UTC based Controller for Scalable Time Driven Switching

Deepak Agrawal*, Michele Corrà**, Viet-Thang Nguyen*, Yoram Ofek*
*Department of Information and Communication Technology
University of Trento, Via Sommarive 14, I-38050 Povo - Trento - Italy
Email: {deepak, nguyen,ofek}@dit.unitn.it
** TRETEC S.r.l., Via Solteri 38, 38100 - Trento – Italy
michele.corra@3tec.it

Abstract— **As data traffic on the Internet continues to grow exponentially, there is a real need to solve the switch bottleneck by developing ultra-scalable switching fabrics and their control. Although in recent years there have been a lot of efforts to solve the switching fabric scalability problem, in the optical domain, the proposed solutions have been (very) expensive, (very) complex and (much) larger than electronic switching fabric alternatives. In order to increase scalability a coordinated universal time (UTC) based time-driven switching (TDS) is proposed. This novel low-complexity switching architecture capitalizes on the ubiquitous of UTC from GPS and Galileo. This work focuses on UTC-based FPGA (field programmable gate-array) controller that was implemented for controlling the TDS switch prototype at the University of Trento, Italy. The switch controller facilitate dynamic configuration of ultra scalable switch. The prototype is implemented using off-the-shelf components. TDS architecture is especially suitable to support high capacity streaming media applications over the Internet.**

Keywords – switching; sub-lambda switching; time-driven switching; FPGA; streaming media; quality of service

I. INTRODUCTION

The aggregate bandwidth of a single optical fiber presents a major challenge to packet switching speed. The overarching emphasis on switching speed results in unconventional hardware architectures that create new challenges for the software engineer as well. So, if the internet traffic is doubling, say, every 18 months there is a real switching bottleneck on the horizon. Note that Cisco's top-of-the-line router with a novel network processor design, CRS-1, has 640Gb/s per chassis [the announcement of 92Tb/s should be divided by 2 (for counting input and output separately) and then by 72 chassis's], which represents a factor of 2 improvement after 5 yrs of development with 500 million dollars of investment.

The newest generation of Field programmable gate arrays (FPGA) have hit performance and cost goals which allow a much wider spectrum of applications support. Now a day, flexibility and large number of functions are needed in each protocol layer for providing quality of service in data networks, but current networks are far from rich in terms of flexibility. To help remedy this situation, we have designed and implemented an FPGA based switch controller for ultra scalable time-driven switching (TDS) architecture.

The function of the controller is to dynamically configure the TDS fabric. In the TDS architecture there is no need for packet header processing therefore the FPGA based switch controller is a simple and low cost alternative to network processor. The TDS architecture guarantees deterministic QoS for streaming media over the Internet and is scalable to speed of 10-100 Terabit per second (per chassis). In order to achieve such performance UTC based pipeline forwarding of IP packets (or Ethernet frames) is used. Pipeline forwarding is a method known to provide optimal performance independent of specific implementation.

The authors of [1] have reported a telecommunication-oriented FPGA controller, supplement to ASIC/MPU, for ATM adapters. Authors of [2] have suggested the integration of FPGA-based controller with optical transponder for static DWDM. A FPGA controller for contention resolution in an optical packet router has been implemented in [7]. A high speed serial transceiver running at sub-nominal rate for data recovery is reported in [10].

In this paper we present the FPGA based controller for dynamic configuration of ultra scalable switch fabric. Details of other parts of prototype are out of scope of the paper. This paper is organized as follows: section II introduces the basic principles on which the operation of prototype is based, section III describes scalable TDS switching architecture with emphasis on FPGA based switch controller, section IV presents experimental demonstration of the switch prototype, section V describes the presented work and discusses some open issues.

II. TIME DRIVEN SWITCHING PRINCIPLES

This section briefly describes the basic principles on which the operation of proposed prototype depends.

