

A VHDL Implementation of ONU Auto-discovery Process for EPON

Alie El-Din Mady

CCSL, Computer Science Department,
University College Cork,
Cork, Ireland.
mael@cs.ucc.ie

Andrea Tonini

Electronic Media Communication SA,
S.Antonino, Switzerland.
andrea.tonini@lu.unisi.ch

Abstract—Recently EPON has been introduced to represent the convergence of low-cost Ethernet equipment and fiber infrastructure, it is appeared to be the best candidate for the next-generation access network. Auto-discovery process is an important process in EPON system. Indeed, this paper provides a MAC design following the IEEE 802.3ah standard of the MPCP auto-discovery process at the ONU side. It describes as well the validation and verification steps using an OLT model and finally, the synthesis results are presented.

I. INTRODUCTION

Full Services Access Network (FSAN) is used to access great variety of services (i.e. Video on Demand (VoD), Television, IP television, Internet, ...), one of its implementation is the Ethernet Passive Optical Network, EPON has been standardized under IEEE 802.3ah. It uses low cost fiber infrastructure i.e. Fiber to the Premises. FFTP provides three types of technologies: EPON, Broadband PON (BPON) and Gigabit PON (GPON) [1]. The EPON technology is mainly used in homes within high population density, the main reasons as explained in [2] [3] are:

1. EPON carries variable-length packets without fragmentation, however both BPON and GPON break the packets into multiple fragments [4].
2. EPON can meet all the service requirements for residential use. All the services required for residential use like telephone, fax, Internet access, and Cable Television (CATV) can be provided by EPON. Using Voice Over IP (VOIP) with embedded Integrated Access Device (IAD) or through TDM (E1/T1) over IP, EPON provides telephone services; It can provide internet access in a straightforward way because of its Ethernet based transport nature; it provides CATV service through a third wave length which is dedicated for analog CATV transport. The fax service can be ensured either through telephone line over IAD or via VOIP T.38.
3. EPON is very easy for Next Generation Networking (NGN) evolution in particular for layer two IP support. One of the key features of the NGN is that it will be packet based, while IP network is widely accepted as the transport network of NGN.
4. The EPON equipments are cheap. This can be explained by the fact, Ethernet is one of the widely used

layer two technology, more than 80% of the IP traffic is now carried by Ethernet. Ethernet protocol is very mature and many types of Ethernet cards have become commodity.

Fig. 1 shows EPON system elements, two types of elements are used:

1. *Passive elements*: are allocated in the optical network distribution such as: single-mode fiber-optic cable, passive optical splitters/combiners, connectors and splices.
2. *Active elements*: are allocated at the end points of the network such as: Optical Line Terminal (OLT) allocated at the Central Office (CO) and Multiple of Optical Network Unit (ONU) allocated at the end user.

The optical signal travels from the OLT through the optical fiber cable to the optical splitter that converts it into multiple optical signals. This scenario is called *downstream*, while the opposite scenario is called *upstream* [5]. EPON provides the time allocation for the ONUs using the Multi-Point Control Protocol (MPCP), one of the important processes in MPCP is Auto-discovery process which is used to identify a new ONU added to the system.

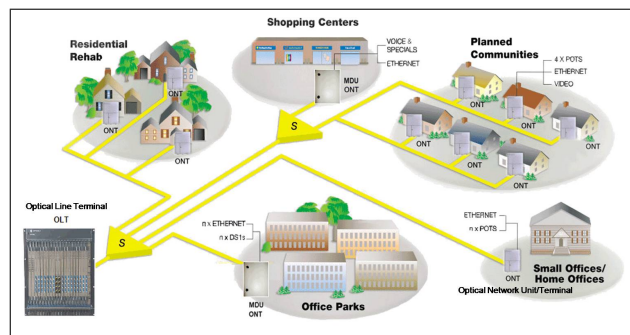


Fig. 1. EPON Architecture

The Objective of this article is to implement the auto-discovery process in VHDL at the ONU side. The auto-discovery mechanism and its performance metrics are explained in Section 2; the VHDL implementation, the role of each block in the RTL design and the design scenarios are illustrated in Section 3; Section 4 provides the design validation/verification and the synthesis results and finally, Section 5 concludes the paper work.

