E-PON IPACT VHDL Implementation

A. Tonini, A. Mady and R. Alimur

Abstract—Ethernet Passive Optical Networks (E-PON) are an effective solution for low cost FTTH access networks because of their simplicity and high speed features. While scalability is still the main issue of E-PON next generation networks, as they allow the highest number of subscribers to share the same PON infrastructure, the E-PON Dynamic Bandwidth Assignement (DBA) schedulers for the Time Division Multiplexing (TDM) cover a key role for an efficient E-PON bandwidth allocation. For this purpose, we have implemented in VHDL code the Interleaved Polling Adaptive Control Time (IPACT) algorithm. Furthermore, to test the whole design, we developed a complete VHDL test bench in order to evaluate and optimize the scheduling algorithm. This test bench allowed us to consider the critical points of the scheduler and the optimization margins in a real hardware environment. The whole design is deployed over hardware E-PON devices using FPGA designs and considering the hardware area and time constraints.

Index Terms—FTTH, E-PON, IPACT, VHDL, FPGA , DBA, CaTV

1 INTRODUCTION

E-PON access networks are an easy and effective solution for high speed data transmission with minimal infrastructure costs. The PON architecture allows a single fiber deployment and simple passive optical components in a multi-user environment but requires a Time Division Multiplexing (TDM) transmission protocol. For E-PON networks the TDM data exchange has been implemented by a Multi-Point Control Protocol (MPCP) that requires a TDM time slot allocation from the Optical Light Terminal (OLT) to the single Optical Network Units (ONUs). For E-PON networks, the main issue is to share the same PON deployment among the highest number of subscribers while they guarantee the largest bandwidth per subscriber. Since the physical medium is shared, even the total bandwidth has to be shared among all users and this leads to the E-PON main bandwidth constraint.

In order to overcome this limit, a few technical solutions have been proposed such as the Coarse Wave Division Multiplexing (CWDM) and Wave Division Multiplexing (WDM) techniques [1], [2] and the 10 Gbps E-PON upgrade [3]. For all these solutions the common key-component is the DBA scheduler at the OLT side. The DBA scheduler manages the whole bandwidth allocation process, according to the MPCP protocol as defined from the IEEE 802.3ah standard group. It distributes the available bandwidth among the registered E-PON subscribers, taking into account their bandwidth request and the maximum allowed upstream transmission delay. Several DBA schedulers have been proposed in literature [4], [5]. The scheduler algorithm that better combines simplicity and assignment efficiency seems to be the IPACT scheme [6], [7]. All the proposed scheduling algorithms appear to have some benefits and drawbacks from a theoretical analysis but a complete scheduler performance analysis in a real hardware environment is still missing in literature. Indeed, since the first years after the E-PON standard definition, the need for a hardware test platform has led to project proposals like [8], as a basis of simple software simulations on the real system. The VHSIC Hardware Description Language (VHDL) provides a suitable tool to this scope and forms the basis for Field Programmable Gate Array (FPGA) devices hardware implementations.

An IPACT VHDL simulation has some undoubted advantages. It permits to evaluate the real scheduler performances in terms of real throughput efficiency and average packet delay. It makes easier some algorithm refinements in terms of logic delays and logic speed. It can be evaluated on real reconfigurable logic, suitable to be re-programmed with algorithm implementations upgrades. Furthermore the scheduler, as a key-component for actual and future E-PON devices, well fits a hardware accelerator, free from typical software latencies and delays. The increasing data rate of the next generation E-PON networks will increase the traffic load to be processed by the scheduler. An all-hardware implementation will guarantee bandwidth efficiency and prompt transmission slots assignment even for 10 Gbps solutions, removing any bottleneck from the MPCP process.

The rest of the paper is organized as follows. In chapter 2 we introduce the IPACT algorithm, in chapter 3 we propose a VHDL architecture for the IPACT scheduler and the synthesis results in terms of area and speed, in chapter 4 we describe the complete VHDL test bench to test the IPACT performance, including an OLT and 32 ONU units. Finally in chapter 5 we present the VHDL simulations results in terms of average packet delay as a function of the traffic load .