A. Time Driven Switching

TDS [4]-[7] was proposed as an effort to realize highly scalable dynamic IP switching. TDS (in the optical domain TDS is referred as Fractional Lambda Switching) enables the switching of IP packets in time frames (TFs) without processing packet header. In TDS, a concept of common time reference using UTC (coordinated universal time) is introduced. A UTC second is partitioned into a predefined number of time cycles (TC) and each time cycle is further

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

partitioned into a predefined number of TFs. TFs can be viewed as virtual containers of multiple IP packets that are switched at every TDS switch. TDS switch operation is based on and coordinated by the UTC signal. In TDS all packets in the same TF are switched in the same way. Consequently, header processing is not required, which results in low complexity (hence high scalability) and enables all optical implementation. As shown in Figure 1, a group of K TFs forms a time-cycle (TC); L contiguous time-cycles are grouped into a super cycle (SC), which is equal to one UTC second, for example, in Figure 1, K =1000 and L =80. To enable TDS, TFs are aligned at the inputs of every TDS switch before being switched. After alignment, the delay between any pair of switches is an integer multiple of TFs. The central element of TDS is pipeline forwarding using common time reference – e.g., UTC. In TDS, a fractional lambda pipe (FλP) p is a predefined schedule for switching and forwarding TFs along a path of subsequent TDS switches. The FλP capacity is determined by the number of TFs allocated in every TC for the FλP. For example, for a 10Gb/s optical channel and K =1000, L =80 if one TF is allocated in every TC or super cycle the FλP capacity is 10Mb/s or 125Kb/s, respectively.

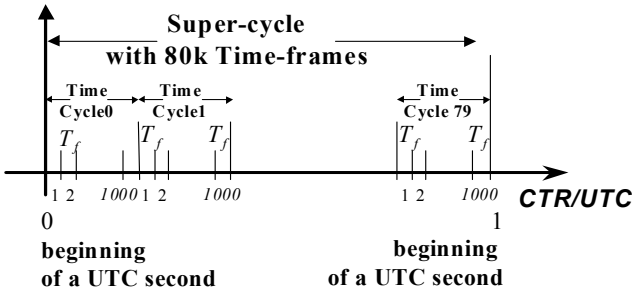

Figure 1: Division of an UTC second in TDS

B. Pipeline Forwarding

The basic pipeline forwarding operation is regulated by two simple rules: (*i*) all packets that must be sent in TF t by a node must be in its output channels' buffers at the end of TF t-1, and (*ii*) a packet p transmitted in TF t by a node n must be transmitted in TF $t+d_p$ by node n+1, where d_p is an integer constant called *forwarding delay*, and TF t and TF $t+d_p$ are also referred to as the *forwarding TF* of packet p at node n and node n+1, respectively. The value of the forwarding delay is determined at resource-reservation time and must be large enough to satisfy (*i*). In pipeline forwarding, fractional lambda pipe (FλP) p, is a predefined schedule for forwarding a pre-allocated amount of bytes during one or more TFs along a path of subsequent UTC-based switches as exemplified in Figure 2, which depicts the journey of an IP packet from node A to node D along three UTC-based switches, the forwarding delay may have different values for different nodes, due to different propagation delays on different links (e.g., T_{ab}, T_{bc}, and T_{cd}), and different packet processing times in heterogeneous nodes (e.g., T_{bb} and T_{cc}). UTC-based forwarding guarantees that reserved real-time traffic experiences: (*i*) bounded end-to-end delay, (*ii*) delay jitter lower than two TFs, and (*iii*) no congestion and resulting losses.

C. Idle Time Consideration

In order to provide deterministic no data loss an idle time between two consecutive TF is required during which input-output switching configuration is changed. Idle time is the time gap between two consecutive TFs during which no data is forwarded.

