint switches, and FPGA chip. The buck converter is realized as a hybrid module with necessary filters to reduce conductive emissions from the module. Additional 2.5VDC, and 1.2VDC power supplies, required by FPGA chip, are provided with a dual linear low drop out regulator. The decision to choose linear regulators in this particular case is based on the practice that whenever it is possible to use them in FPGA power supplies, they should be used because of the low noise generation [7].

The board clock is realized by 25MHz crystal. Because FPGA chips heavily depend on low jitter clock sources, the crystal is a very suitable solution due to low phase noise and good frequency accuracy. This assertion is valid for crystals with frequencies approximately up to 50MHz.

Firmware for the FPGA chip is written in VHDL using Xilinx Ise WebPACK 13.4 environment. It is divided into three major blocks: the block that provides communication with the processor on the main board via a parallel bus, the I^2C block with a multiplexer responsible for reading SFP module parameters, and the block which implements traffic protection algorithms and management of crosspoint switches.

Data transmission between the realized unit and the main board is done via a parallel bus. In FPGA chip are realized registers that are memory mapped in the address space of the processor on the main board. Processor on the main board reads all parameters obtained by FPGA chip via a parallel bus. On the other hand, SFP module parameters are read and set by FPGA chip via I^2C bus.

SFP module signals such as loss of signal (LOS), presence of SFP module in the cage, or laser fault indication are monitored by FPGA chip, and are used as parameters in implementing traffic protection algorithms. Traffic protection algorithms are realized on the physical layer, which means that the main criterion for the choice of working path is LOS signal set by a particular SFP module.

The user can set working and protection transmission paths, but settings depend primarily on the LOS signal status on a particular SFP module. This means that a particular SFP port can be set only if the LOS signal on that port was not active at least one integration period. All LOS signals are monitored by the algorithm implemented in FPGA chip, and adequate status registers are provided so the processor on the main board can read a current state of the SFP modules.

Integration period is also set by the user and can take values in the range 10ms-150ms in 10ms steps.

The user can also set one of the protection algorithms. The difference between commands and protection algorithms is in the level of user's involvement in selecting working and protection paths when errors occur. In other words, if the user sets the integration time, working and protection transmission path, and the type of protection algorithm, operation of the unit performs

automatically. When a LOS signal occurs on a working path, switching to protection path occurs immediately and automatically only if there was no LOS signal for at least one period of integration on a protection path. Fig. 2 depicts all possible traffic routing combinations.

A protection algorithm can be revertive or non-revertive. For example, if the user selects the revertive protection algorithm, the working path will be active again when it becomes loss-free for at least one integration period. On the other hand, if the user selects the non-revertive protection algorithm, the return to working path will only happen if the protection path reports loss of signal and the working path is loss-free for at least one integration period at that moment. Finally, there is an option to force setting the working path in spite of the presence of the loss of signal.

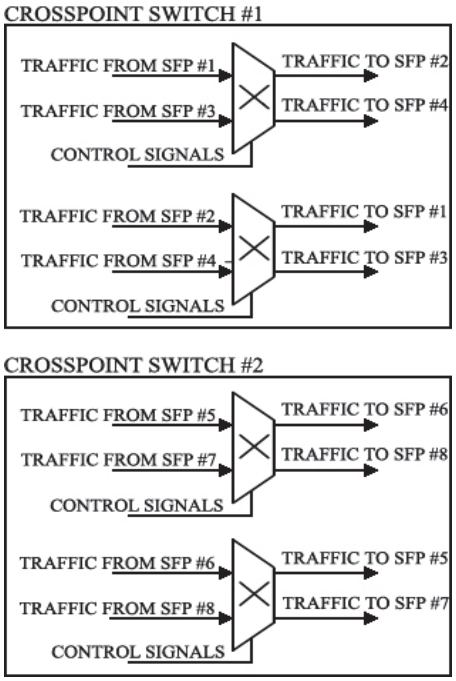

Fig. 2. All possible traffic routing combinations.