- *Andrea Tonini is with Electronic Media Communication SA S. Antonino-Switzerland.*
- *Alie Mady is with the Department of Physics, Cork University, Fort Collins, CO 80523.*
- *Rashid Alimur is with SAMSUNG Bamgladesh R&D Center (SBRC).*

```
For every received REPORT i of ONU i
{StartTIME = Tscheduled + A
IF StartTIME < localTIME + Tprocess THEN
        StartTIME = localTIME + Tprocess
Length = REPORT_Length + B
if Length > MaxLength then
        Length = MaxLength
GATE = {StartTIME, Length, forceREPORT = TRUE}
send GATE
Tscheduled = StartTIME + Length}
where
Tscheduled is the transmission time of the upstream channel
A is a Guard Time
B is Time constant due to the laser on, off and synchronization time and the time to process a report
MaxLength is the maximum Timeslot size
LocalTime is the local MPCP clock
```

Fig. 1. The IPACT algorithm in Limited Service formulation

2 IPACT ALGORITHM

The IPACT algorithm, in its Limited Service formulation, is well known and its basic code is reported in Fig.1. During the scheduling process, the OLT grants, by a GATE message, an ONU bandwidth request (REPORT) that asserts the number of bytes to be transmitted (Report_Length) in the upstream channel. The scheduler plans a transmission time (T*scheduled*) in response of the ONU requests. For every registered ONUs, the scheduler sets cyclically a StartTime value. At the local clock time StartTime, each ONU is allowed to transmit the upstream data stored in its memory. The scheduler also performs a preliminary check on this StartTime that must be greater than the minimum ONU GATE processing time, spent to extract all the scheduling information. The scheduler finally assigns each ONU a transmission bandwidth (Length) as a number of bytes and sends iteratively the GATE message. In the GATE message the forceREPORT variable, forces an ONU REPORT transmission during the assigned timeslot. In the Limited Service IPACT version a MaxLength grant has to be assigned in order to avoid excessive bandwidth assignment to a single ONU. The ONUs upstream channels are interleaved for transmission in the shared medium and iterate their packet transmission cyclically every Cycle Time. In this way the OLT polls every Cycle Time each ONUs and assigns them a bandwidth based on their request at the previous cycle. The Cycle Time also affects the upstream packets average delay. The leaving packets must wait a Cycle Time for their transmission and a maximum value (Max.Cycle) must be set in order to avoid excessive delays that could deteriorate the quality of services like Voice over IP. This also affects the maximum GATE assigned Length as in (1), where No.of.ONUs is the number of registered ONUs. Anyway, since low-bandwidth requests shorten the Cycle Time while high-bandwidth requests enlarge it, the IPACT algorithm adaptively allows the largest bandwidth requests to fill the lacks of the small bandwidth requests. In Fig.2 a graphical description of the cyclic bandwidth assignment process has been represented.

(1)

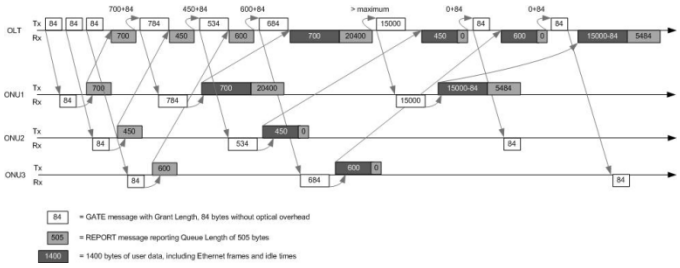

Fig. 2. IPACT process time representation

In the following chapter we provide a solution to implement the whole Limited Service IPACT scheduler in VHDL language at the OLT side. The basic blocks of this design will be the memories to store the REPORT and the GATE information and a processor that will build GATEs packets after having read and processed the REPORT information.

3 IPACT VHDL ARCHITECTURE

3.1 OLT Processor Architecture

At the OLT side, the top-level of the VHDL architectural design has been reported in the block scheme of Fig.3.