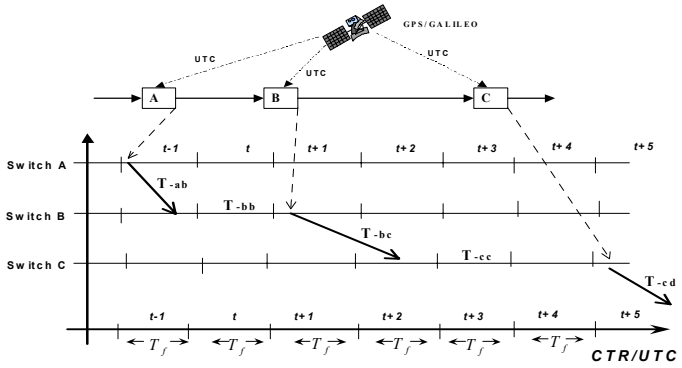

Figure 2: Pipeline Forwarding

III. Ultra Scalable Switch Architecture

The block diagram of a TDS switch is shown in Figure 3. It has three major parts: switch controller (or simply controller), switch fabric, and a GPS time receiver. The switch fabric is a set of interconnected switching chips (can be connected in various forms, e.g. matrix, single stage, multistage etc.). The switching chips are controlled by the FPGA switch controller. GPS receiver is connected to an antenna which is mounted externally facing the sky (not shown in the Figure 3).

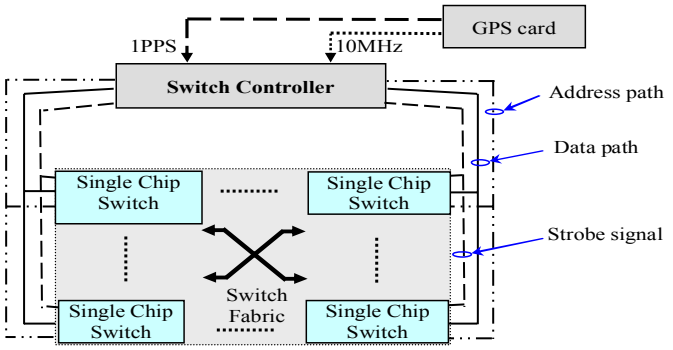

Figure 3: Functional block diagram of a 10Tb/s Time-Driven Switching

The communication between controller and switching chips can be parallel or serial. There are multiple control signals such as I/O enable, switching chip status and so on. However, for the sake of simplicity only three important classifications address, data path and strobe signals are shown. Control data (including output register addresses and their input channel selections for every TF) are stored in the memory of the controller. The data and address path are used to transfer control register address and input channel selection, which are then written in registers of main-switching chips. The writing process should end before the falling edge of the strobe signals, which corresponds to start of the next TF. At the falling edge of strobe signal new switching configuration is latched on the switch chips.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

A. GPS Receiver

GPS receiver, EPSILON Board OEM II [11], provides accurate and stable time and frequency signals for synchronization. It provides 1PPS (pulse per second) and 10MHz sine wave, and time of day output. Furthermore, the 10MHz frequency reference is cycle locked to the 1PPS, which is the standard UTC second.

B. Mindspeed Switch Chip

Mindspeed switch chip M21151/6 [12]; is a low-power CMOS, high-speed 144x144 cross point switch chip, each input/output capacity is up to 3.2 GHz with nominal aggregate capacity of 460 GHz). Each chip has an integrated clock data recovery (CDRs), input equalization, and built-in system test and broadcasting features. It offers programmable switch configuration to switch off unused portion thus reduces power consumption.

C. Implementation of Switch Controller

Opal Kelly XEM3001 [8] module is used for implementation of switch controller. The block diagram of implemented FPGA controller is shown in Figure 4.

