IMPLEMENTATION OF A TESTBED FOR EVALUATING ENERGY EFFICIENCY IN PASSIVE OPTICAL NETWORK.

By

NEELAKANDAN MANIHATTY BOJAN

A THESIS

Submitted in partial fulfillment of the requirements for the degree

MASTER OF SCIENCE IN PHOTONIC NETWORKS

OPTICAL NETWORKS GROUP

SCUOLA SUPERIORE DI STUDI UNIVERSITARI

DI PERFEZIONAMENTO SANT'ANNA

PISA

ITALY

AUGUST 2012

Approved by:

Dr. LUCA VALCARENGHI

Abstract

In recent years, there has been an increasing trend to save energy in communication networks, not just because of environmental awareness but also because of the rise in operating expense for the telecom operators, network and internet service providers. This has triggered a lot of activity in standardization bodies, industry and academia to come up with energy-efficient solutions targeting access networks that constitute a large proportion of the network energy consumption. One of the main research areas in lowering energy consumption is the implementation of sleep mode techniques in the elements of the access network. Such techniques power off the network element or the system based on the control logic implemented within/outside the system or based on the network load.

This thesis implements an energy efficient Optical Network Unit (ECONU) on an Altera Stratix 4GT Field Programmable Gate Array (FPGA). The ECONU architecture with its new frame format design is presented to increase the energy saving in point to point Ethernet passive optical networks (EPONs). A Finite State Machine (FSM) implementation based on the sleep request encoded in the frames for switching the ONU ”on” and ”off” is proposed. The performance of the ECONU prototype is evaluated through hardware testbed and the obtained results confirm the energy savings.

Table of Contents

List of Figures

List of Tables

List of Acronyms

ALM	Adaptive Logic Module
ALR	Adaptive Link Rate
APD	Avalanche Photodiode
ASIC	Application Specific Integrated Circuit
Avalon-MM	Avalon Memory-Mapped
Avalon-ST	Avalon Streaming
BBCoC	Code of Conduct on Energy Consumption of Broadband Equipments
CLB	Configurable Logic Block
CM-CDR	Continuous Mode Clock and Data Recovery
CO	Central Office
CPE	Customer Premises Equipment
CPLD	Complex Programmable Logic Devices
CRC	Cyclic Redundancy Check
CSR	Control and Status Register
DBA	Dynamic Bandwidth Allocation
DDR	Double Data Rate
D-MUX	Demultiplexer/De-serializer

DRC	Design Rule Check
DSL	Digital Subscriber Line
DSLAM	Digital Subscriber Line Access Multiplexer
DSP	Digital Signal Processing
DUT	Design Under Test
ECONU	Energy Efficient Optical Network Unit
EDA	Electronic Design Automation
EOP	End of Packet
EPON	Ethernet Passive Optical Network
ERC	Electronic Rule Check
EU	European Union
FEC	Forward Error Correction
FIFO	First In First Out
FPGA	Field Programmable Gate Array
FSM	Finite State Machine
FTTCab	Fiber-To-The Cabinet
FTTEx	Fiber To The Exchange
FTTH	Fiber To The Home
Gb/s	Giga bits per second
GMII	Gigabit Media Independent Interface
GPON	Gigabit Passive Optical Network
GTC PON	GPON Transmission Convergence
HDL	Hardware Description Language

HFC	Hybrid Fiber Coaxial
IEEE	Institute of Electrical and Electronics Engineers
IP	Intellectual Property
ITU-T	International Telecommunication Union- Telecom Standards
JTAG	Joint Test Action Group
LA	Limiting Amplifier
LAB	Logic Array Block
LFSR	Linear Feedback Shift Register
LPI	Low Power Idle
LUT	Look Up Table
LVDS	Low Voltage Differential Signalling
LVS	Layout Versus Schematic
MAC	Medium Access Control
MII	Media Independent Interface
MPCP	Multi-Point Control Protocol
MPMC	Multi Point Media Access Control
NRZ	Non-Return to Zero
OCDMA	Optical-Code Division Multiple Access
OLT	Optical Line Terminal
OMCC	ONT Management Communication Channel
OMCI	ONU Management and Control Interface
ONU	Optical Network Unit
OOFDMA	Optical-Orthogonal Frequency Division Multiple Access

OPEX	Operating Expense
OSI	Open System Interconnection
P2MP	Point To Multipoint
PCS	Physical Coding Sublayer
PHY	Physical Layer
PI	Programmable Interconnect
PLL	Phase Locked Loop
PLOAM	Physical Layer Operation, Administration, and Management
PMA	Physical Medium Attachment
PMD	Physical Medium Dependent
PON	Passive Optical Network
PSM	Power Save Mode
QOS	Quality Of Service
RAM	Random Access Memory
RFPON	Radio Frequency Passive Optical Network
RN	Remote Node
ROM	Read Only Memory
RS	Reed-Solomon
RTD	Round Trip Delay
RTL	Register Transfer Logic
RTT	Round Trip Time
SDR	Single Data Rate
SERDES	Serializer/Deserializer

SFP	small form-factor pluggable
SLD	System Level Debug
SOP	Start of Packet
SOPC	System On a Programmable Chip
TC	Transmission Convergence
TCL	Tool Control Language
TDMA	Time Division Multiple Access
TIA	Transimpedance Amplifier
UNI	User to Network Interface
VHDL	VHSIC hardware description language
VLSI	Very Large Scale Integration
WDM	Wavelength Division Multiplexing
XGMII	10 Gigabit Media Independent Interface
XG-PON	10 Gigabit Passive Optical Network

Acknowledgments

I owe my deepest gratitude to my supervisor Dr. Luca Valcarenghi for his guidance and continuous support throughout my thesis. I would also like to thank Mr. Dzung Pham Van for his invaluable help during the implementation period. It is a great pleasure to acknowledge here the useful discussions we had during our weekly meetings.

I am grateful to MAPNET for providing me with this opportunity to take up this incredible masters program.

I would like to thank my colleagues at TECIP for their amicable support, both on and off premises.

A special thanks to all my friends who had always supported me.

As always, I am greatly indebted to my family for their continuous support and love.

Lastly I would like to thank the Almighty for having bestowed the blessings on me to complete this thesis work successfully.

Chapter 1

Introduction

The ever increasing demand for internet and broadband service coupled with the growing traffic demand, has spurred a lot of research activities in evaluating ways to reduce the energy consumption of networks and exploring opportunities to improve the network's energy efficiency. Figure 1.1 shows the schematic view of the universal broadband TC network considered in [1].

Telecom providers are seeking to reduce communication network energy consumption not only to lower greenhouse gas emissions but also to decrease their operating expenses (OPEX). Figure 1.1 shows the operator's part of network segment in the broadband telecommunication network considered in [1]. Apart from this, there is also the home network (not shown here). The operator's network is divided into fixed access network, aggregation network, backbone network, mobile radio network and the data centers.

An overall forecast of a universal network operator's energy consumption is made in [1] considering both the operator's network sections and home networks. Energy efficiency calculations are also made in [1] considering network elements with state of the art technology, and no technology related advances were considered during this point.

The fixed access network partition is composed of fiber-to-the-exchange (FTTEx) and fiber-to-the cabinet (FTTCab) approaches, where either the asymmetric digital subscriber line (DSL) or digital subscriber line access multiplexer (DSLAM) is located in the central office or the very-high speed digital subscriber line DSLAM resides in the street cabinet, respectively. For the forthcoming years, demand for fiber-to-the-home (FTTH) access networks is also evaluated by making use of passive optical network (PON) technology. Access networks (fixed and mobile) are recognized as the major contributors to current communication networks energy consumption. This is mainly due to the high number of involved elements (i.e., the customer premises equipments - CPEs). Access networks based on Passive

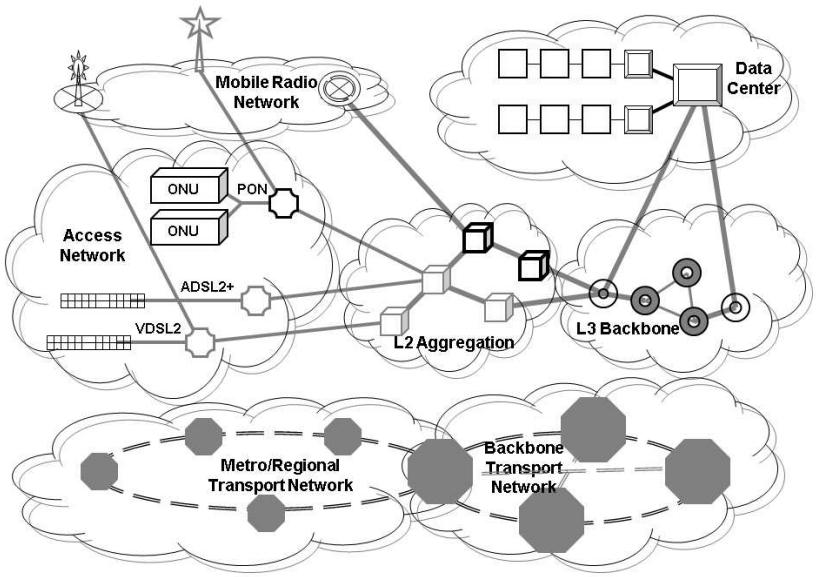

Figure 1.1: Schematic view of Operator's Broadband Telecommunication network.

Optical Networks (PON) are expected to greatly reduce the access network energy consumption with respect to DSL based solutions. However, fixed optical access networks are forecast to be the largest contributors to fixed optical communications networks energy consumption for the next ten years [2].

In Time Division Multiple Access (TDMA) PON, Optical Network Units (ONUs) (i.e., PON CPEs) are the devices that have the highest energy consumption per bit. Indeed, they are numerous and often idle, because of the current low utilization of PON capacity and the shared medium architecture. In [2], it is shown that ONUs contribute for over 65% to the total PON power consumption. ONUs are therefore the major target for energy saving in PONs and their energy efficiency can be significantly improved by turning off the idle ONUs.

Several papers have shown the advantages and the drawbacks of turning ONUs "on" and "off" [3, 4, 5]. Some experimental evaluations have been formerly presented in [6] and [7]. In [7] the cyclic sleep technique with adaptable sleep period is implemented by utilizing an small form-factor pluggable (SFP) board transceiver at 1.25 Gb/s. The transceiver is connected to differential serial interface of an Altera Transceiver Signal Integrity Development Kit, Stratix IV GT Edition, capable of supporting a transmission rate of 1.25 Gb/s.

The implemented ONU shows to be capable of effectively receiving SLEEP REQUEST and replying with NACK or ACK. In [6] a sleep control (i.e., Sleep and Periodic Wake-up) in conjunction with adaptive link rate

(ALR) control for EPONs is implemented. Sleep control switches the ONU mode, i.e. active or sleep mode, depending on the presence or absence of traffic while the latter switches between two downlink rates, i.e. 1G and 10G, depending on the quantity of traffic. The goal of such combination is to overcome the disadvantages of the individual sleep and ALR control functions.

1.1 Passive Optical Network Overview

A passive optical network has a single, shared optical fiber that uses a passive optical splitter to divide the signal towards individual subscribers. PON is called passive because other than at the central office, there is no active element within the access network. Due to the lack of active units in the light path, the architecture of PON is simple, cost effective and offers high bandwidth. This is not possible to achieve by other access methods.

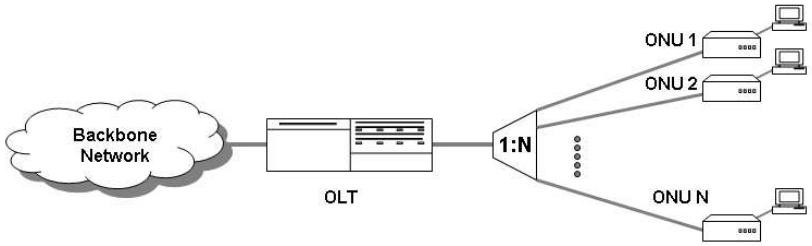

Figure 1.2: Typical PON Architecture.

The optical line terminal (OLT) is the main network element with PON line card (aggregation switch), usually located in the Central Office (CO). OLT is the gateway between access and metro network. Optical splitter is a passive device which splits the optical power at the input, evenly between outputs. So it has single input and multiple outputs. Splitters can be placed anywhere in between CO and subscriber premises. It is used to connect an optical port of OLT with multiple subscribers. Signals can travels from OLT to ONU (downstream communication) and can also travel from the ONU to OLT (upstream communication). ONUs serve as an interface to the network and are deployed at customer premises. It provides several interfaces for accessing triple play services and in the upper side it connects with the OLT via optical splitter.