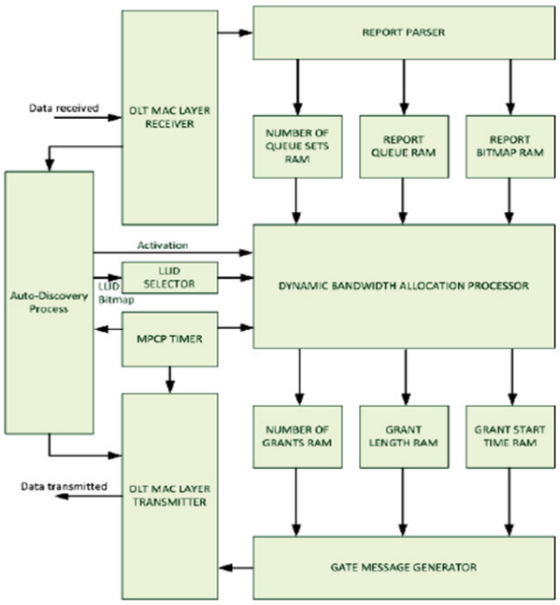

Fig. 3 Block scheme of the scheduler part with IPACT logic

Once the Auto-Discovery procedure [9], [10] is completed, it activates the Dynamic Bandwidth Allocator (DBA) IPACT Processor. The Logical Link IDentifier (LLID) selector reads the LLID Bitmap register defined from the Auto-Discovery Process that takes track of the registered ONUs and forwards the registered LLID addresses to the DBA Processor.

The OLT MAC Layer receiver block collects the MPCP packets coming from the upstream channel, detects the REPORT packets and sends them to a report parser. The parser extracts the REPORT Number of Queue Sets, the REPORT Queue and REPORT Bitmap fields. The parsed fields are then stored in a RAM memory, driven from the dynamic Bandwidth Allocation Processor. This processor handles the REPORT information stored in the dedicated

RAM and builds all the fields of the GATE message. Then, it stores the grant information, consisting of the GATE Start Time and the GATE Length, in the Grant memory. The GATE Message Generator block reads the grant fields and builds the whole GATE message, ready to be sent downstream to the ONUs by the OLT MAC Layer Transmitter. The whole architecture has been designed in a structural and modular way, in order to allow portability of the single components. This design strategy will make easier future design upgrades with DBA scheduler different from IPACT, without affecting the remaining part of the design.

3.2 IPACT Block

Focusing more deeply inside the DBA processor, its architecture is composed of a Finite State Machine (FSM), a Request Length processor for Grant Length calculation and the an IPACT logic for grant Start Time calculation. The Finite State Machine (FSM) reads the information from the REPORT RAMs and drives a Request Length Processor that re-orders the REPORTs list. After simple Digital Signal Processing (DSP) calculations, the Request Length Processor outputs the Grant Length.

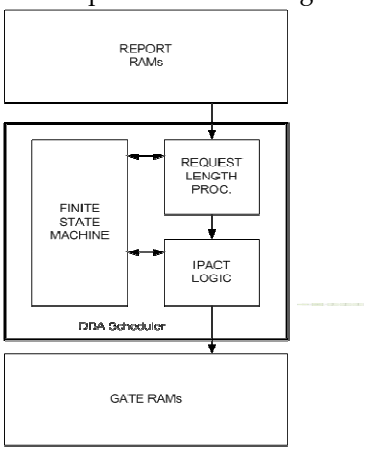

Fig. 4. OLT VHDL internal Processor architecture

The IPACT logic receives this Grant Length value, performs a DSP treatment and finally provides the GATE RAM with the GATE fields sequence. In Fig.4, a block scheme of the whole IPACT DBA scheduler processor shows its internal architecture.

The FSM manages the whole process, from the memory addressing to the GATE fields synthesis. A detailed flow diagram in Fig.5 shows the state sequence of the DBA scheduler FSM. The machine starts in the IDLE state. At the START command it goes into the initialization (INIT) step, where all the variables get their default values. In the following states the machine waits for the arriving reports (WAIT FOR REPORT) till a new REPORT arrives. At the REPORT arrival, it reads the REPORT fields in their memory locations during the states READ NUMBER of QUEUE SETS MEMORY, READ BITMAP MEMORY and READ QUEUE MEMORY and drives the Gate Length and Gate Start Time calculation. After that, it writes these values into the GATE memory