II. MULTI-POINT CONTROL PROTOCOL

MPCP has been introduced in IEEE 802.3ah standard to support a timeslot allocation; this protocol has improved the Ethernet protocol (IEEE 802.3) by adding five new Medium Access Control (MAC) control messages:

- Discovery/Normal GATE message is used to assign a transmission timeslot (discovery window, grants) for an ONU.
- REPORT message is used by an ONU to convey its local conditions like buffer occupancy to the OLT in order to provide an intelligent allocation decision.
- REGISTER_REQ message is used to respond to discovery GATE message.
- REGISTER is used to assign a unique Logic Link ID (LLID) to a newly discovered ONU.
- REGISTER_ACK message serves as an ONU's final registration acknowledgment.

MPCP has introduced two main processes, *Auto-discovery* process to detect a new ONU and *Bandwidth Assignment* process [6] to assign the suitable bandwidth for each ONU.

A. Auto-discovery Process

Auto-discovery mechanism is used to detect the new (uninitialized) ONU in EPON system or if the system is powered up. This mechanism provides the round-trip delays, MAC addresses and the resources of the ONUs to the OLT in order to reserve resources for each ONU. As explained in the outlined IEEE 802.3ah draft the Auto-discovery process performed as following [7]:

1. The OLT uses discovery GATE messages to declare discovery windows for the ONUs to submit their registration requests (REGISTER_REQ).
2. The ONU randomly selects an offset from the beginning of the discovery window to transmit the REGISTER_REQ message.
3. The ONU determines the result of its registration request by listening to OLT acknowledgment (REGISTER). A collision is depicted if the expected REGISTER message does not arrive. The ONU uses an exponential back-off mechanism to select a subsequent discovery window to make another attempt.
4. The OLT sends the grant times for the registered ONU using GATE message.
5. The ONU acknowledges the REGISTER with a REGISTER_ACK. Then ONU can request bandwidth from the OLT using the REPORT message, without the risk of contention.

B. Auto-discovery Mechanism Performance Metrics

Three metrics are used to evaluate the MPCP auto-discovery performance. *Registration delay* is defined as the time between an ONU attempts to register to the OLT and the time it gets registered, *synchronization* and *discovery window efficiency* is the ratio between the overall discovery window capacity and the actual capacity used by the registration process. However, there is a tread off between these metrics, for example, increasing discovery window

size and/or discovery window frequency may reduce delay but will cause lower discovery window efficiency at the same time [8].

III. IMPLEMENTATION OF ONU AUTO-DISCOVERY PROCESS

Auto-discovery process has been divided into four main processes as it is shown in Fig. 2: discovery gate generation, request reception, register generation, and final registration process. The Auto-discovery process scenario is described as following [4]:

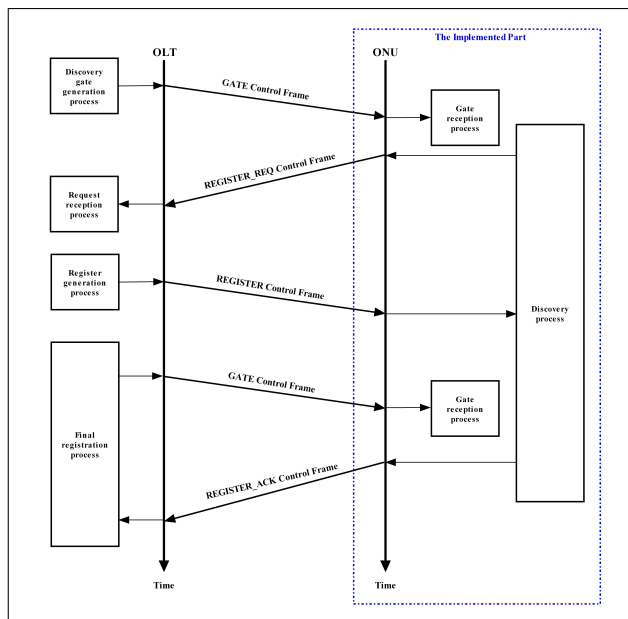


Fig. 2. Auto-discovery Messages Exchange

1. OLT broadcasts the discovery GATE message using the discovery gate generation process to all ONUs (with LLID = (7F FF)H). Using the gate reception process the ONU can detect the GATE discovery message, if the ONU is uninitialized, it saves the grant values in order to calculate the discovery window for sending a registration request to the OLT.
2. The uninitialized ONU sends the REGISTER_REQ message to the OLT to inform it about the maximum pending grants. This message remains buffered until the discovery grant is activated.
3. Using register generation process, the OLT sends the REGISTER message including the LLID (uniquely assigned to the ONU). REGISTER is addressed to an individual ONU but it is transmitted on the broadcast channel, because the unique LLID has not been assigned to the ONU yet.
4. The OLT sends point to point GATE message carrying the transmission grant times to the ONU. The received grants are stored until the local MPCP clock reaches the grant start time value.
5. When the grant gets activated, the ONU finally sends a REGISTER_ACK to the OLT to finalize the registration process.

Fig. 3 shows the Register Transfer Level (RTL) block diagram of the Auto-discovery process design written in VHDL. In following we explain the role of each block in the ONU Auto-discovery architecture and describe a tracing scenario of upstream /downstream communication. For more implementation details we refer the reader to [9].

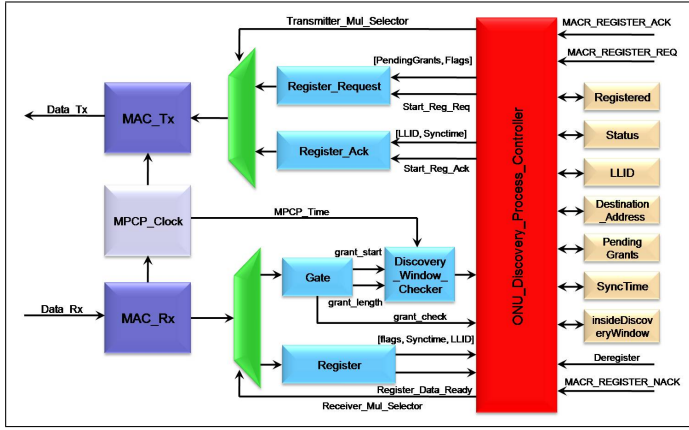


Fig. 3. ONU Auto-discovery Process Architecture

A. RTL Blocks Description

The main design blocks are:

1. *MAC_Rx*: is used to receive the transmitted frame nibble (4 bits) by nibble and parse it on flow. It detects the time stamp and the type of the received message (GATE, REPORT, REGISTER, REGISTER_REQ, REGISTER_ACK), and calculates the frame Cyclic Redundancy Check (CRC32) [10] as one of the most powerful error-detecting.
2. *MAC_Tx*: is used to create a transmitted frame using frame data (from 40 byte RAM), frame header (destination address, source address, opcode and time stamp), preamble, Start Frame Delimiter (SFD) and CRC32. The MAC_Tx sends the frame nibble by nibble with active data available signal (RXD).
3. *Gate*: is the ONU Gate reception process, it is responsible for parsing the received GATE message. This block is triggered by the gate opcode signal from the MAC_Rx, then Gate block either provides the start time (Grant_Start.Time) and the length of the discovery window (Grant_Length.Time) in discovery GATE case or trigger grant_check signal after checking the received grants in normal GATE case. The Gate block saves the received grants in a Grant buffer after checking the validation of these grants as following [4]:
 - (a) $\text{Grant_Start_Time} - \text{local time} < \text{max_future_grant_time}$
 - (b) $\text{Grant_Start_Time} - \text{local time} \geq \text{min_processing_time}$
 - (c) $\text{Grant_Length_Time} > \text{laserOnTime} + \text{SyncTime} + \text{LaserOffTime} + \text{TailGuard}$
 - (d) Discard all discovery gates after ONU has been registered: NOT (discovery and registered)