1.2 Types of PON

There are different types of PON technologies; the following provides a brief overview of the different types of PON and their functionalities:

The exponentially growing demand for bandwidth-intensive multimedia services in recent years has triggered research in increasing the transmission capacity of the access network [8]. In particular optical orthogonal frequency division multiple access (OOFDMA) PONs are an emerging topic in this field. In OOFDMA PONs, an overall channel bandwidth (for upstream and downstream transmissions) can be divided into a large number of orthogonal subcarriers of different frequencies, one or more of which can be easily assigned statistically to a specific ONU, and a single subcarrier can also be further shared among different low- speed applications via TDM. These PONs offer a lot of advantages like: high bandwidth for increased number of subscribers in a cost effective way, system flexibility, backward compatibility and dynamic bandwidth allocation (DBA) based on digital signal processing(DSP).

Optical code division multiple access (OCDMA) technologies for usage in local area networks are explored as they can simplify the network control and management as compared to (TDM or WDM) technologies. The large bandwidth available in the optical networks can be easily and flexibly allocated using the OCDMA PONs without relying on other nodes [9]. 1 dimensional (1-D) and 2 dimensional (2-D) OCDMA codes have been proposed. The orthogonal codes are chosen to have the most favorable autocorrelation and cross-correlation characteristics to over the channel cross talk (multi-user interference). Such PONs can provide asynchronous communication and security against unauthorized users. Such types of PON are expensive (due to the encoders and decoders used) and the number of users are limited (by the noise and interference).

Radio-frequency PON (RFPON) is a PON that is capable of supporting RF video broadcasting signals over optical fibers [10]. During network expansion, new RF components with additional bandwidth are deployed and this some times require RF electronics changes. This can result in lower signal quality and higher operating and maintenance costs in hybrid fiber coaxial (HFC) networks. RF PONs are backward compatible with HFC networks and are cost effective.

The TDM PON is an access network based on optical fiber. It is designed to provide virtually unlimited bandwidth to the subscriber. In case of TDM PON, the same signal from the OLT is broadcast to different ONUs by the power splitter. Signals for different ONUs are multiplexed in the time domain. The downstream and upstream signals are carried over the same fiber. In the downstream direction the signal sent by the OLT arrives at the splitter's input and later the same signal reaches every ONU. The transmitting method in downstream direction is broadcasting. In the upstream direction, the signals from different ONUs arrive at inputs of the splitter. Although the signals can not reach different ONUs, as they get mixed with each other (when passing through splitter) and the superposition of all signals is received at the OLT. Hence in the upstream direction

the TDM method is used to avoid the interference of signals from different ONUs.
An alternative to expand PON capabilities is wavelength division multiplexing PON (WDM PON). A WDM PON is characterized by a WDM coupler, which replaces the power splitter at the remote terminal. WDM PONs are the next generation in development of access networks. Ultimately, they can offer the largest bandwidth at the lowest cost. In principle, the architecture of WDM PON is similar to the architecture of the PON. The main difference is that multiple wavelengths operate on single fiber and ONUs operate on different wavelengths enabling either more bandwidth per ONU or more ONUs for every distribution fiber. WDM-PON provides better privacy and better scalability. However, WDM devices are expensive, which makes WDM-PONs economically less attractive at this moment.

Super PON was proposed to achieve better economy by increasing the reach of PON systems beyond 20 km and supporting higher splitting ratios of 1:64 or even 1:128. However, super PON faces many challenges such as, economical fiber amplifiers are not available at the 1.3/1.49 μ m upstream and downstream wavelengths, dispersion effect around 1.5 μ m wavelength is non negligible, round trip time (RTT) increases between the OLT and ONU. These challenges of super PON in both the physical and medium access control (MAC) layers may demand advanced technologies which eventually outweigh the economic advantages achieved through longer transmission distances and bigger share groups.

In Ethernet passive optical network (EPON), the MPCP (multi point control protocol) is mandatory to perform bandwidth assignment for sustaining communication between OLT and ONUs and auto-discovery process for newly activated ONUs. Meanwhile, in gigabit passive optical network (GPON), the GTC (GPON transmission convergence layer) does similar tasks. In the default state of PON, an ONU is not allowed to exchange data. An ONU cannot transmit data unless it is granted by the OLT. This grant, however, will never arrive because the OLT doesn't know and cannot know that a new ONU has joined since the ONU has to remain silent. To resolve this sort of situation, in EPON an auto discovery mode is defined in MPCP protocol. The auto discovery mechanism is used to detect newly connected ONUs and learn the round-trip delays (RTDs) and MAC addresses of these ONUs. The RTDs between the OLT and ONUs can help OLT appropriately offset the upstream time slots granted to ONUs to avoid upstream collision. Similarly, ranging procedure is defined in GPON that is characteristically divided into two phases. In the first phase, registration of the serial number for unregistered ONU and allocation of ONU-ID for the ONU are executed. The serial number is an ONU-specific ID and must be universally unique. The ONU-ID on the other hand is used for controlling, monitoring, and testing the ONU. So it can be unique only within its

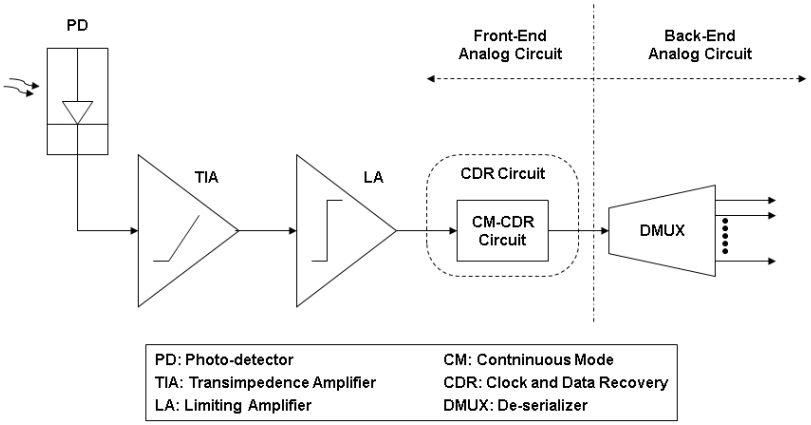

Figure 1.3: ONU Receiver Architecture.

own PON. Next, the round trip delay (RTD) is measured for each newly registered ONU. This phase is also applied to ONUs which lost their signals during communication.

1.3 ONU Receiver Architecture

Generally the sleep technique implementations for lower energy consumption target ONUs because of their large energy consumption (over 65% as described in [2]) and low utilization. By understanding the ONU receiver architecture, it provides a clear idea on how the sleep technique can be implemented and which components in the ONU are to be switched off during the sleep period. And the overall power of the access segment can be drastically reduced by optimizing the energy consumption of the devices at the customer premises.

The amount of energy savings achieved depends on which components at the ONU are turned off. Therefore, it is necessary to understand the ONU receiver architecture.

Figure 1.3 describes the ONU receiver architecture and its workflow operation among the components inside. The received optical signal is first converted into electrical signal (i.e., current) by an avalanche photodiode (APD). Two stages of electrical amplifiers, including transimpedance amplifier (TIA) and limiting amplifier (LA), that amplify the electrical signal before sending it into the continuous mode clock and data recovery (CM-CDR) circuit. The CM-CDR forwards the recovered data and clock to the de-serializer (DMUX). The DMUX then outputs parallelized data to lower speed, back-end digital circuit for further processing. The back-end digital circuit comprises primarily memory and various digital data processing blocks.

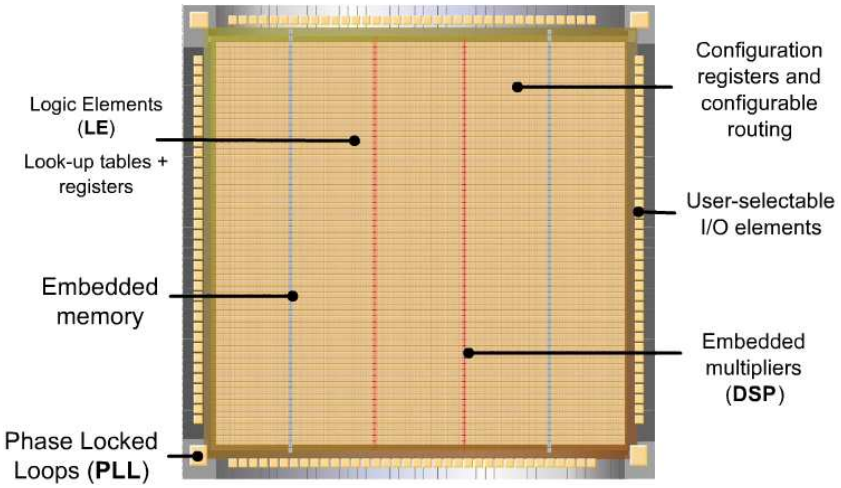

Figure 1.4: FPGA architecture.

1.4 Field Programmable Gate Array

At the highest level, Field Programmable Gate Arrays (FPGAs) are reprogrammable silicon chips. Using prebuilt configurable logic blocks (CLBs) and Programmable interconnects (PI, as programmable routing resources). Chips can be configured to implement custom hardware functionality without ever having to pick up a breadboard or soldering iron. Digital computing tasks can be developed in software and compiled down to a configuration file or bit stream that contains information on how the components should be wired together. In addition, FPGAs are completely reconfigurable and instantly take on a brand new ”personality” when recompiled with a different configuration of circuitry. The rise of high-level design tools, had made FPGA programming more simpler, with new technologies that convert graphical block diagrams or even C code into digital hardware circuitry. The FPGAs are programmed and configured using hardware description languages (HDL) like Verilog and VHSIC hardware description language (VHDL), similar to that used for an application-specific integrated circuit (ASIC).

1.4.1 Logic Cell

The FPGAs are composed of thousands of logic cells (also called as CLB’s). A ”logic-cell” in turn is composed of three basic components: a small Lookup Table (LUT), a D-Flip-flop and a 2-to-1 Multiplexer.

As shown in the Figure 1.5, the Multiplexer may be used to bypass the flip-flop. Flip-flops are binary shift registers, which are used to synchronize logic and save logical states between clock cycles. On every clock

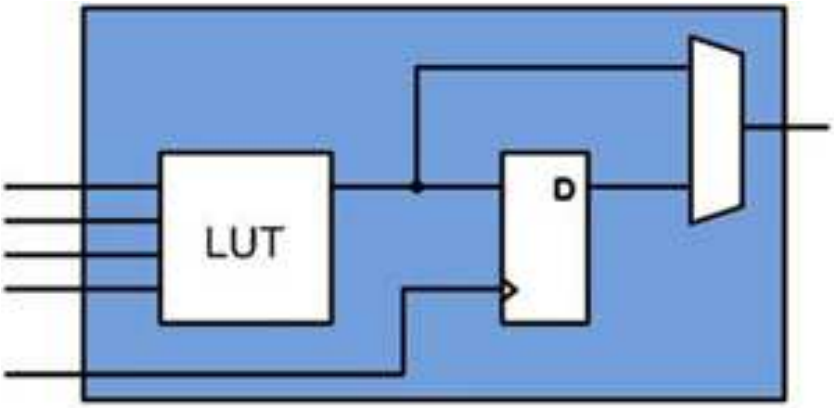

Figure 1.5: Logic Cell.

edge a flip-flop latches a 1 or 0 value on its input and holds that value constant until the next clock edge. A look up table (LUT) may be regarded as a small random access memory (RAM) which is capable of implementing any logic function.

1.4.2 Programmable Interconnect

Each logic-cell when taken individually may be of little use. However, a large number of logic-cells when connected together may be employed for implementation of complex logical functions. Programmable interconnect routes the signals between CLBs and to and from I/Os. Routing is also designed to interconnect between CLBs to fast horizontal and vertical long lines spanning the device to global low skew routing for clocking and other global signals.

1.4.3 Embedded Memory

The IO Blocks, which are implemented at the boundary of the FPGA chip, provide connection between internal logic and the package pins for interfacing the internal logic cells with the components connected externally with the FPGA chip [11].

1.4.4 IO Blocks

Embedded Block RAM memory is available in most FPGAs, which allows for on-chip memory in your design. These allow for on-chip memory for your design.

1.4.5 Phase-locked Loop

Phase-locked loops (PLLs) are used for clock management. They provide precision clock synthesis combined with jitter reduction and filtering.

This thesis describes the implementation of an Energy Efficient ONU (ECONU) prototype for EPON in Altera FPGA based on the finite state machine (FSM) described in [12] that slightly simplifies the one defined in [13]. It considers downstream traffic only. A packet format is defined to include within the Ethernet frame payload control messages to switch the ONU between ”on” and ”off” states. Results show that the ECONU successfully switches between the FSM states according to sleep requests received from downstream packets and on predefined timers.

1.5 Thesis Organization

This thesis contains six chapters each covering a different aspect of the thesis work.