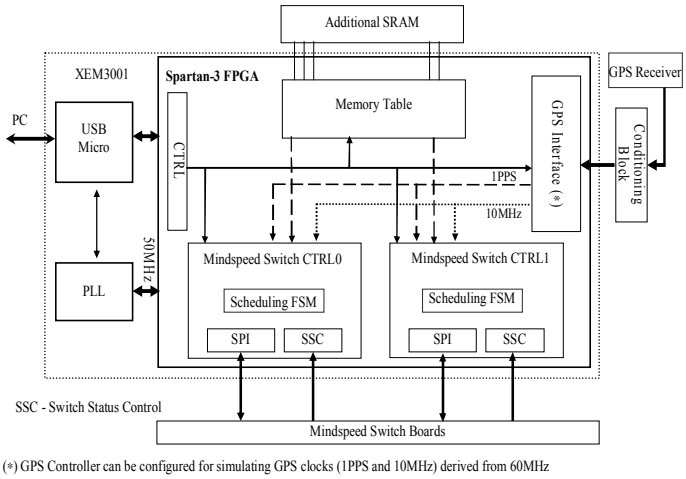

Figure 4: Block Diagram of Switch Controller

Very high speed integrated circuit hardware description language (VHDL) is used for the implementation of the FPGA sub module controller. The FPGA sub module blocks are control logic (CTRL), memory table, GPS Interface and switch controller. The implementations of all functional blocks in VHDL are synthesized in a single bit file which is stored on a PC and uploaded to the FPGA through a USB connection. It provides flexibility for updating the controller with new version of VHDL program. The bit file is downloaded to configure the FPGA every time the controller is started. Master clock for all blocks of Spartan-3 FPGA sub module is taken from an external PLL chip. The GPS Interface communicates with the GPS receiver via a serial link and allows remote configuration and status reporting. Two signals, 10 MHz sine-wave and 1PPS, are used as clock source for the scheduling FSM. GPS status control sub block are used to monitor correct locking of the receiver with signal from GPS. The "Conditioning Block" conditions clock signal coming from GPS receiver to make it compatible with CMOS pins of the Spartan-3 FPGA sub-module. USB control block (CTRL) is the interface between PC and the internal sub block's configuration registers implemented in other sub blocks on Spartan-3 FPGA. Memory table and switch control are described below.

1) Memory Table: Memory table is in a matrix format and stores switching configurations of all channels and for all TFs. The size and structure of the table depends on timing parameters, TF duration, number of TFs per TC, number of TC per UTC second and number of switching channels. Memory table can be configured from the graphical user interface (GUI) on PC through the USB communication link. For complete switch configurations (i.e., 144 channels per switching chip, with multiple chips), 216-kbits of block RAM [8] embedded in the FPGA will not be enough then an external SRAM is required.

2) Mindspeed Switch Controller: This block (Mindspeed Switch CTRL 0-1) represents a FSM that controls and connects all the sub-blocks implemented on FPGA module. Every clock cycle reads the input status register connected to a USB wire. This block consists of Scheduling FSM, SPI Interface and Switch Status Control sub blocks. These blocks are described below.

3) Scheduling FSM: One scheduling FSM is required for each Mindspeed switch chip and number of FSM modules can be implemented for number of switching chips. There are three important counters: TF counter, TC counter, and second counter. There are two reference input signals to controller through GPS receiver. One is 1PPS and another 10MHz. The counter resolution is 1/(10MHz)=100ns. All counters are 16bits. The TF duration is the integer multiply of time resolution. The number of TF, TC and TF duration can be modified by changing the corresponding parameters stored in register accessible via USB from PC. The product of TF duration*number of TF*number of TC must be equal to one UTC second. At the beginning of every TF, the FSM download

Figure 5: Timing diagram of Scheduling FSM

downloads a new switching configuration cyclically to the Mindspeed switch chip and activates hardware strobe via SPI interface. All the data are stored into a memory table which is read sequentially. When a new time-cycle starts, the pointer in the memory table is reset to the first memory location. This is the cyclic operation of the controller. 1PPS pulse resynchronizes all the FSMs. If the download terminates

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

before the 1Hz clock goes high the FSM stops in a waiting status until the 1PPS clock reset and restart whole process. It means any malfunction will last for maximum one UTC second. Figure 5 shows simulated hardware strobe generated by TF counters using 10MHz GPS clock. The right side of Figure 5 shows the effect of re-synchronization with the UTC 1PPS.