4. *ONU_Discovery_Process_Controller*: is the main controller, it is used to control the whole MAC layer components and interface with the upper layer (*Nios Embedded Processor*) [11].
5. *Register*: is the ONU register reception process, it is used to parse the received REGISTER message, after getting triggered from MAC_Rx, and provide echo pending grants, sync time, flags and assigned port to the ONU main controller.
6. *Register_Ack*: is the register acknowledge generator, it is used to load (after being triggered) the RAM transmitter with REGISTER_ACK frame data (LLID, SyncTime) received from the main controller.
7. *Register_Req*: is the register request generator, it is used to load (after being triggered) the RAM transmitter with REGISTER_REQ frame data (pending grants, registration/deregistration flag) received from the main controller.
8. *MPCP_Clock*: is used to generate the time stamp for each transmitted frame. In order to synchronize between the OLT and ONU, the ONU initializes its MPCP_Clock with the time stamp parsed from the discovery GATE message, and hence the system is synchronized for every discovery request.
9. *Discovery_Window_Checker*: is used to inform the main controller, if the ONU is in the discovery window or not. The Gate block parses the received GATE and provides the discovery window start time and length to the Discovery_Window_Checker in order to decide, rely on the MPCP_Clock current value, if the ONU can send REGISTER_REQ or not.
10. *Upper Layer Interface*: in order to connect the MAC layer with the upper layer, Registered, Status, LLID, Destination_Address, PendingGrants, SyncTime and InsideDiscoveryWindow registers have been implemented to get/provide information from/to the MAC/upper layer, in addition the signals (Deregister, MACR_REGISTER_ACK, MACR_REGISTER_REQ and MACR_REGISTER_NACK) have been implemented in order to receive upper layer tasks.

B. Upstream/Downstream Tracing Scenario

1. In the beginning the MAC_Rx waits till receives a GATE message, then it parses the received GATE to get the message time stamp, which gives the message transmitting time, and by uploading the ONU MPCP_Clock with this time stamp, a synchronization can be achieved.
2. The ONU_Discovery_Process_Controller configures the demultiplexer in order to transfer the received GATE message from the MAC_Rx to the Gate reception process.
3. Using the Gate reception process, the GATE message is parsed to detect the grant start time and the grant length, these information are then transmitted to the Discovery_Window_Checker.
4. The Discovery_Window_Checker checks if the ONU current time is inside the discovery window or

not by comparing the current time with the sum of grant start time and the grant length. The ONU_Discovery_Process_Controller gets informed about the duration of the discovery window using a boolean signal that comes from the Discovery_Window_Checker, hence, the ONU can receive a dynamic discovery window [6].

5. If the ONU_Discovery_Process_Controller is inside the discovery window, this means that the ONU can send a request to the OLT to get registered, hence if the ONU_Discovery_Process_Controller has received the register request signal (MACR_REGISTER_REQUEST) from the higher layer then it reads the pending grants and registration/deregistration flag from the upper layer registers and provides this data to the Register_Request generator.
6. If the Register_Request received a firing signal from the main controller, it collects the required data of REGISTER_REQ message provided from the main controller and sends it to the transmitter multiplexer.
7. Using the transmitter multiplexer, the ONU_Discovery_Process_Controller selects the correct routing of the REGISTER_REQ data and hence the MAC_Tx transmits REGISTER_REQ to the OLT.
8. The ONU_Discovery_Process_Controller selects the correct routing for the receiver demultiplexer in order to transfer the received REGISTER message to Register reception process.
9. The Register reception process parses the REGISTER message and provides the sync time and the LLID, then gives them to the main controller which saves them in the upper layer registers.
10. When MAC_Rx receives a GATE message, the Gate reception process checks the received grants and sends the checking decision as a boolean signal to the main controller. When the main controller receives the registration decision from the higher layer (MACR_REGISTER_ACK or MACR_REGISTER_NACK), it sends the LLID and synctime values to Register_Ack generator. The REGISTER_ACK message is sent to OLT using the MAC_Tx via transmitter multiplexer.
11. If the ONU_Discovery_Process_Controller received Deregister signal from the higher layer then the ONU sends to the OLT asking for a deregistration.