- Chapter 1 introduces the topic of the thesis with a brief introduction of PONs and FPGA. How they are used to achieve our motivations and goals.
- Chapter 2 provides an overview on the energy consumption in PON's, followed by the different solutions proposed by industry and academia - and their classification based on the OSI layer
- Chapter 3 presents the architecture of the testbed developed during this thesis. The chapter explains the various modules constituting the design, their functionality and their integration.
- Chapter 4 shows the testbed implementation. Starting with the illustration of the bird's eye view of the entire design followed with an overview of the FPGA platform used for the design implementation. Explanation of the implementation of exponential inter-arrival time, packet extractor module, Sleep Finite State Machine (FSM) from the simulation stage till the implementation stage is provided. This section also includes the tools used for the implementation and debugging, and finally the resource utilization for the design is provided.
- Chapter 5 presents the results of the implementation in two modes (internal loop back and local loop back). Followed by some results showing the energy efficiency statistics and performance statistics for the proposed technique.

- Chapter 6 closes the report presenting the conclusions and some suggestions about possible future work.

Chapter 2

State of the Art

This chapter provides an overview of the current efforts in reducing energy consumption in passive optical access networks (PONs) by standard bodies, industry and academia.

2.1 Power consumption in PONs

Today's widespread PON standards are the Gigabit capable passive optical network (GPON ITU-T Rec. G.984.x) with capacity of 1 Gb/s, the 10-Gigabit-capable passive optical network (XG-PON ITU-T Rec. G.987.x) with capacity of 10 Gb/s, the Ethernet Passive Optical Network (EPON IEEE 802.3-2008 section 5) with capacity of 1 Gb/s, and the 10 Gb/s Ethernet Passive Optical Networks (10G-EPON IEEE 802.3av-2009) with capacity of 10 Gb/s.

ONU power consumption data can be collected from standards [14], component data sheets and research papers. Table 2.1 summarizes the power consumption for various PON standards collected from component data sheets from various vendors. Table 2.1 clearly shows that clock and data recovery (CDR) and serializer/deserializer (SERDES) consume the highest power in both EPON and GPON - more than 80% of the power of the entire ONU receiver. Although data are not available for 10G-EPON and XG-PON, a similar behavior can be expected.

However, the SERDES is commonly shared by the ONU receiver and transmitter. Moreover, the collected data confirms the details presented in [15]: between 60%-70% of the overall ONU power consumption is due to the PON transceiver and back-end electronic circuit.

Table 2.1: Power consumption [mW] of discrete components in ONU.

Receiver Front-End Component	EPON		GPON		10G-EPON		XG-PON	
	Avg	Range	Avg	Range	Avg	Range	Avg	Range
APD	2.6	2-3.75	26	2-3.75	2.6	2-3.75	2.05	0.5-3.75
TIA	83.4	56-112	83.4	56-112	123	105-160	123	105-160
LA	121	89-140	126	100-165	145	110-165	154	125-180
CDR	545	540-580	520	260-790	356	NA	356	NA
SERDES	550	530-660	560	530-660	NA		NA	
Total Receiver Front-End	1302		1292		NA		NA	
Transceiver	1350	1100-2500	1500	1040-2250	1600	1300-2300	1800	1800-1800
Back-End Circuit	2700		3150		5850		6750	
Whole ONU (Services)	6000 (Ethernet Data Port + IPTV)		7000 (Tripleplay + Multicast Video)		13000 (prediction)		15000 (PoE on Gigabit Port)	

2.2 Solutions for saving energy in PONs

The classification of solutions for saving energy is based on the open system interconnection (OSI) model. So in order to understand the classification better, a brief introduction to the the layering architecture of EPON is provided. Ethernet covers the physical layer and data link layer of the OSI reference model. Figure 2.1 shows a comparison of the layering model of the point-to-multipoint (P2MP) EPON architecture.

The physical layer (PHY) is connected to the data link layer using the media-independent interface (MII) or gigabit media-independent interface (GMII).

The optional MAC sub-layer is a mandatory multipoint media access control (MPMC) layer in EPON. The MPMC layer runs the multipoint control protocol (MPCP) to coordinate the access to the shared PON medium among EPON ONUs. Although the OLT and ONU stacks look nearly identical, the MPCP entity in an OLT functions as the master and the MPCP entity in ONU as the slave. The reconciliation sub-layer has been extended in EPON to handle a P2P emulation function so that IEEE 802.1-based bridging protocols will continue to function with EPON.

The PMD layer specifies the physical characteristics of the optical transceivers. Ethernet has the tradition of adopting mature low-cost designs to promote mass deployments. This philosophy has been the key to the tremendous commercial successes of Ethernet.

The physical coding sublayer (PCS) is the layer that deals with line-

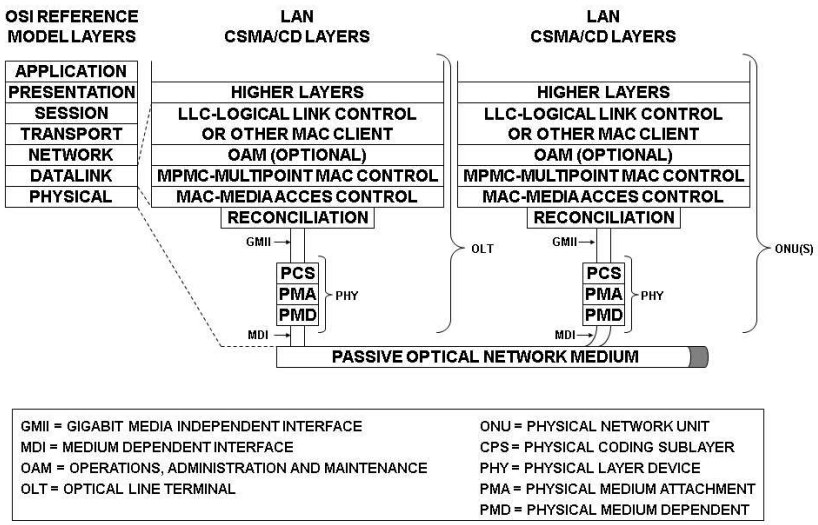

Figure 2.1: Point-to-multipoint (P2MP) EPON layering architecture.

coding in Ethernet physical layer devices. The use of forward error correction (FEC) is optional in EPON. The IEEE 802.3ah standard defines reed-solomon (RS) block codes in the EPON PCS layer. This is the same code used in G-PON. EPON transmits data as native Ethernet frames in the PON section. Ethernet frames are variable size frames.

The classification of the solutions for saving energy in PONs is based on the layer they address in the OSI model: physical layer, data link layer, and hybrid. Figure 2.2 provides an overview of the classification.

2.2.1 Physical Layer Solutions

Physical layer solutions aim at reducing PON energy consumption by targeting the physical layer of PON architecture without modifying other higher layers. They are divided into: device-oriented and service-oriented.

Device-oriented solutions aims to reduce the energy consumption of the physical devices. Service oriented solutions aim at improving the performance of the services provided by the physical layer to enable upper layer solutions (e.g., sleep mode).

The implementation of device-oriented physical layer solutions is promoted by standard bodies and European Union (EU) sponsored research centers. One such example is the ”Code of Conduct on Energy Consumption of Broadband Equipments” (BBCoC) published by the institute for Energy of the Joint Research Center of the European Commission Directorate General [16]. This document provides guidelines on the power consumption targets for both CPEs and network equipments for the forthcoming years.

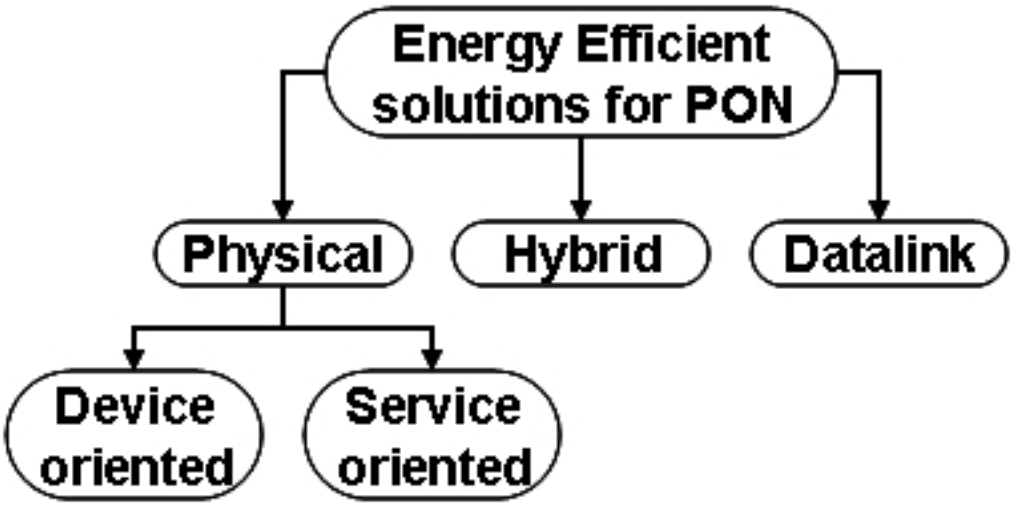

Figure 2.2: Energy efficient solutions for PON based on OSI layer.

Table 2.2: Power consumption reduction by means of integration.

Variables	EPON	GPON	10G-EPON	XG-PON
	Power [mW]			
CDR	545	520	356	356
CDR+LA	410	410	350	NA
CDR+DMUX	910	790	NA	NA
LA+CDR+DMUX	610	610	NA	NA

One of the most important physical layer solutions to reduce energy consumption is device level integration of the components. Table 2.2 shows how electronic integration leads to decrease in component power consumption. For example, an integrated solution containing CDR, LA, and SERDES consumes only half the power consumption when using discrete components in EPON.

IEEE 802.3az EEE task force proposed adaptive link rate (ALR) and is discussed in [17]. ALR saves power by adapting the interface line rate to the network traffic. ALR is based on the fact that transmission at low rates requires less power. However, implementation of ALR is complex.

New devices have been proposed for extending the reach of next generation PONs. Such devices can also be exploited to decrease the transmitted power along the fiber in current PON. One such example is the

Table 2.3: Clock recovery time reduction by means on integration.

	$T_{rec}[ms]$							
	EPON		GPON		10G-EPON		XG-PON	
	Avg	Range	Avg	Range	Avg	Range	Avg	Range
Discrete components								
CDR	2.05	0.6-14	2.2	0.6-14	1.1	0.2-2	1.1	0.2-2
Integrated components								
CDR+LA	1		1		2		NA	
CDR+DMUX	3		2		NA		NA	
LA+CDR+DMUX	0.6		0.6		NA		NA	

quasi-passive optical switch described in [18]. This switch located at the remote node (RN) allows to reduce the RN splitting loss and thus reduces the power transmitted by the OLT. But the drawback of this solution the decrease in optical power budget. Hence the energy consumption cannot be lowered beyond a certain point.

Electronic integration can be exploited to achieve low power consumption in device level. Table **??** shows that the integration of LA, CDR and SERDES can shorten clock recovery time T_{rec} to almost one fourth than the one of a discrete CDR.

New solutions based on changing the modulation format to allow faster clock recovery has also been proposed in [19]. By using Manchester modulation format instead of the standard Non-Return to Zero (NRZ) format enables faster clock recovery. Another service-oriented physical layer solution is optically powered fiber networks [20]. In which remote devices (e.g., sensors) accumulate and store energy within their idle time, and then perform their functions for a short time and send the acquired information. Similar scenarios can be exploited for remotely triggering of control circuit that wakes up ONUs after a sleep period.

2.2.2 Datalink Layer Solutions

Data link layer solutions target the data link layer of the IEEE 802.3 architecture (i.e., the MAC layer) or the Transmission Convergence (TC) layer in GPON (ITU-T Rec. G.984.3) and XG-PON (ITU-T Rec. G.987.3). The main aim of the is to reduce energy consumption by switching the network ele-

ments to sleep mode.

ITU-T G.Sup45 [14] proposes three types of power conservation techniques, namely *power shedding*, *dozing*, and *sleeping* (which is further divided into *deep* and *fast sleep*). These methods mainly differ in the way the ONU transmitter and receiver behave.

In *Power shedding* all non-essential functions and services (e.g., interfaces)are switched off but optical link is operational (working). In *Dozing*, the ONU receiver is always on but the ONU transmitter may be switches off/on based on availability of upstream traffic. *Deep sleep* turns off the whole ONU transceiver and all the ONU functions and services, except activity detection.