4) SPI Interface (SPI): It is used for two way data transfer between PC and Mindspeed chips through USB. This block implements a FSM that meets the timing/shape specifications given by Mindspeed M21151/6 data sheet [12]. Maximum input clock is 50MHz and SPI operate at 25MHz i.e. half of input clock. Data writing using SPI takes 20 SPI clock cycles (1 start bit, 1 RD/WR bit, 10 address bits, 8 data bits) that is 0.8µsec for downloading one channel configuration. For a configuration with 100µS TF length, a single FPGA implemented SPI peripheral is able to load up to 125 channel configurations. When more number of switch configurations loading is required, then it is possible to programme SPI clock up to the maximum speed (25MHz) accepted by Mindspeed chip. More than one SPI peripherals can be implemented in the same FPGA for increasing the speed and for scalability.

5) Switch Status Control (SSC): SSC sub block checks loss of signal (LOS) status of the chip. A LOS circuit is included on each input and detects whether valid data is present or not. LOS acts as an alarm and can be used to inhibit the signal into the switch core when the data to the input terminal is lost. If the input signal is clamped high or low, or if the difference between the input data rate and the programmed data rate is greater than approximately ±100 Mbps, the LOS alarm will be activated.

6) Software Interface: The interface between PC and FPGA board consists of two layers. First layer software is based on C++ classes of Opal Kelly library [8]. This type of interface uses more than one 8-bit pipe in order to transfer data. The second layer is a simple Graphical User Interface (GUI), developed using VC6++ [9]. The GUI is a set of dialog boxes that react to events like switch start/stop, input/output channel connection configuration, full or partial connection and communicate the events set by user.

IV. Implementation and Testing of Prototype

The switch prototype photograph is shown in Figure 6. The prototype setup's major components are streaming media sources with audio, video and text, a network interface for packets scheduling, 25km single mode optical fiber and multiple Mindspeed switch fabrics. The source side switch fabric is a two stage network implemented by connecting two Mindspeed switches. Two switches are controlled and configured by single FPGA switch controller. On receiver's side of the 25km single mode optical fiber there is another switch fabric consists of one Mindspeed switch, with one FPGA based switch controller and two receivers for playing the two movies with sound and subtitles. Standard single mode and multimode transceivers have been used for optical interconnections between various electrical to optical and vice versa conversion points. Bidirectional single mode transceivers have been used for interconnecting the two switches via the 25km single mode fiber.

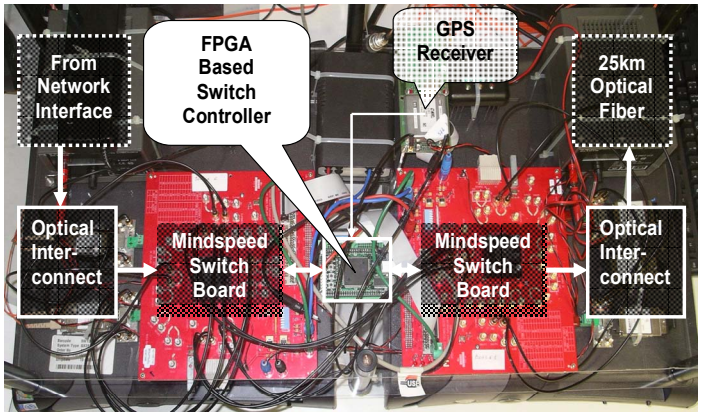

Figure 6: Photograph of Switch Prototype

The switch prototype experimental setup is shown in Figure 7. Two streaming media destined for two different receivers, one DVD movie with soundtrack and subtitles, while other animation movie with soundtrack, and are transmitted from one PC (shown IP Stream) using VLC media player [13].