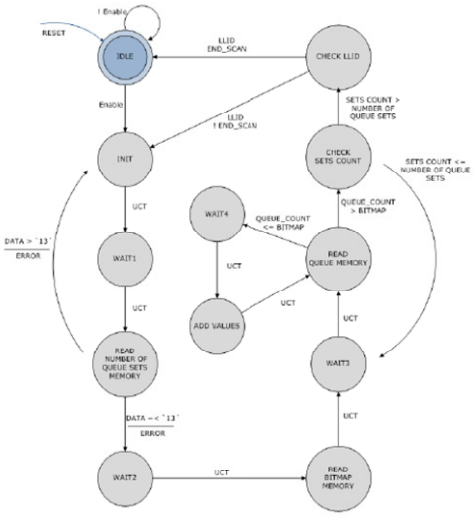

Fig. 5. OLT VHDL Processor FSM Flow diagram

during the WRITE NEW DATA state. The FSM checks the result in the CHECK SETS COUNT and CHECK LLID states and goes back into the IDLE state. This process iterates for every registered ONU owning a Logic Link Identifier (LLID). The whole architecture has been synthesized and downloaded into a low cost FPGA with 25K cells. In Table 1 there is a report of FPGA area resources utilization.

TABLE 1
IPACT SYNTHESIS RESULTS

IPACT SCHEDULER FPGA IMPLEMENTATION	
TOTAL LOGIC ELEMENTS	2,792
TOTAL COMBINATIONAL FUNCTIONS	2,755
DEDICATED LOGIC REGISTERS	579
TOTAL REGISTERS	579
TOTAL PINS	104
TOTAL MEMORY BITS	35,328
EMBEDDED MULTIPLERS	0
TOTAL PLLs	1

IPACT FPGA implementation Resource usage.

4 THE VHDL TEST BENCH

Based on the architecture defined in the previous chapter, we prepared a VHDL Test Bench to test and validate the IPACT scheduler block. To accomplish this task, we carefully implemented a VHDL code of the whole IEEE802.3ah standard at the OLT side and at the ONU side as well, defining the real MPCP packet time and delay overheads and replicating the standard Finite State Machines. With reference to the complete standard, we only neglected the Round-Trip RTT_i time of the i^{th} ONU since we consider all the ONUs at the same distance from OLT. This assumption won't affect the simulation results,

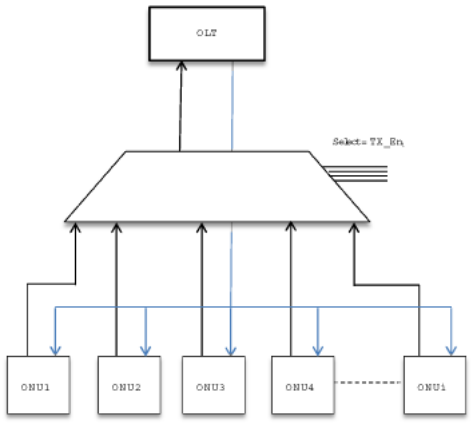

Fig.6. VHDL block scheme of the complete test bench

since in the real case, a ranging process equalizes the ONUs distances from the head end.

In Fig.6 a block scheme of the whole test bench is sketched. The fiber optic PON network has been simulated with a digital multiplexer (MUX) for the upstream channels and with a broadcast transmission line for the downstream link. The MUX forwards to the output the input upstream channel selected by the Transmit_Enable signals of the ONUs transmitters. Since all input signals are synchronous, the input selection process will be univocal. The ONUs VHDL blocks contain a Physical Layer (PHY) with internal data RAMs enclosing the Ethernet packets to be transmitted.

TABLE 2
IPACT SIMULATION PROPRIETIES

Parameter	Description	Value	Units
N	Number of ONUs	32	
RU	Bit Rate of ONU-User link	1	GBPS
RN	Bit Rate of OLT-ONU link	1	GBPS
Q	Buffer Size in ONUs	16	Mbytes
G	Guard Interval between Timeslots	116	TQ
T	Max Cycle Time	2	ms
		125000	TQ
Tproc	Process Time	1024	TQ
MaxLength	Maximum Gate Length	3906	TQ

In Table 2 we reported the simulation parameters for the ONUs at the physical layer and data link. The ONU User Rate (RU), at the Ethernet PHY, is 1Gbps like the E-PON Network Rate (RN). The ONUs enclose 16 MBytes internal FIFO buffers for the data storage. We assumed 32 ONUs in the subnet since that is the maximum number of users an E-PON network can provide. The IPACT maximum cycle time has been set to 2 ms to guarantee an acceptable packet delay even for voice over IP signals. The MaxLength parameter comes from this value, dividing the Max Cyle Time by the number of ONUs. The process time and the Guard time are defined from the IEEE802.3ah protocol and indicate respectively the time