In *deep sleep*, the ONU loses the incoming downstream traffic and may wake up only when the ONU is switched on local stimulus or at the expiration of a locally timer. *Fast sleep* is based on sleep cycles that alternate sleep periods, when the whole ONU transceiver is turned off together with all the non-essential functions, and active periods, when the optical transceiver and the necessary supporting functions are turned on. The OLT learns the capabilities of the ONU to support a specific power saving mode through extensions of the Management and Control Interface (OMCI) management channel and it can also negotiate with the ONU on which modes to select. Transitions to and from a particular power save mode of the ONU can be supported by different signaling messages, such as PLOAM power save mode (PSM) message or by the OMCI framework as proposed in ITU-T Rec.G.988.

[15] shows a combination of sleeping and power shedding. It has the potential of reaching power savings upto 80%. However, all the packets arriving to the ONU user to network interface (UNI) during the power shedding are lost. So this forces us to use power shedding only when the ONU is idle for a long time. Moreover, G.Sup45 also proposes a new solution that enables operators to turnoff ONU services during times of the day when they are not needed.

Although IEEE 802.3az includes the support of Low Power Idle (LPI) mode for point to point links, EPON (IEEE 802.3- 2008) and 10G-EPON (IEEE 802.3av-2009) do not contain any specification of data link layer solutions for energy consumption reduction in PON. Methods for enhancing MPCP with sleep mode control messages have been proposed, but not standardized yet [21].

Interest among several vendors seeking data link layer solutions for reducing ONU energy consumption has recently increased. This is observed by the large number of patent applications that have been recently submitted. For example, in [22] a cyclic sleep mode is proposed to be implemented in IEEE 802.3 style networks. Both OLT driven and ONU driven methods are considered, but the final decision of switching to sleep mode

is taken by the OLT. Extension to the EPON MPCP protocols to implement the handshake between the OLT and the ONU to switch to sleep mode are also proposed that introduce new messages such as SLEEP, SLEEP_REQ, CHANGE_SLEEP_MODE. Many vendors are also advertising the introduction of power saving modes into their devices [23]. In this case, the ONU supports power saving modes for power consumption reduction during power outages and standby periods.

In [24] two data link layer solutions are presented for reducing energy consumption of EPON and 10GEPON. The less aggressive solution defines sleep-mode control and sleep aware traffic scheduling to allow ONUs to sleep over one DBA cycle. The more aggressive solution allows ONU to sleep within one DBA cycle. In both solutions, four ONU power levels are defined based on the possibility of selectively turning off the ONU transmitter or the receiver. If both transmitter and receiver are off, even the ONU MAC and SERDES are turned off.

In [4] the performance evaluation of two power saving methods standardized in XG-PON (i.e., ITU-T Rec. G.987.x), namely cyclic sleep (fast sleep) and doze mode (as defined in ITU-T G.Sup45) is performed through simulations. One of the key features in XG-PON power saving modes is the utilization of timers to regulate the duration of sleep periods. This solution simplifies the coordination between ONU and OLT because timers can be updated yet but not necessarily for each sleep period. Another feature is the utilization of the sleep (or doze) aware state in which ONU periodically (i.e., based on timers) wakes up to collect potential wake-up messages from the OLT. In addition to the timers, the XG-PON power saving solution utilizes scheduling message to schedule ONU into sleep or doze mode. Simulations performed in [4] show that, in cyclic sleep, a larger sleep period (determined by the value of the timer T_{sleep}) allows for larger power consumption reduction. However, if the traffic is low and the T_{sleep} is high, the delay of downstream message is very high (close to 90ms). On the contrary, upstream traffic delay is not affected by the sleep duration because ONU initiated wake-up does not depend on T_{sleep}. In doze mode, the absence of delay impairments on the downstream traffic are improved at the expense of reduced energy saving. Therefore, no upper bound is necessary for T_{sleep}, resulting in the possibility of longer sleep periods which, in turn, similar or even higher energy savings than cyclic sleep.

2.2.3 Hybrid Solutions

Hybrid solutions combine both physical and data link layer solutions to reduce energy consumption. One hybrid solution is proposed in [6] for 10G-EPON where EPON and 10G-EPON OLTs and ONUs coexist. The approach is based on combining sleep and periodic wake-up (SPW) and

adaptive link rate (ALR). In the SPW, the OLT requests an ONU to switch to sleep mode because of the absence of downstream traffic. The OLT sets the ONU sleeping time after which the ONU wakes up for requesting the OLT whether it can sleep further or not. In the ALR the downlink rate is switched between 1 Gb/s and 10 Gb/s based on the amount of traffic. It is again the OLT issuing a control message to the ONU for switching between 1 Gb/s and 10 Gb/s downstream transmission. The ONU replies with an acknowledgement message after having performed the receiver switching. Simulation results show that the hybrid mechanism is capable of providing larger power saving than the two methods in isolation. In addition, in [6] the successful implementation of the proposed method in a testbed is also shown.

Chapter 3

Testbed Architecture

This chapter describes the ECONU architecture. The data path from OLT to ONU is sketched followed by a brief description of the major components along the path. The packet format and the sleep FSM are described to show the transition among the ONU states.

3.1 ECONU architecture

The ECONU implementation is based on the IEEE 802.3az-2010 standard [13]. This implementation considers downstream transmission only. The testbed architecture showing data path from OLT transmitter to ONU receiver is illustrated in Figure 3.1. The main components of the system are Ethernet packet generator, 10GbE subsystem, green header extractor and the sleep FSM. The design uses packet generator and the 10GbE subsystem provided from Altera's intellectual property (IP) and the other components are designed from scratch.

At the OLT transmitter, the client traffic is provided by an Ethernet packet generator that drives the 10GbE subsystem by transmitting a stream of Ethernet frames to the OLT transmitter FIFO. The transmitter 10Gb/s Ethernet MAC (10GbE MAC) receives the generated packets, stores them in the FIFO, and appends control fields to form an EPON frame. It also pads bit to meet the minimum frame length, if necessary. The MAC frame is passed to the 10GBASE-R which is further divided into Physical Coding Sublayer (PCS) and Physical Medium Attachment (PMA) sublayer. Both the PCS and the PMA process the frame, serialize the data, and transmit them along the medium to the ONU.

OLT and ONU functions partly use Altera Intellectual Property (IP cores), like Altera 10GbE MAC and Altera 10G BASE-R PHY IP cores. The 10GbE MAC implements the Ethernet MAC, while the 10GBASE-R PHY implements the physical layer functions, i.e., the PMA and PCS as specified in

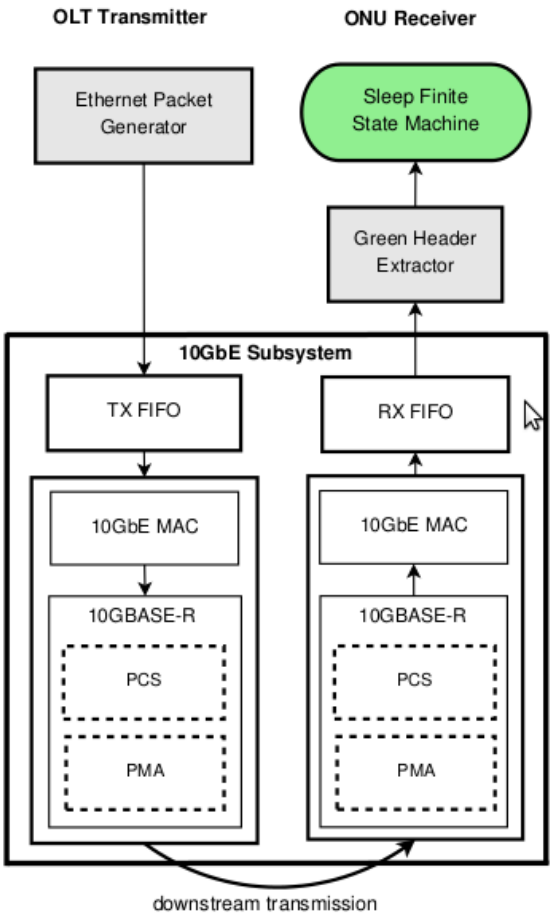

Figure 3.1: Testbed Architecture.

IEEE 802.3-2008 standard. Altera provides 10GbE subsystem (as a reference design), which integrates 10GbE MAC and 10G BASE-R PHY. The design also includes transmitter and receiver FIFO buffers, MAC, PCS and PMA as shown in Figure 3.1.

The MAC frame format of the 10GbE IP core is different from the one defined in the IEEE 802.3-2008 standard as depicted in Figure 3.3. When compared to the standard Ethernet frame format, the packet received by the 10GbE TX MAC from the OLT FIFO has an additional 2-byte sequence number field. This allows Avalon streaming (Avalon-ST) to keep track of the generated packets.

The packet generator (utilizing the Avalon-ST interface) generates a packet with MAC addresses (i.e., destination and source), length/type, payload data, sequence number, and it calculates and appends the cyclic redundancy check (CRC) after the payload. The packet is then forwarded to the TX MAC. In addition, a 7 byte start of payload field and 1 byte green header are inserted at the beginning of the payload. The start of payload is defined as a sequence of 56 bit ”1s”, while the remaining 1 byte contains the green header which encodes the control bit. This control bit allows the OLT to request the ONU to switch to sleep state (when ”1”) or to maintain its current state (when ”0”).

The FIFO's temporarily store packets and it interfaces the Ethernet packet generator and the 10GbE MAC through Avalon Streaming (Avalon-ST) protocol. It connects the generator of the data stream (source) to a receiver of the data (sink).

The 10GbE MAC and the 10GBASE-R interface each other by using the single data rate (SDR) version of the XGMII protocol. The SDR XGMII consists of 64-bit data bus and 8-bit control bus operating at 156.25 MHz. The SDR XGMII is slightly different from the Double Data Rate (DDR) XGMII that utilizes two 32-bit interface with the frequency of 156.25 MHz as defined in IEEE 802.3-2008 Clause 46. All components use Avalon memory-mapped (Avalon-MM) control and status register (CSR) interface. This is a master-slave interface that uses byte addressing to enable host PC to access 32-bit configuration and status registers, and statistics counters by means of Quartus 2 on-chip debugging tools.

The ONU receiver works complementary to the OLT transmitter, with some significant modification done on the ONU to implement the sleep technique. After stripping off the control fields and padding, which were inserted at the MAC transmitter, the packet is stored in the ONU receiver first in first out (FIFO)buffers. The FIFOs based on Avalon-ST interface: receives, buffers and transmits data between the MAC and the client, i.e., to the sleep controller module. This implementation uses FIFO buffers which are 64-bit wide and 512-word deep. The ONU receiver strips off the frame overheads and forwards the data to the header extractor module.

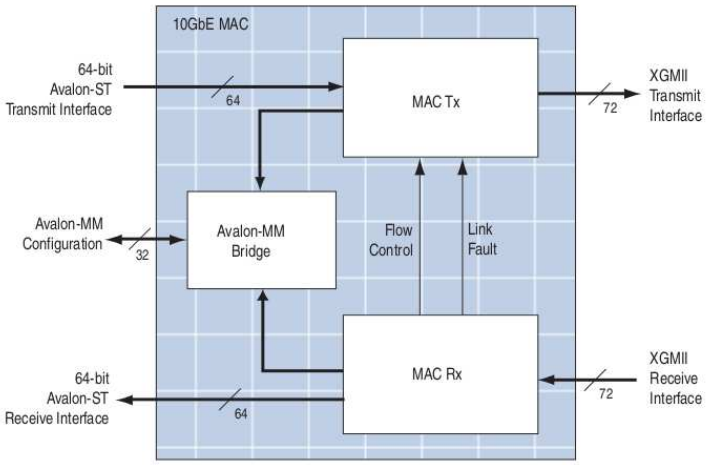

Figure 3.2: 10GbE MAC IP Core Block Diagram.

The packet extractor module is implemented to extract the green header (control header), which regulates the ONU behavior. The content of the green header is processed by the Sleep FSM at the ONU side that determines if the ONU must be switched to ”on” or ”off” state.

3.2 10GbE subsystem

The Figure 3.2 describes briefly all the components which constitute the 10GbE Subsystem provided by Alter IP cores [25, 26].

3.2.1 10GbE MAC

The 10GbE MAC IP core is composed of three main blocks: MAC receiver (MAC Rx), MAC transmitter (MAC Tx), and Avalon-MM bridge. The MAC Rx and MAC Tx are responsible for handling the data flow between the client and Ethernet network. In each of these blocks, there is a 64-bit wide Avalon-ST interface on the client side and a SDR 10 gigabit media independent interface (XGMII) on the network side. The Avalon-MM bridge provides a single interface to all Avalon-MM interfaces within the MAC, which allows a host to access 32-bit configuration and status registers, and statistics counters.