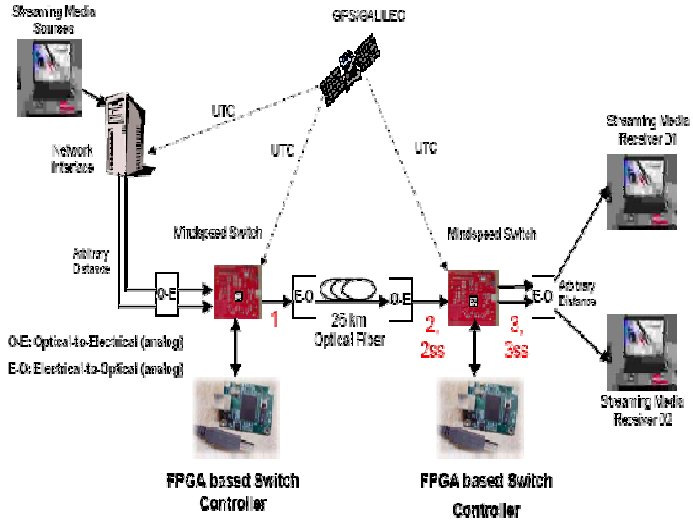

Figure 7: Switch Prototype Setup

Asynchronous packets are sent to network interface (detail of network interface is out of scope of this paper) which schedules the packets forwarding in synchronization with 1PPS signal. The streaming media packets from two sources are forwarded via optical link by network interface through transceivers to source side switch fabric during different predefined TFs. The packets are split into two streams by first stage switching. These separate streams from first stage are forwarded to second stage through electrical connection. At second stage both streams are again mixed by the cross-point switches. Then the mixed stream is transmitted to receiver's side through transceiver and 25km single mode optical fiber link. On the receiver side mixed stream received through transceiver is separated into two streams by switching.

1-4244-0357-X/06/$20.00 ©2006 IEEE
This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.

Separated streams are forwarded to two receivers, through transceiver kept at arbitrary distance. Switching of all three cross-point switches and network interface are synchronized with 1PPS received from GPS.

A. Experimental Results

The eye pattern test is quick method for visually examining the quality of serial signals, e.g., amount of timing jitter and amplitude variation in a serial data stream. This single display provides information about the eye opening, noise, jitter, rise and fall times, and amplitude. The two-dimensional shape can easily be compared to a standard mask. Figure 8-10 shows actual eye diagrams as captured on a real-time Tektronics TDS 6604 [14], 6GHz digital storage oscilloscope. The oscilloscope has inbuilt standard Gigabit Ethernet 1000 base SX-LX mask test. The measurements are carried at various locations (1, 2, and 3) indicated in Figure 7. At points 2 and 3 measurements were repeated using 25km single mode fiber (2ss, 3ss). Note that the boundaries of all these eye diagrams, which pass the compliance test, are within the ranges expressed by masks (in dark blue color). Moreover there is sufficient margin between the signal and the mask.

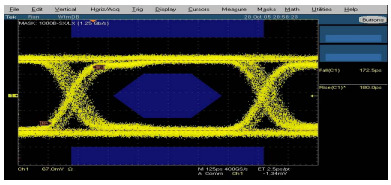

Fig. 8: Eye Pattern with Mask: 1000B – SX/LX (1.25Gb/s) at location 1

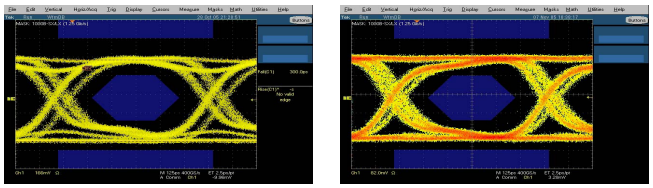

Figure 9: Eye Pattern with Mask 1000B–SX/LX (1.25Gb/s) at location 2 with Multimode (left) & single mode Fiber (right)