Implementation of these functionalities is based on several processes implemented in FPGA chip:

- process that makes a sample rate of 2.5MHz for monitoring LOS signal on SFP modules
- process that buffers status signals from SFP modules including LOS signal
- process that implements commands for selecting working and protection paths
- process that enables integration time settings
- process that enables rising the interrupt when LOS signal occurs, and its deletion by software executed on the main board processor
- process that monitors the LOS signal on each SFP module
- process that implements lower layer of the state machine which implements protection algorithms

Spartan 3 XC3S400 is used in the implementation, and the device utilization summary is given in Table 1. The decision to choose this FPGA chip, in spite of low utilization, is based on the adequate number of I/O pins for this design.

Two 2x2 crosspoint switches provide traffic routing from one port to another based on control signals from FPGA chip. Each switch provides traffic routing for a group of 4 SFP interfaces. Control signals from FPGA chip select SFP modules based on protection algorithms.

There are also three light emitting diodes (LED) placed on the front panel of the board, which are used as indicators. The green LED denotes that power supply is present; the yellow LED is reserved for low-priority alarms, while the red LED indicates high-order alarms. They are used for a quick, visual inspection of the unit without involving network management software.

TABLE 1: FPGA UTILIZATION SUMMARY.

Logic Utilization	*Used*	*Available*	*Utilization [%]*
Number of Slice Flip Flops	1,298	7,168	18
Number of 4 input LUTs	1,642	7,168	22
Number of occupied Slices	1,406	3,584	39
Number of Slices containing only related logic	1,406	1,406	100
Number of Slices containing unrelated logic	0	1,406	0
Total Number of 4 input LUTs	2,160	7,168	30
Number used as logic	1,634		
Number used as a route-thru	518		
Number used as Shift registers	8		
Number of bounded IOBs	149	264	56
IOB Flip Flops	80		
Number of RAMB16s	1	16	6
Number of BUGGMUXs	3	8	37
Average Fanout of Non-Clock Nets	3.09		

Finally, design of the stack-up of the printed circuit board was specific, as well as the trace geometry and routing strategy, due to the presence of high-speed signals with a very short rise time. Stack-up was realized as a ten-layer board, with six routing layers and four plane layers. High speed links are routed in two inner routing layers. Striplines are chosen because they offer lower far-end crosstalk due to homogeneous dielectric around them [8], [9]. In contrast, that routing strategy brings the greater via stubs, but in this particular case they did not degrade the signal integrity enough to cause malfunctioning of the board. A controlled, differential impedance of 100Ω was implemented to minimize reflections. Discontinuities in return planes below differential pairs were avoided to preserve the signal integrity.

The physical appearance of the board is given in Fig. 3.

Fig. 3. The physical appearance of the board.

III. Software Implementation

Software implementation encompassed writing embedded software that executes on the main board processor including the software for board functional verification during the development stage.

Embedded software is written in C and cross-compiled to be executed on the main board processor where Linux operating system is installed with kernel 2.6. One of the processes is responsible for communication with PC via Fast Ethernet 100BASE-TX or RS-232 port. When the message from PC is received, it is decoded and sent further to the process which is responsible for the requested action (i.e. configuring one of the boards within the ODS2G5 optical transmission system). On the other hand, messages are also sent from the device to PC when, for instance, alarms are read on user request.

In the same manner, one of the processes manages the optical transponder unit. Messages are received from PC via Fast Ethernet or RS-232 port and requested actions by the user are performed by writing to or reading from the registers implemented in FPGA chip on the board. The values that are read are SFP module parameters such as transmit or receive power, laser diode polarization current, temperature, SFP presence on the board and SFP loss of signal. They can be sent to PC periodically or on user request. Network management software needs these parameters, among others, for the purpose of monitoring and managing the whole network of devices. Communication between the optical transmission platform and PC is done via predefined message formats.

A user can set the desired mode of operation or read alarms from the optical transponder unit via a single form under the application that executes on PC.

When the board is inserted, it is registered by the process that executes on the main board processor, and a message is sent to PC. After that, the form for managing the board will automatically appear.

The common way of configuring the optical transponder board is to choose the working and protection path, for example, inbound signals on SFP interfaces number 2 and 4. According to Fig. 2, the outgoing traffic can be directed to interfaces 1 and 3. Next, let us assume that SFP#2 and SFP#4 are specified for the wavelength of 1310nm, whereas SFP#1 and SFP#3 have to work on the wavelength of 1550nm because an interconnection between two operators has to be established. It is obvious that one operator just has to plug SFP interfaces with required wavelengths and set a desired cross-connection. Additionally, the operator can set traffic protection (including integration time), which adds additional reliability to transmission.