3.2.2 Physical Coding Sublayer

Part of the physical layer specification for networking protocols is implemented by the PCS. The function included by the PCS depends on the

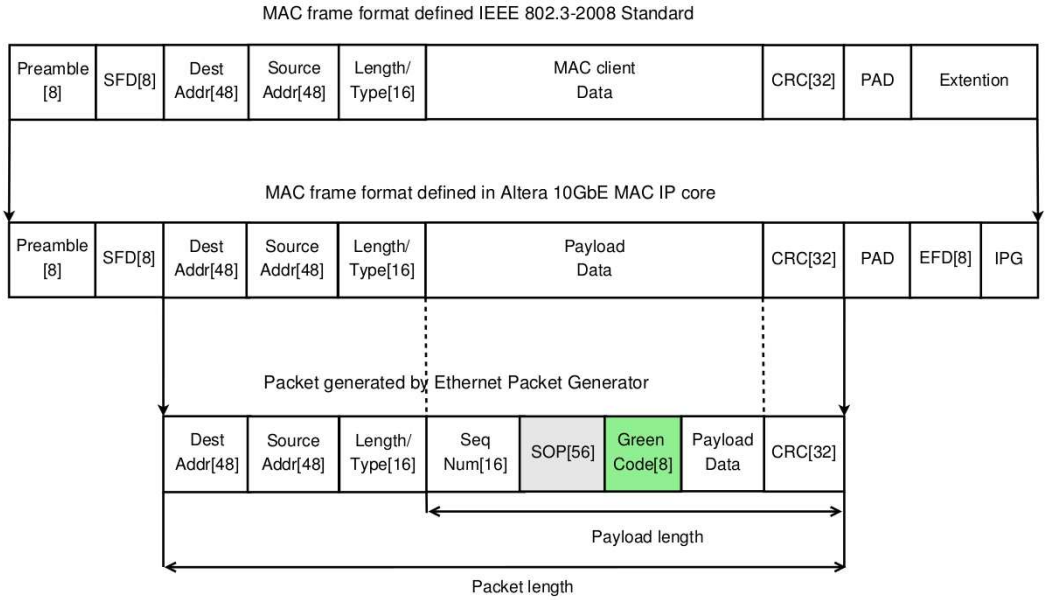

Figure 3.3: Frame Format.

type of network protocol used by the design. The following list includes list of the functions within PCS: 8B/10B, 64B/66B, or 64B/67B encoding and decoding, rate matching and clock compensation, scrambling and Descrambling, word alignment, phase compensation, error monitoring, and gearbox.

3.2.3 Physical Medium Attachment

The main functionality of the PMA is to convert the parallel input data streams to serial data. The PMA receives and transmits differential serial data on the device external pins. The transmit (TX) channel supports programmable output differential voltage (V_{OD}) and programmable pre-emphasis. The receive (RX) channel supports programmable equalization and offset cancellation (to correct for process variation). It converts serial data to parallel data for processing in the PCS. The PMA also includes a clock data recovery (CDR) module with separate CDR logic for each RX channel.

3.3 Frame format

The MAC frame format of the 10GbE IP core is different from the one defined in the IEEE 802.3-2008 standard as depicted in Figure 3.3. When compared to the standard Ethernet frame format, the packet received by the 10GbE TX MAC from the OLT FIFO has an additional 2-byte sequence

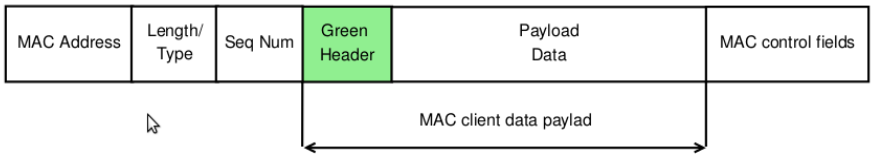

Figure 3.4: Packet Format.

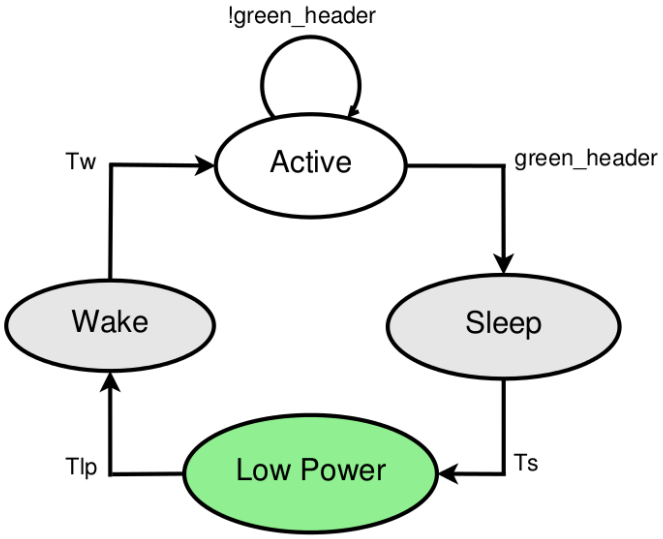

Figure 3.5: Sleep Finite State Machine.

number field. This allows Avalon-ST to keep track of the generated packets.

The packets sent by the OLT to the ONU are a standard Ethernet frame except for the inclusion of the green header as shown in Figure 3.4. The green header is explicitly defined by us and it carries the control information that allows the OLT and ONU to exchange the sleep control signals keeping the standard Ethernet packet format.

3.4 Sleep Finite State Machine

The sleep FSM implemented at the ONU receiver is based on [12] and it is a simplified version of the low power idle (LPI) FSM defined in [13]. At any given time, the ONU can only be in one of the four states as shown in Figure 3.5. The ONU receives data only in Active state. Only after receiving a sleep request from the OLT, the ONU switches from Active state to Sleep state. The sleep request is carried by the green header in the EPON frame. The ONU stays in the Sleep state for a period of T_s, which is the

time required by the ONU to turn off its transceivers. When the T_s expires, the ONU switches to the Low Power state and sleeps for a period T_{lp}, resulting in low power consumption. When the ONU is in Low Power state, the received downstream packets can either be buffered or ignored. This implementation adopts the latter. When the T_{lp} expires, the ONU switches from Low Power state to Wake state for a period of T_w , which is the time required by the ONU to turn on the transceivers and to synchronize with the network. The T_s , T_{lp} and T_w values are defined as in [13].

Chapter 4

Testbed Implementation

This chapter presents a step-by-step approach adopted for implementing the testbed.

4.1 VLSI Design flow

Figure 4.1 represents a typical very large scale integration (VLSI) design flow cycle. The electronic design automation (EDA) tools for implementing the testbed are shown on the right.

4.1.1 Design specification

A design specification provides precise and explicit information about the requirements for a product design. It is the result of lot of market research which is done by the company for its target users, finally given to design architect. ”It is the bible for the design engineers”. Any design specification should be able to produce a consistent outcome for any number to times repeated, irrespective of the person who models it. After this, the specification should be frozen because it should not disturb the hardware team.

4.1.2 Behavioral description

The design architect then models the behavior of the target product based on specification. Design architect breaks the behavior into various function blocks. At this stage the model breaks from a complex problem to smaller modules.

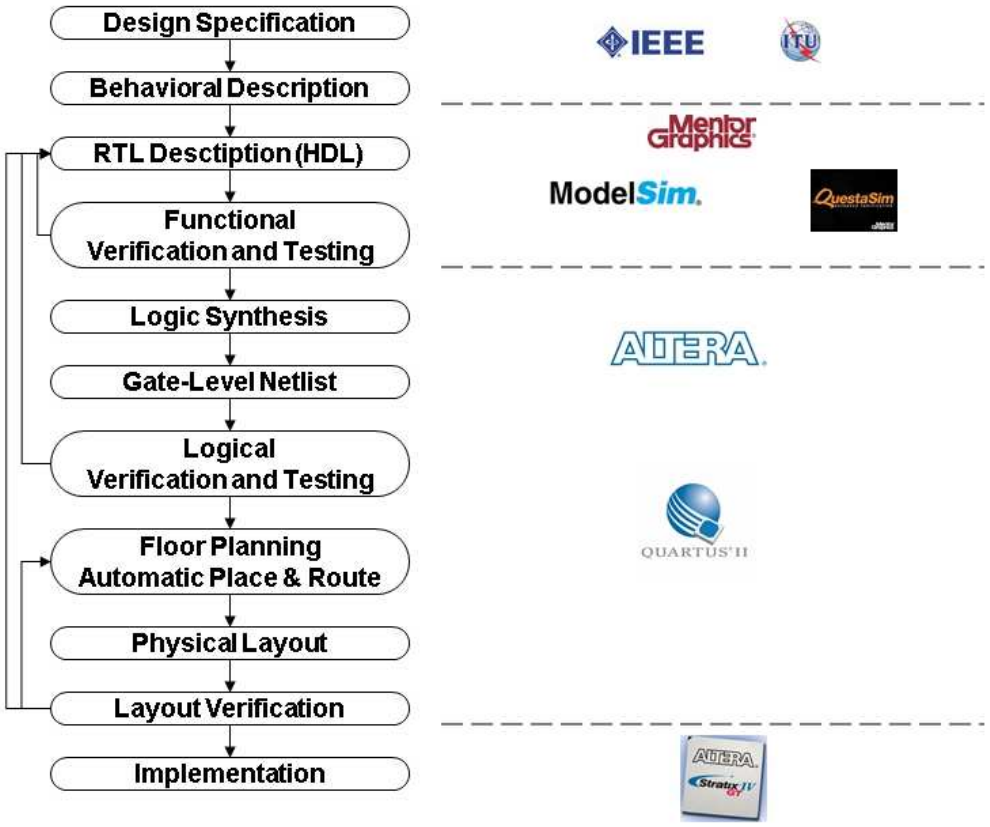

Figure 4.1: VLSI Design Flow.

4.1.3 RTL description

Now the smaller modules are given into design engineers .The design engineers mimic the functions of the device using register transfer logic (RTL) descriptions using HDL (Hardware Description Language). They can follow both the top down and bottom up approach. It is also possible that the verification team also starts with test benches for testing the RTL for Design Under Test (DUT). During this stage, the designer uses the EDA tools for functional behavior, verification and testing. During this thesis, ModelSim tool from Mentor Graphics was used.

4.1.4 Functional verification and testing

The verification team (if it's a large project) is ready with test benches and test cases to check the design. Test bench is the top module which includes the DUT. Test cases are various possible input combinations (for n bit input, there 2^n possible combinations). If any flaws are encountered , immediately the design term is notified. The design team modifies the design and is again tested and verified by the verification team. EDA tools like ModelSim and QuestaSim from Mentor Graphics is generally used during this design cycle.

4.1.5 Logic synthesis, verification and testing

Synthesis is the process of creating a gate level net list from the HDL. The synthesizer will try to match gates for the HDL from the standard libraries. It is a small prototype of the logic in the first level (gate level). The logic circuit is tested in the gate level (delays and drive strength) to meet the timing requirements. From this stage, Altera's Quartus 2 tool is used for synthesis and configuration of the FPGA.

4.1.6 Floor planning, place and route

Floor planning is the process of identifying structures that should be placed close together, and allocating space so as to meet requirements like space (cost of the chip), required performance, and the desire to have everything close to everything else. In certain chips it is often the case that the smallest area design is also the highest performance design. During this stage, the design can be optimized for either speed or area.

4.1.7 Physical layout and verification

Physical verification is a process whereby an IC is checked via EDA software tools to see if it meets certain criteria. Verification involves design rule check (DRC), layout versus schematic (LVS) and electrical rule check (ERC).

4.1.8 Implementation and tape-out

The next step will be to implement the design on a FPGA and is also sometimes referred to as FPGA prototyping. Once the design meets the functional and timing constraints, it moves into the final step. The last step is the tape-out, where the design is fabricated and realized as a chip.

4.2 Modules developed and used in this design

(1) The exponential distribution function, header extractor, sleep FSM (and part of Altera packet generator) module followed the VLSI design flow. A lot of time was spent in understanding the concept, dividing them into smaller functional blocks, writing the RTL in Verilog HDL, simulating, verifying, synthesizing and implementing the design on FPGA.

(2) In case of the 10G Ethernet module which is an Altera IP core function, the steps have already been followed; hence the design was used directly.