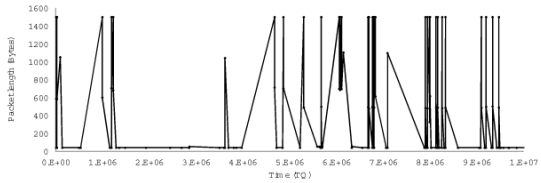

Fig. 7. Upstream Burst Ethernet packet sequence

to process the gate signal at the ONU side and the guard time for laser synchronization and clock recovery instances. We inserted into each ONU a Long Range Dependent (LRD) traffic generator, typical of burst-mode Ethernet traffic. The Ethernet packets have been produced by acquiring samples of a real upstream traffic, using the Ethereal tool. In Fig.7 a temporal sequence of the upstream packets transmitted from an ONU, has been reported. The whole Test Bench represented in Fig.6 has been simulated and debugged by a VHDL time-event simulator. In Fig.8, we reported the time sequence of the upstream packets sent from the registered ONUs and the OLT received packets, at the MUX output, for a 16 ONUs link (easier to view than a 32 ONUs link). It can be noticed that the ONUs upstream packets are TDM synchronized by the scheduler according to the IPACT algorithm. The largest packets contain upstream data while the shortest are single 64 Bytes REPORT Data Units (DU), when no data are present in the ONU FIFOs.

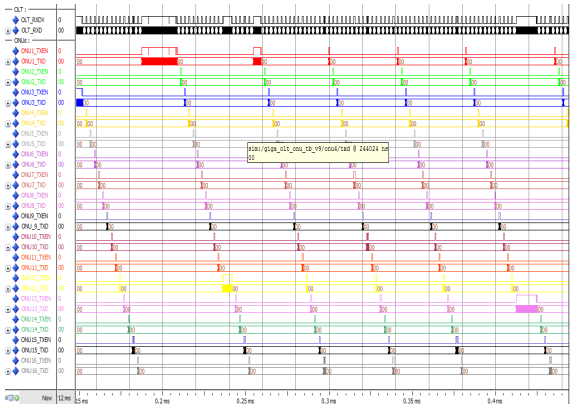

Fig. 8. Wave simulation graph of the Online scheduling process for 16 ONUs

5 RESULTS

Having set up the test-bench described in the previous chapter, the scheduler performances have been evaluated by a VHDL simulation on a large number of data samples. In order to get the real time values of the tested quantities, we integrated into the design some measuring units, whose purpose is only to monitor and measure the parameters under test. We have characterized the whole network in terms of Average Packet Delay as a function

of the Network Average Offered Load (NAOL). NAOL is a measure of the total network traffic load normalized to the network capacity. NAOL is defined in (2), where N is the ONU number; R_U is the user bit rate, R_N is the network bit rate, ϕ_i is single ONUs offered load, normalized to R_U, while their values have been defined in Table 2. In our measurements we averaged the transmitted bit rate on the measurement time to calculate the ONUs effective offered load. The ONUs average delay is the average time difference between the arrival time and the departure time of the data packets stored in the FIFOs at ONU side.

$$\phi = (R_u/R_N)\ \sum_{i=1}^{N} \phi_i \qquad (2)$$

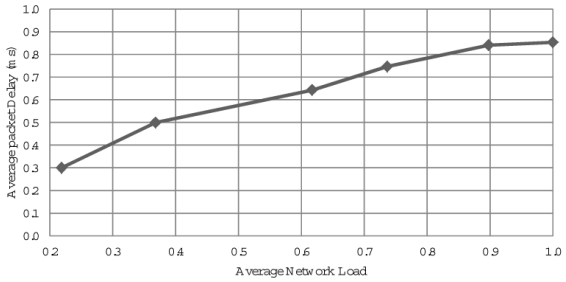

Fig.9 Average Packet Delay as a function of the Network Offered Load for a 32 ONUs E-PON Network

To test the IPACT scheduler, we have changed the 32 ONUs upstream aggregate data traffic by varying the transmission rate of the single ONUs and the applications running on it. The ONUs upstream traffic information, as represented in Fig.7, has been stored in local memories and read from the ONUs Physical Layers. We have increased the network NAOL from 0.2 up to 1 and an average of 16 acquisitions for each ONUs has been measured. The final delay result comes from a further average on the 32 ONUs average packet delay array.