On the other hand, if Gigabit Ethernet is considered, media converter functionality is also as easy to implement as to switch the type of SFP module between optical and electrical. In this case, traffic protection can also be applied, as in the previous example.

IV. Verification Strategy and Results

The verification included measuring the jitter, testing the logic functionalities, and measuring the switching time characteristics between working and protection path.

All measurements were performed with *Anritsu MP1590B* network performance tester including the jitter measurements. According to the ITU-T G.783 recommendation, the duration of the unit interval for jitter measurement at a speed of 2.5Gbps (STM-16) was 0.4ns, whereas for 622Mbps (STM-4) it was 1.61ns. Summarized jitter measurement results are given in Table 2.

As can be seen in Table 2, all the results are at least twice better than the values required by the ITU-T G.783 recommendation.

Traffic generation for testing the logic functionalities was also done by *Anritsu MP1590B*. The working and protection paths were simulated with an optical splitter. The board was initialized in every operating mode and a LOS signal was artificially inserted by simply plugging out a fiber from SFP module.

Table 2: Jitter measurement results.

PORT No.	JITTER ON STM-16 [U.I.p-p]		JITTER ON STM-4 [U.I.p-p]	
	HP1+LP (5kHz-20MHz)	HP2+LP (1MHz-20MHz)	HP1+LP (1kHz-5MHz)	HP2+LP (250kHz-5MHz)
OP#1	0.066	0.048	0.026	0.012
OP#2	0.038	0.026	0.033	0.017
OP#3	0.031	0.024	0.035	0.017
OP#4	0.048	0.033	0.02	0.009
OP#5	0.062	0.046	0.03	0.009
OP#6	0.039	0.027	0.031	0.013
OP#7	0.064	0.051	0.032	0.011
OP#8	0.043	0.032	0.034	0.012

Timing constraints include the switching time between working and protection path in less than 50ms, according to the ITU-T G.841 recommendation. This was validated by a software test. Namely, the test was carried out by reading the values from status registers in FPGA chip, which denote whether a particular SFP interface is active or not. When the loss of signal occurs on a particular SFP module, also an interrupt occurs, which must be reset by software. After resetting the interrupt, a poling loop is started periodically checking the activation status register of the protection SFP module. After every cycle, there is a fixed time delay to make the switching time directly proportional to it. Hence, the switching time is a

multiplication of a time delay and number of passes through the loop. The result of this measurement was switching time of approximately 2ms (when all latencies in software were included) which is 25 times better than the ITU-T G.841 recommendation requires. Because of the advantageous result in unfavorable test conditions, further, more precise tests were not developed and carried out.

V. Conclusion

The described implementation offers a very short switching time, as well as good jitter characteristics. The short switching time is achieved due to FPGA chip deployment and therefore truly parallel monitoring and management of the SFPs parameters. On the other hand, well structured protection algorithm implementation through a state machine has also contributed to a short switching time. Good jitter characteristics enable the unit to be used as the source of the synchronization in SDH/SONET transport networks. In addition, the unit offers traffic protection on the physical layer, based on revertive and non-revertive algorithms.

Finally, the implemented unit has demonstrated very good deployment performances.

References

[1] H. J. R. Dutton, *Understanding Optical Communications*. IBM, 1998.
[2] M. Azzadeh, *Fiber Optics Engineering*. New York: Springer, 2009.
[3] S. Kartalopoulos, *Next Generation Intelligent Optical Networks*. New York: Springer, 2008.
[4] R. Ramaswami, K. N. Sivarajan, G. H. Saski, *Optical Networks*. London: Elsevier, 2010.
[5] R. Elsenpeter, T. J. Velte, *Optical Networking*. Osborne: McGraw-Hill, 2002.
[6] "ODS2G5 Technical Description", IRITEL, 2012.
[7] V. A. Pedroni, *Circuit Design With VHDL*. London: MIT press, 2004.
[8] E. Bogatin, Signal *and Power Integrity*. San Francisco: Prentice Hall, 2010.
[9] S. H. Hall, H. L. Heck, *Advanced Signal Integrity for High-Speed Digital Designs*. New Jersey: Wiley, 2009.