4.3 Board specification

The testbed for implementing the energy-efficient ONU utilizes the Altera Transceiver Signal Integrity Development Kit, Stratix IV GT edition board (EP4S1002F40I1). This is a low power and high performance FPGA with integrated 11.3 Gbps transceivers. Figure 4.2 shows the board layout describing various components available in the Altera board. More detailed information about the board can be found in [27]

4.4 Bird's eye view of the entire design

Figure 4.3 provides a bird's eye view of the entire design. This view is captured from the RTL viewer option in Quartus 2. The view depicts the relation between the various components used in the design. At the top level

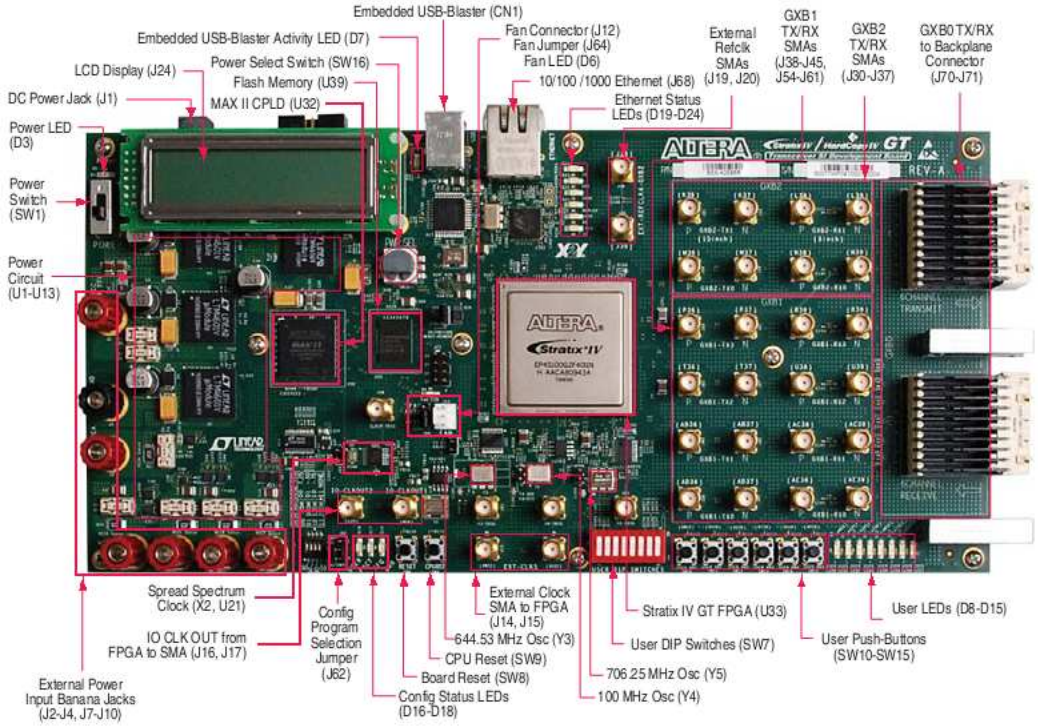

Figure 4.2: Altera Stratix IV GT board.

abstraction, there is the Ethernet block (implementing the Ethernet functionality provided by Altera), followed by the the sleep controller module (integrating the header extraction module and the sleep FSM), System controller (acts like a central processing unit, controlling the whole system through control signal generation provided by Altera), Avalon ST traffic generator module (integrates the generator and the monitor modules provided by Altera).

4.5 Exponential distribution implementation

Assuming that the packets arriving in the network follow a Poisson process, the inter-arrival times is modeled using the exponential distribution.

4.5.1 Generation of an exponential distribution function

The exponential distribution function is modeled using a C program. The inverse transform method is used for the generation of exponential distribution as describe in [28].

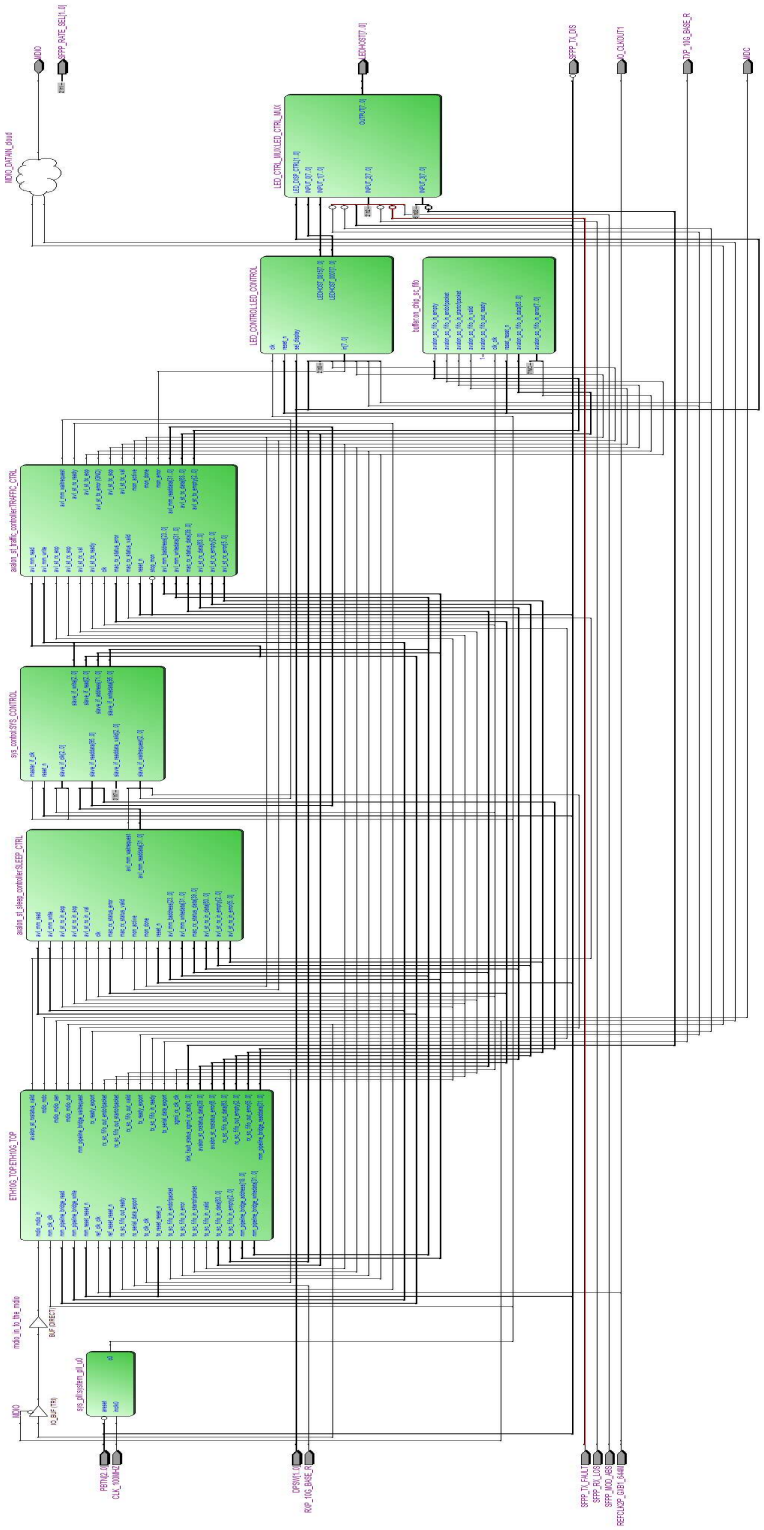

Figure 4.3: Bird's eye view of the entire design from Quartus viewing tools.

4.5.2 Inverse Transform

First step towards generating the exponential distribution is the generation of random numbers. There are many different algorithms for generating random variables depending on the distribution we wish to generate.

Suppose a random variate X is to be generated, which is continuous and has distribution function F that is continuous and strictly increasing when $0 < F(x) < 1$. Let F^{-1} denote the inverse of the function F . Then an algorithm for generating X having distribution function F can be described as follows:

1. Generate $U \sim U(0, 1)$

2. Return $X = F^{-1}(U)$ Where $F^{-1}(U)$ is always defined as $0 < U < 1$, the range of F in [0,1]. In our application, the uniform random distribution between 0 and 1 is generated by the following function $u = (double)rand()/(double)RAND\_MAX;$

   The exponential distribution function with mean β is given as follows:

$$f(x) = \begin{cases} 1 - e^{\frac{-x}{\beta}} & \text{x}\geq 0 \\ 0 & \text{x}<0 \end{cases}$$

To obtain the inverse of the function F^{-1}, we set $u = F(x)$ and solve for x to obtain
$F^{-1}(u) = -\beta \cdot \ln(1 - u)$

Thus to generate the desired random variable, we first generate $U \sim U(0, 1)$ and then let $X = -\beta \cdot \ln(1 - u)$.

The $delta\_t$ corresponds to the inter-arrival time. The value of β is the ratio between the packet time (1.25 μs) and the bit time (6.4 ns)

$delta\_t = -(200 \cdot \log_2(1 - u))/64$

The following value delta corresponds to the integer value of the inter-arrival time which is of floating point data type
$delta = (int)delta\_t$

The following plot shows the relation between the uniformly distributed random number u along the y-axis against the exponential distribution delta_t along the x-axis.

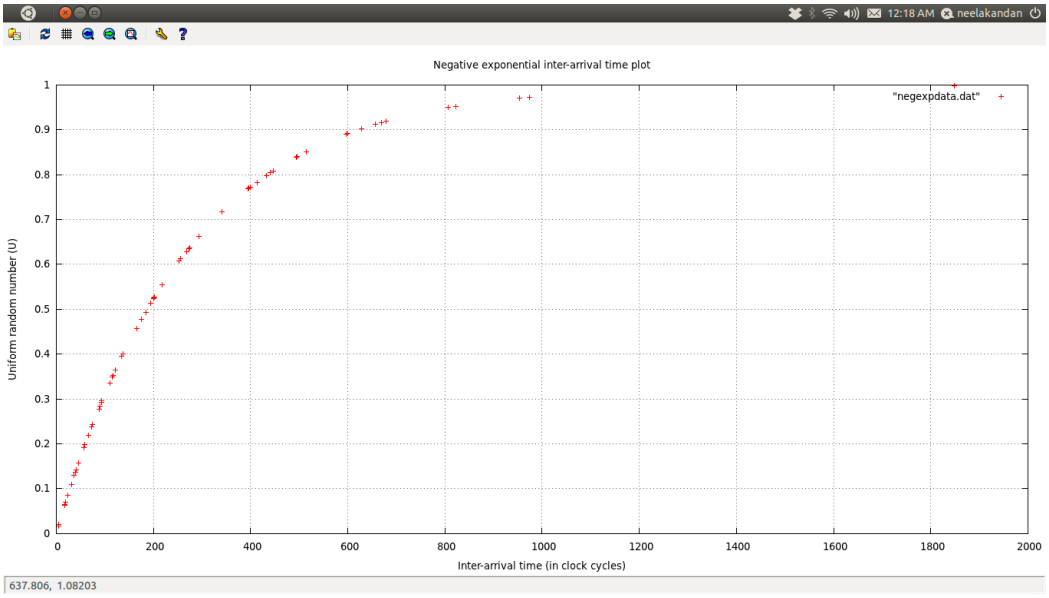

Figure 4.4: Negative exponential distribution.

4.5.3 Implementation of the Exponential Distribution function

Figure 4.5 shows the block diagram explaining the implementation of exponential distribution function which imitates the arrival of packets in a Poisson distribution.

A C program is used to generate the exponential inter-arrival time based on the Inverse-transform. The generated interval times are then converted into number of clock cycles with reference to the base clock. The resulting values are stored in a .mif (memory information file) format by using the file operations in C. This resulting .mif, holding the exponential inter-arrival times (in number of clock cycles) is used as an initialization file for a single port RAM module generated by Altera Megacore functions. A hexadecimal file (.hex) can also be used to store the values instead of .mif. There is the exponential reader module, which is the core module implementing the functionality of the exponential distribution. The reader module initializes a counter with the value obtained from the MEM_DATA signal from RAM, which then decrements the counter till it goes to zero. As soon as it reaches zero, it generates a control signal. This control signal triggers (acting as the input to) the COUNTER module decides on whether to increment the counter. The value at the output of the COUNTER module corresponds to the RAM_ADDRESS of the instantiated RAM Megacore function.