It was not possible to get the highest NAOL value of the E-PON network under test, simply capturing the real upstream traffic of a standard network. Indeed ADSL or DOCSIS data networks can't reach the E-PON bit rates, even if high bandwidth consuming software applications are running on them. To overcome this limits, we introduced some bit-rate multiplicators that allowed us to reach network NAOL up to 1. As can be noticed in Fig.9,

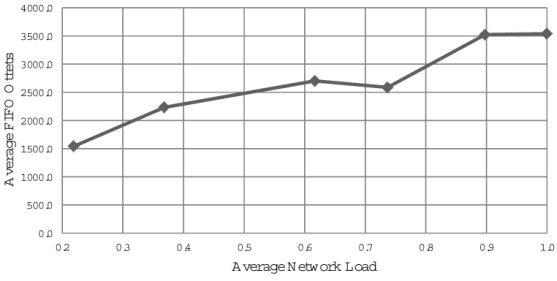

Fig. 10. Average FIFO stored 8-bits words number as a function of Network Offered Load for a 32 ONUs E-PON Network

the average packet delay grows from 0.3 ms for 0.2 NAOL up to 0.85 ms for 1 NAOL. At 0.9 NAOL the curve starts to saturate, due to the fact that the algorithm, in the limited service version, fixes an upper limit for the ONU upstream transmission time-slot length, in order to fit the maximum E-PON available bandwidth.

In Fig.10, a plot is represented with the average number of data octets stored in the ONU-side FIFOs, ready to be sent. This number gives us information about the data buffered in the FIFOs as a function of NAOL. The data storage grows with the NAOL even if this grow depends on the running application. A more regular upstream traffic like the one generated from Peer-to-Peer applications, seems to produce a lighter load on the input buffers than a more bursty traffic, even if the former produces a higher NAOL. In Fig.11 the average residual data octets after time slot allocation have been represented. Not all of the data stored in the ONUs FIFO are emptied by the assigned transmission timeslot for two main reasons: 1) the data continue to access the ONU FIFOs even once the requested timeslot has already been assigned, 2) the maximum fixed timeslot for the Limited Service IPACT, limits the number of words that can be sent in the upstream link. For these reasons, the Average Residual Octets give a measure of the IPACT algorithm ineffectiveness.

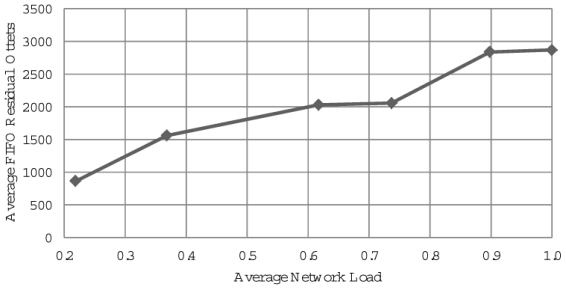

Fig.11 Average FIFO residual 8-bits words number as a function of the Network Offered Load for a 32 ONUs E-PON Network

As can be observed in Fig.11, the Average Residual Octets grow with the AOL, giving Residual Octets values comprised between 50 %, at light loads and 80 % at higher loads, of the whole stored octets. Of course the process becomes more inefficient in wasting the ONUs FIFOs at high load, when the output FIFOs data traffic can't compensate the high input data rate. We think that these percentages can be significantly improved, using an Elastic Service IPACT algorithm instead of the Limited Service formulation of the present paper. This check will be matter of future work.

7 Conclusion

A complete VHDL design of E-PON IPACT scheduler has been designed and tested following the IEEE 802.3ah standard. The scheduler has been synthesized for FPGA implementation and simulated with a complete Testbench that includes also the ONU side, in order to evaluate the real scheduler performances in a real hardware

environment. The IPACT scheduling processor allowed us to get short Average Packet Delays for a 32 ONUs E-PON and provided the possibility to improve the performances with simple adjustements of the hardware delays. Further refinements of the scheduler will be evaluated in future work, and will include prediction features and statistical packet parsing in order to improve the efficiency under heavy unbalanced network loads.