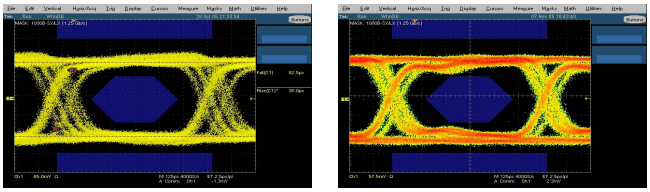

Figure 10: Eye Pattern with Mask 1000B–SX/LX (1.25Gb/s) at location 3 with Multimode (left) & single mode Fiber (right)

V. DISCUSSION AND OPEN ISSUES

This paper presented the design principles, implementation, and demonstration of a low cost ultra scalable TDS switch prototype, while focusing on the FPGA switch controller implementation. The FPGA is used for dynamically configuring the switch fabric and it is the main logic component of this switch prototype. The beauty is that the simplicity of this realization did not compromise two most desired performance properties for the future Internet: (1) switching scalability 10-100 Tb/s in a single chassis and (2) predictable QoS performance for streaming media and large (content) file transfers.

One of the open issues is the scalability of the FPGA switch controller for controlling large number of channels and switching components. Another issue is corresponding to the FPGA memory requirements for storing configurations for all channels and time frames. Memory is scalable because there is a provision for adding external SRAM chips. Another important aspect is time taken by controller to configure very large matrix of switching elements for supporting larger number of optical channels. There are two possible solutions. One is number of SPI Interfaces on the same FPGA for configuring multiple switching elements simultaneously. Another way is to develop a dedicated parallel connectors for fast configuration of large number of Mindspeed switching chips. However, this may come at the expense of more complex interconnection. Onther open issue is TF alignment when the delay between two switching channels is not integer multiple number of time frames. To overcome it TF alignment module is needed [5]. Bit synchronization was properly solved in the current prototype; however, further investigation is required in order to determine the most appropriate solution.

REFERENCES

[1] T. Miyazaki et al, "PROTEUS-Lite project: dedicated to developing a telecommunication-oriented FPGA and its applications," IEEE Transactions on Very Large Scale Integration Systems, vol 8, issue 4, Aug 2000, pp401– 414.

[2] A. Aloisio, F. Cevenini, V. Izzo, "An approach to DWDM for real-time applications," IEEE Transactions on Nuclear Science, vol 51, issue 3, part 1, Jun 2004, pp 526 – 531.

[3] M. Baldi, Y. Ofek, "Fractional lambda switching," Proc. of ICC 2002, New York, USA, Vol.5, pp 2692-2696.

[4] A. Pattavina, M. Bonomi, Y. Ofek, "Performance evaluation of time driven switching for flexible bandwidth provisioning in WDM networks," Proc. of Globecom 2004, Texas, USA, Vol. 3, pp1930-1935.

[5] M. Baldi, Y. Ofek, "Fractional lambda switching - principles of operation and performance issues", SIMULATION: Transactions of The Society for Modeling and Simulation International, Vol. 80, No. 10, Oct 2004, pp 527-544.

[6] D. Grieco, A. Pattavina, Y. Ofek, "Fractional lambda switching for flexible bandwidth provisioning in WDM networks: principles and performance", Photonic Network Communications, Issue: Vol. 9, No. 3, May 2005, pp 281-296.

[7] Fei Xue et al, "Design and experimental demonstration of a variable-length optical packet routing system with unified contention resolution," Journal of Lightwave Technology, vol 22, issue 11, Nov 2004, pp 2570 – 2581.

[8] XEM3001 - Xilinx Spartan-3 Experimentation Module, product information available at http://opalkelly.com/products/xem3001/

[9] Microsoft Visual Studio 6.0, product information available at http://msdn.microsoft.com/vstudio/Previous/vs6/default.aspx

[10] Du˘san ˇSuvakovi´c and Ilija Had˘zi´c, "An FPGA Application with High Speed Serial Transceiver Running at Sub-Nominal Rate", IEEE,2005.

[11] www.temex-telecom.com

[12] www.mindspeed.com

[13] http://www.videolan.org/vlc/

[14] www.tek.com

1-4244-0357-X/06/$20.00 ©2006 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2006 proceedings.