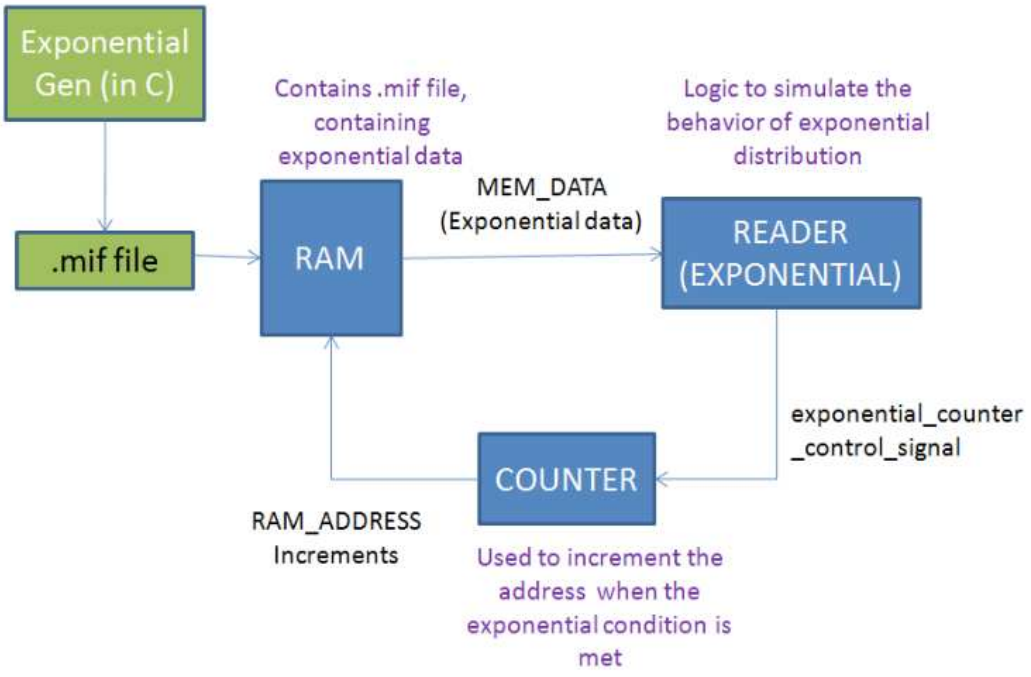

Figure 4.5: Block diagram of the negative exponential distribution in FPGA.

4.6 ONU Implementation

Figure 4.6 depicts the Sleep controller module that covers the main work done during the thesis. It involves the development of two modules, the header extractor and the sleep control FSM. Both the modules have adopted the Avalon-MM standards (refer Figure 4.6). By using this standard, the individual modules can

- keep track of the packets using Start Of Packet (SOP) and End Of Packet (EOP) signals.

- configure read and write instructions into the control and status registers (CSR) through the TCL interface in System Console, that can be used to extract the statistics during run time.

The modules also have some signals from monitor (like mon_active and mon_done) which indicates when the monitor starts and monitor ends. These signals are used as control signals to extract the energy efficiency statistics from the receiver state machine.

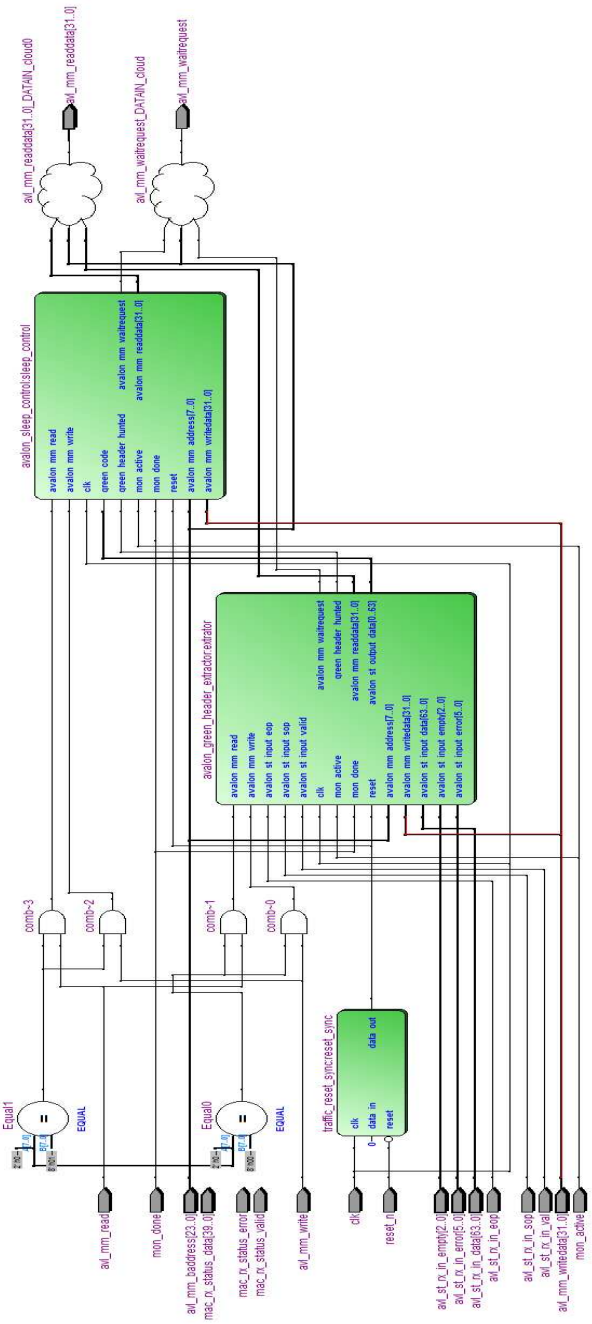

Figure 4.6: Bird's eye view of the Header extractor module and the FSM.

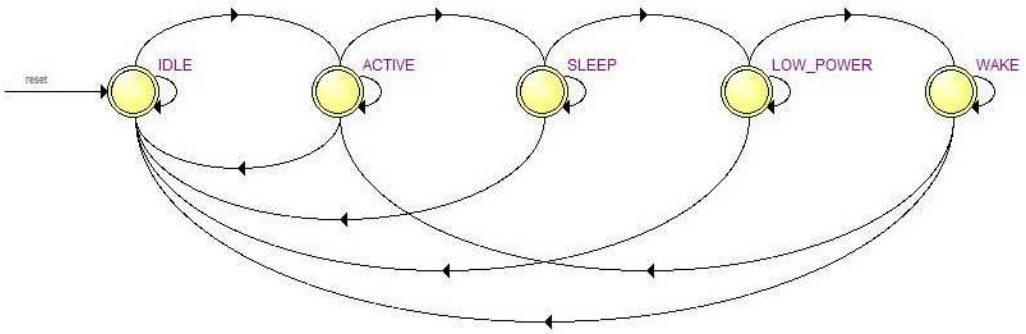

Figure 4.7: Graphical representation of the FSM transition among the states.

4.6.1 Header extractor module

The extractor module is used to extract the green packet from the incoming packet stream. In this thesis, the design tracks the second data chunk corresponding to the source address to track the green packets. Once the packets are tracked, they are passed on to the state machine to traverse through the states.

A specific packet format for this design as defined in Figure 3.4, with the insertion of the green header inside the user payload data of the Ethernet frame. The design implementation of the extractor deals with identifying the source address and then immediately capturing the next data chunk to extract the first byte containing the green packet. This enables the exchange of sleep request between OLT and ONU by driving the Sleep FSM. This module is a trigger and register based design.

4.6.2 Finite State Machine for implementing the Sleep mode

The FSM used for the implementation of energy efficient ECONU has four states as described in the Figure 4.7: Active, Sleep, Low power and Wake. The following section gives a brief description of the states, their conditions for transition and the operation of the ONU in the specific state.

Based on the content of the green header first bit and the timers defined for each state, the sleep FSM, switches among states. For example, if a one is received when the ONU is in the Active state, the process of setting the ONU to sleep is triggered. Constant values for the timers regulating the sojourn time in each state are utilized. The values of such timers are the ones recommended in the IEEE 802.3az standard [13]: $T_s = 2.88\mu s$, $T_w = 4.48\mu s$, and $T_{lp} = 39.68\mu s$. The following describes the condition and the timers used for transitions among different states at the ONU.

As soon as a frame arrives and if the 1st bit of the green code corresponds to 1, there is an immediate transition from the Idle state to the Active state. During the Active state frames are received. When the frames are received in this state, the DATA_RX_COUNTER and STAT_ACTIVE_COUNTER is incremented. The FSM continues to remain in the same state till another frame with green code corresponding to 1 arrives. This state can also be designed to work with a timeout, so that the ONU remains in this state till the timer (corresponding to the timeout) expires.

When another frame arrives and if the 1st bit of the green code corresponds to 1, the FSM transitions from the Active state to the Sleep state. In ideal cases, transition should occur only when the link has completely transferred all the buffered frames. Our design implements a simplified version and does not include the buffering of frames. During this state STAT_SLEEP_COUNTER and DATA_LOST_COUNTER is incremented. The STAT_SLEEP_COUNTER indicates the overhead time associated with the transition towards the low power state (i.e., time taken by the ONU to switch off its transceivers). According to IEEE 803.az standard, this time corresponds to $2.88\mu s$. So in this as the ONU is working with 156.25 MHz clock cycles, the sleep time corresponds to 450 cycles.

Once the minimum sleep time ($2.88\mu s$) in the sleep state has been spent, the ONU immediately transitions to the Low Power Idle state. Here the ONU doesn't need to wait for the green header as it has already received a sleep request. The time spent in this state is indicated by the STAT_LOWPOWER_COUNTER. Since this state is the actual power saving state, the amount of time ONU spent in this state should be more than the amount of time taken for the sleep and wake transition . The design implements a low power time equal to $39.68\mu s$ as defined in the IEEE 803.az standard. So the ONU stays in this state for 6200 cycles (with reference to the frequency of 156.25 MHz used in the design). DATA_LOST_COUNTER is also incremented as the ONU is assumed to be sleeping.

The last state of the FSM is the Wake state. The transition to the Wake state from the Low power idle state occurs when the minimum Low Power Idle state time ($39.68\mu s$) has been spent. DATA_LOST_COUNTER is incremented as the ONU is still in sleep. The STAT_WAKE_COUNTER indicates the overhead time associated with the transition from the low power idle state (i.e., the time taken by the ONU to switch on its transceivers). According to IEEE 803.az, the wake time corresponds to $4.48\mu s$ which corresponds to 700 cycles. When the minimum wake up time ($4.48\mu s$) is reached, the ONU transitions to the Active state and continues.

4.6.3 Implementing the sleep time in the FSM

The following section explains the technique for implementing the sleep time, wake up time and the low power time explained in the IEEE 803.az. This section immediately follows with the results obtained on Signal tap logic analyzer (On-chip debugging tool) to verify the timings.

For 10Gbps, the reference clock frequency is $f = 156.25MHz$ then the time corresponding is given by $t = 1/f = 6.4ns$

This implies that the a 64-bit data arrives every $6.4ns$.

If the sleep time of the ONU is $2.88\mu s$, then

Number of cycles to be slept = $2.88\mu s/6.4ns$ = 450 cycles

From the above discussion it is clear that by knowing the number of cycles for which the ONU sleeps (wakes or remain in low power state), these values can be set as sleep, wake and low power idle time counters. Once the FSM comes to one of the states, the corresponding local counter will increment till it reaches the threshold value set by the standards. Once the local counter matches the threshold, control signals are generated. Based on the input conditions, control signals and the state machine design, the state transitions occur.

4.7 SignalTap II Embedded Logic Analyzer

SignalTap II is an embedded logic analyzer that samples, captures and stores the logic state of an FPGA internal signals on the rising edge of the defined sampling clock. It provides the opportunity to monitor the signals in realtime for on-chip debugging the Quartus II software through the joint test action group (JTAG) connection. SignalTap II is used for debugging during the process flow.

4.7.1 Debugging using SignalTap

The following steps briefly describe the debugging task flow described in Figure 4.8

- Create a new instance (or multiple instances) of the SignalTap II logic analyzer and add it to the design.
- Buffers in SignalTap can be configured to have a maximum of 1024 data channels, 128K sample depth and up to 10 trigger conditions.
- Compile the entire design with the SignalTap instance. Sometimes when the design is large or the number of signals tapped is large, the

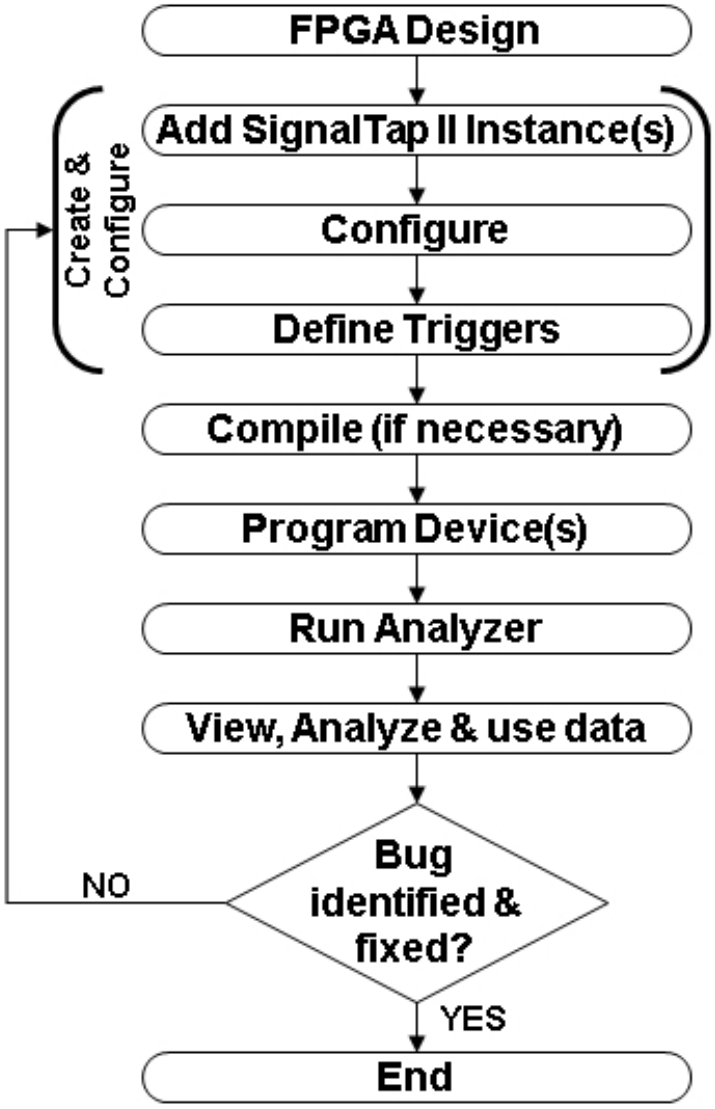

Figure 4.8: Debugging task flow using SignalTap logic analyzer.

compile time can be very long. Under such condition incremental compilation can be used to to reduce recompile times.