IV. DESIGN VALIDATION AND VERIFICATION

In order to verify and validate the ONU Auto-discovery design, we have followed three steps, simulation, emulation and hardware implementation.

A. Design Simulation

The functionality of the design has been tested using *ModelSim* [12]. For this end we have developed a Test Bench acted as an OLT side to be integrated with the ONU design as shown in Fig. 4. The tester is written in

completely behavioural way and hence it is impossible to be implemented in a hardware platform, for instance if the OLT waits to read a memory contents in order to generate the GATE frame; the memory will be considered as a text file containing the GATE frame data. The tester reads the file and sends the data to the Design Under Test (DUT) as following:

```
while (not endfile(Gate_Frame)) loop
  readline(Gate_Frame,Line);
  hread(Line,RAM_Data);
  S_RAM_data    <= RAM_Data;
  S_RAM_wraddress <= S_RAM_wraddress + 1;
  S_RAM_wren    <= '1';
wait for 10 ns;
end loop;
```

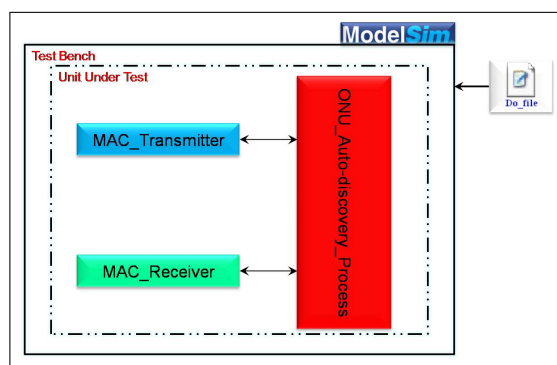


Fig. 4. Test Bench Architecture

Fig. 5 shows the simulation results of the Test Bench. The result are divided into four groups of signals; transmitter signals, receiver signals, Auto-discovery process signals and discovery controller signals. It is obvious that the registration process is completed at the "registered" state in the main discovery process controller and hence the process generates a signal (appear at 4200 ns) as a flag for finishing the registration process at the ONU side.

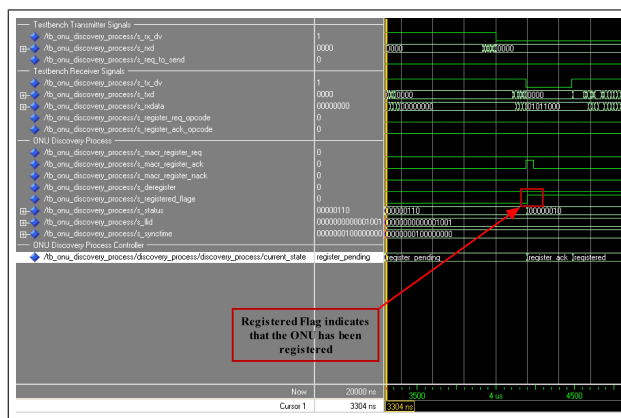


Fig. 5. Simulation Results

B. Design Emulation

The OLT Emulator design is used to test the Auto-discovery design on the real hardware, it uses two leds

at each side (ONU side and OLT side) to test the registration completeness. Fig. 6 shows the emulation design; it contains two main blocks, ONU Auto-discovery process which has been discussed in Section 3 and OLT emulator. The OLT emulator acts as an OLT model to provide the same OLT behaviour to the ONU; it is designed using OLT transmitter, OLT receiver and OLT controller. OLT transmitter and receiver are exactly the same have been used at the ONU; however the OLT controller is used to load the transmitter RAM (40 bytes) with the required packets to be sent to the ONU or to check the received packets from the ONU. Fig. 7 depicts the OLT emulator FSM as following:

1. In the beginning the OLT controller waits in linewait1 state for 16 nibbles (8 bytes) to provide a transmission guard and then in MACR_REGISTER_REQ state, the OLT sends MACR_REGISTER_REQ signal acting as an ONU upper signal.
2. In Load_GATE state, the OLT Updates the transmitter RAM with the discovery GATE frame in order to send it to the ONU side in Transmitting_Request1 state and waits in this state till receiving REGISTER_REQ packet and then waits to send a transmission guard in linewait2 state.
3. Again, the OLT loads and transmits the REGISTER packet in Load_REGISTER and Transmitting_Request2 states.
4. The OLT in MACR_REGISTER_ACK state requests for acknowledgement from the ONU by sending MACR_REGISTER_ACK ONU upper layer signal and then waits in this state till receives the REGISTER_ACK packet.
5. Finally, if the received REGISTER_ACK package confirms the registration then the OLT led light is turned on.

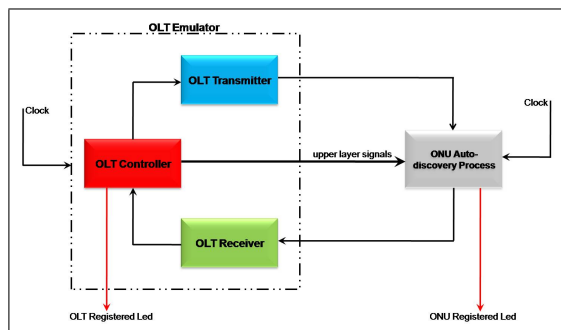


Fig. 6. Design Emulation

C. Hardware Implementation

Fig. 8 shows the test board that has been created to implement the design. The board contains the following peripherals: FPGA ALTERA Cyclone III (EP3C25Q240C8), Ethernet connector, fiber optics connector, clock generator of 50 MHz and power supply provides 1.2volt, 3.3volt and 1.8volt. Table I provides the synthesis design results using *Quartus II* [13] and Synopsys Power Compiler [14] tools for the whole design including the OLT emulator and the ONU

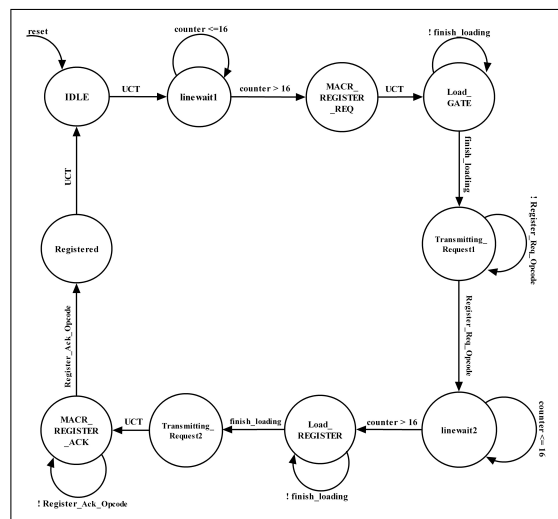


Fig. 7. Emulator FSM

discovery process. As shown in the table, the ONU discovery process and OLT model consumed two memory blocks for 512 byte each in the transmitter block, and total logic elements of 1,199 logic elements. There is always a tread off between the memory blocks and the logic elements so that if we use less memory blocks, the logic elements used are increased. The rest table data provides the consumed logic elements and memory block in more details. Finally, the power consumption of the proposed design has been measured using Synopsys Power Compiler.