ACKNOWLEDGMENT

The authors wish to thank Mr. Christian Fochetti for the great help in software set up and the Advanced Learning and Research Institute of University of Italian Switzerland (USI) that started and supported this work.

REFERENCES

[1] A. Tonini, M. Al Hissi, and A. El-Din Mady "EPON CWDM Optical Light Expander with Adaptive Prediction Scheduler" *Proc. IARIA Symp. Conference on Access Networks, Services and Technologies (ACCESS '10)*, pp. 197-200, Sept. 2010

[2] A. Banerjee, et al., "A review of WDM-PON Technologies for Broadband Access" *OSA Journal of Optical Networking*, vol. 4, no.11, pp. 737-758, 2005

[3] S. Sallent, M. De Andrade, and L. Gutierrez "New Trends On Optical Access Networks: DBA for 10G EPONs and Long- Reach PON" *Proc. Euro-NF Symp. Conference on Network and Optical Communications (NOC '09)*, pp 487-494, Jun. 2009

[4] D. Nowak, J. Murphy, and P. Perry, "Adaptive Dynamic Bandwidth Allocation Algorithm for Ethernet PONs," in *Proceedings of 30th European Conference on Optical Communication*, vol. 3, pp. 740-741, Sept. 2004

[5] B. Chen, J. Chen, and S. He, "Efficient and Fine Scheduling Algorithm for Bandwidth Allocation in Ethernet Passive Optical Networks", *IEEE Journal of Selected Topics in Quantum Electronics*, Vol. 12, No. 4, pp. 653-660 , 2006

[6] G. Kramer, B. Mukherjee, and G. Pesavento, "Interleaved Polling with Adaptive Cycle Time (IPACT): A Dynamic Bandwidth Distribution Scheme in an Optical Access Network", *Photonic Network Communications*, vol. 4, no. 1 pp. 89-107, January 2002

[7] Y. Zhu and M. Ma, "IPACT With Grant Estimation (IPACT-GE) Scheme for Ethernet Passive Optical Networks," *J. Lightwave Technol.* 26, pp. 2055-2063, 2008

[8] T.S. Lyster et al., "Generic Ethernet Passive Optical Network Platform for Advanced Protocol development", *Proc. Symp. Conference on Applied Informatics and Communications (WSEAS)*, pp. 126-131, Sept. 2005

[9] A. Mady and A. Tonini, "A VHDL Implementation of ONU Autodiscovery Process for PON" *Int. Conference on Networking and Media Convergence (ICNM09)*, pp. 36-41, 2009

[10] A. Mady and A. Tonini, Ethernet Passive Optic Network (EPON) System : *A VHDL Implementation of ONU Autodiscovery Process of the IEEE802.3ah MPCP Protocol*, Lambert Academic Publishing, 2010

Andrea Tonini received a M.Sc. degree in Opto-Electronic Engineering from the Politecnico of Milan University in 1995.
In 2006 he earned a M.Sc. in Embedded Systems Design from the Advanced Learning and Research Institute (ALaRI) of Lugano, Switzerland. Actually, he is working as a R&D Engineer in EMC Electronic Media Communication SA in St. Antonino- Switzerland, where he develops R&D projects concerning FTTH networks, next generation CaTV DOCSIS - based fiber optic networks and high speed digital applications. He owns several patents and papers.

Alie El-Din Mady received a B.Sc. degree in electronics and communication engineering with honour degree from Alexandria University, Alexandria, Egypt, in 2006, and an M.Sc. degree in embedded system design from Universita della Svizzera Italiana (USI), Advanced Learning and Research Institute (ALaRI), Lugano, Switzerland, in 2008. He is currently a Ph.D. researcher at Cork Complex Systems Laboratory (CCSL), University College Cork (UCC), Cork, Ireland, where his research aspect is component-based embedded system for distributed wireless sensor-actuator networks. His main research interests are embedded systems, cyber-physical system and HDL design.

Sk. Alimur Rashid received a B.Sc. degree in Computer Engineering in 2007 from the American International University - Bangladesh (AIUB), Dhaka, Bangladesh. In February, 2010, he earned his M.Sc. in Embedded Systems Design from the Advanced Learning and Research Institute (ALaRI), Lugano, Switzerland. Presently, he is working as an Embedded Software Engineer in Samsung Bangladesh R&D Center (SBRC) in Bangladesh.