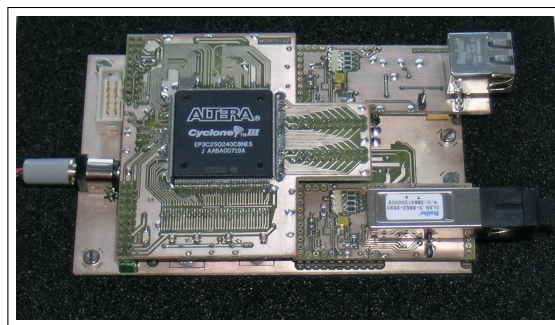


Fig. 8. Hardware Board

V. CONCLUSION

In this article, we have provided a MAC design of auto-discovery process introduced in MPCP for EPON system. The proposed design has been created in RTL form, written in VHDL language and following IEEE 802.3ah standard. the correctness of the design has been performed using three phases, simulation, emulation and hardware implementation. Finally, we have provided some synthesis and power consumption results.

ACKNOWLEDGMENT

The authors would like to thank Dr. M. Boubekeur and the conference reviewers for their useful review comments.

TABLE I
SYNTHESIS RESULT

Resources	Usage
ONU_Discovery_Process [Transmitter]	512 byte
OLT_Model [Transmitter]	512 byte
Estimated total logic elements	1,199
Logic element usage by number of LUT inputs	
2 input functions	211
4 input functions	310
3 input functions	678
Logic elements by mode	
Arithmetic mode	181
Normal mode	1018
Total memory bits	1024
Total combinational functions	1018
Total registers	391
Dedicated logic registers	391
Power consumption	1.3360 mW

REFERENCES

- [1] M. Hajduczenia, H.da Silva and P. Monteiro, "EPON versus APON and GPON: a detailed performance comparison," *Journal of Optical Networking*, 2006.
- [2] W. Jianli, "FTTH in China," Deputy CTO, Fiberhome Technologies Group, Wuhan, China, Dec 2005.
- [3] Fransman, M. (2007), "Why Japan and Korea lead the USA and Europe in broadband," *Int. J. Technological Learning, Innovation and Development*, Vol. 1, No. 1, pp.112-124.
- [4] G. Kramer, "Ethernet Passive Optical Networks," McGraw-Hill Professional, ISBN: 0071445625, Mar 2005.
- [5] IEC, "Ethernet Passive Optical Networks," IEC on radio Tele-Semana, commentary on NXTComm, 2007.
- [6] T. Fan and H.T. Mouftah, "A QoS-based Dynamic Bandwidth Allocation Algorithm For EPONs," *23rd Biennial Symposium on 1929*, 2006.
- [7] IEEE , "Media Access Control Parameters, Physical Layers and Management Parameters for subscriber access networks," IEEE Draft P802.3ah/D1.0TM (Amendment to IEEE Std 802.3-2002TM), Aug 2002.
- [8] W. Chen, Montuno D.Y., "Felske K., Enhanced EPON auto-discovery for fast network and service recovery," *Canadian Conference of Computer Engineering*, 2004.
- [9] A. Mady, A. Tonini and D. Finardi, "A VHDL Implementation of ONU Auto-discovery Process of the IEEE 802.3ah MPCP Protocol," Master Thesis, ALaRI, Lugano, Switzerland, July 2008.
- [10] G. Campobello, G. Patane and G. Russo, "Parallel CRC Realization," Dept. of Phys., Messina Univ., Italy, *Computers, IEEE Transactions*, Oct 2003.
- [11] http://www.altera.com/products/ip/processors/nios2/tools/ni2-development_tools.html
- [12] Model Technology, a Mentor Graphics Corporation company, "ModelSim SE Tutorial," Version 5.7f, Apr. 2003.
- [13] ALTERA CORPORATION, "Introduction to the Quartus II Software," Version 8.0, 2007.
- [14] A. Mady, A. Tonini and D. Finardi, "Design Space Exploration of PISA Architecture For ONU Auto-discovery Process," *6th International Conference on Electrical Engineering (ICEENG 2008)*, Cairo, Egypt, Page 21, 27-29 May 2008.