- Program the FPGA using the JTAG.

- SignalTap II logic analyzer captures data when the defined trigger conditions are met.

- Then SignalTap provides the results, which can be analyzed to locate and fix bugs in the design

- There are many options available like the storage qualification, state-based triggering flow and power-up triggers. They can be used based on our requirements.

4.7.2 Implication on the design

- When a SignalTap instance is created to view signals in different different clock domains, SignalTap utilizes the unused logical and memory blocks.As it occupies logic and memory (buffer) resources of the FPGA, a careful evaluation of the resource utilization is to be performed on what signals are to be tapped, if the design in itself occupies more space.

- LE count increases based on the channels monitored, trigger conditions and sample depth.

- We can have 1024 data channels, 128K sample depth and upto 10 trigger conditions.

- Since there is low buffer space, only data samples of interest can be buffered based on data storage qualification.

- Every time a new or additional signal is added to the SignalTap instance, the design has to be recompiled (as the resource utilization changes). The compilation time increases with the complexity of the design. It can be reduced by partitioning the design.

- SignalTap can be used only for functional debugging. It doesn't perform any timing analysis.

- SignalTap can be used for only designs targeting FPGA and not CPLD.

4.8 System console

The System Console performs low-level hardware debugging of the design. This is a Tcl console that provides access to the hardware modules instantiated in the FPGA. The System Console was mainly used in the design to read or write Avalon-MM slaves using special masters. The read and write operation is performed on the CSR based on the commands typed on the system console. The CSR generates control signals and manipulates the design based on desired requirements.

4.8.1 Uses of System console

- To start, stop, or step a Nios II processor
- To sample the system on a programmable chip (SOPC) system, clock as well as system reset signal are used.
- To run JTAG loopback tests to analyze board noise problems
- To shift arbitrary instruction register and data register values to instantiated system level debug (SLD) nodes

4.9 Resource Utilization in FPGA

A major consideration when programming FPGAs is the amount of circuitry (resources) the code uses on the chip. It is sometimes useful to know how much space a specific function will use when translated to LUTs and flip-flops. This section provides complete resource utilization statistics for the design on the Stratix 4GT FPGA. Some of the parameters considered for the resource utilization statistics are: logic elements, registers, Adaptive Logic Modules (ALMs), Logic Array Blocks (LABs), I/O pins, global clocks, Phase-Locked Loops (PLLs) of GXB transmitter and receiver channels, memory blocks and interconnects.

Table 4.1presents a brief overview on some of the important parameters with reference to the device capabilities. Table 4.2 provides the resource unitization of the design. The design shows a higher utilization of the memory because it also takes into account the signal tap instance, which is a on-chip debugging tool provided by Altera along with Quartus 2.

From Table 4.2 it can be understood that, design utilizes only a low percentage of the FPGA resources. This is of great benefit to us, as the design can be scaled to higher dimensions and also the implementation

can be made more accurate (and complex) by increasing the control logic involved.

Table 4.1: Stratix 4 GT device overview.

Family	Stratix 4
Device	P4S100G2F40I1
Equivalent LEs	228000
Adaptive Logic Modules (ALMs)	91200
Registers	182400
M9K Memory Block	1235
M144K Memory Blocks	22
Embedded Memory (Kb)	14283
MLAB (Kb)	2850
18 * 18 Multipliers	1288

Table 4.2: Resource utilization of the design.

Parameter	Available	Utilized	% Utilization
Logic Utilization			16%
Combinational ALUTs	15869	182400	9%
Memory ALUTs	73	91200	1%
Dedicated Logic Registers	16710	182400	9%
Total Registers	16710		
Total Pins	31	798	4%
Total block memory bits	7889260	1.5E+07	54%
Total PLLs	2	8	25%
Total GXB Receiver Channel PCS	1	24	4%
Total GXB Receiver Channel PMA	1	36	3%
Total GXB Transmitter Channel PCS	1	24	4%
Total GXB Transmitter Channel PMA	1	36	3%

Chapter 5

Results

Implementation was done in two different modes internal loop back and local loop back. This chapter analyses and presents the obtained results along with energy efficiency and performance statistics for the above modes. Finally the results of the design with negative exponential distribution based packet arrival is also presented.

5.1 Internal Loopback

Internal loopback is available for all transceiver configurations. Generally, the serial loopback is used as a debugging aid to ensure that the enabled physical coding sublayer (PCS) and physical media attachment (PMA) blocks in the transmitter and receiver channels are functioning correctly. Furthermore, the serial loopback can be enabled dynamically on a channel-by-channel basis through the TCL command line interface provided in System Console. A more detailed information about the loopbacks can be found in [29]. This mode is used to take the initial set of energy-efficiency and performance statistics for this design.

5.1.1 Experimental setup

The following description explains how the internal loopback (Tx-to-Rx) was performed. The system can be configured to send 1000 packets with one iteration having a data length of 1500 bytes each. Experiments were performed with a target confidence interval of 95%. It is found the data and statistics obtained are very stable. On average the experiments were run for 50 iterations for each packet length (100,500,1000,1500) with burst size of 1000 (So the data bursts are 1000, 2000,3000... 10000).

To configure and run this test, the following steps were performed:

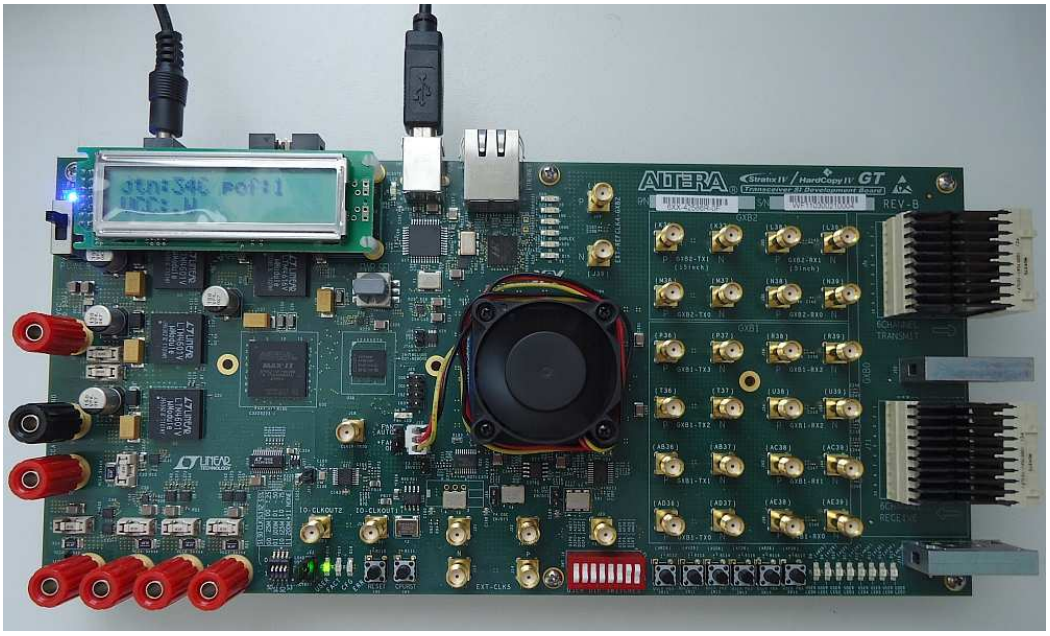

Figure 5.1: Internal loopback.

(1) The FPGA is programmed with the .sof file corresponding to our design.

(2) The transceiver tool kit is opened and the TCL transcript window is used.

(3) The path corresponding to the source TCL files is provided using the following command.
cd <path>

(4) The specific test file is sourced using the following command.
source <testfile.tcl>

(5) The loopback required is chosen from the TCL interface by typing in the command line.

(6) To run the internal loopback test (and start both the Packet Monitor and the Packet Generator) the following command is typed in the command prompt.
TEST ALTPMA <BURST SIZE> <ITERATION>

(7) The tests can be configured to run for different packet sizes by changing the packet size variable in the parameter.tcl, which is found inside the folder containing all the script files.

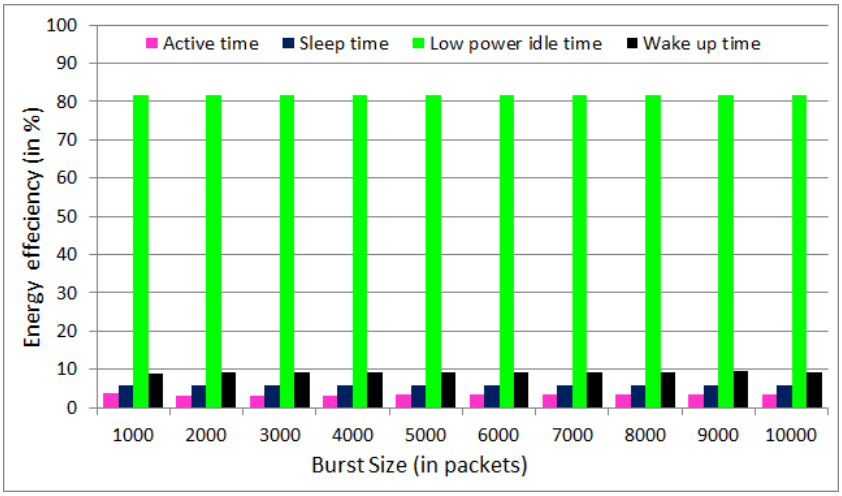

Figure 5.2: Energy Savings for Packet size of 1500.

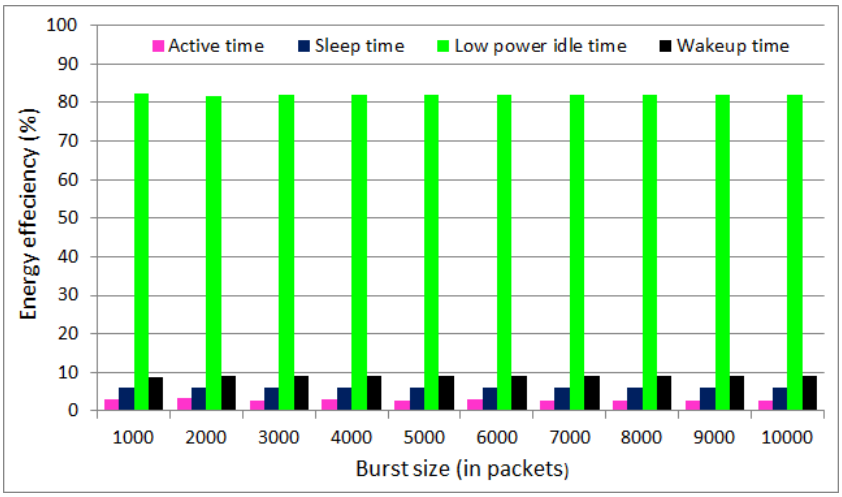

Figure 5.3: Energy Savings for Packet size of 1000.

5.1.2 Results

Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5 represent the energy efficiency characteristics of the ONU for packet sizes of 1500, 1000, 500 and 100 respectively. The results show a uniform pattern of the energy efficiency characteristics for different burst sizes. The four horizontal blocks for each burst size corresponds to the time spent by the ONU in predefined states : Active, Sleep, Low power idle and Wake. Our main interest is the Low power idle state in which the ONU power consumption is low. The Sleep and the Wake states corresponds to the transitions when the ONU goes into/comes out of Sleep state. Active state is the one in which the ONU performs its normal operation. Detailed analysis can be found in the inference section

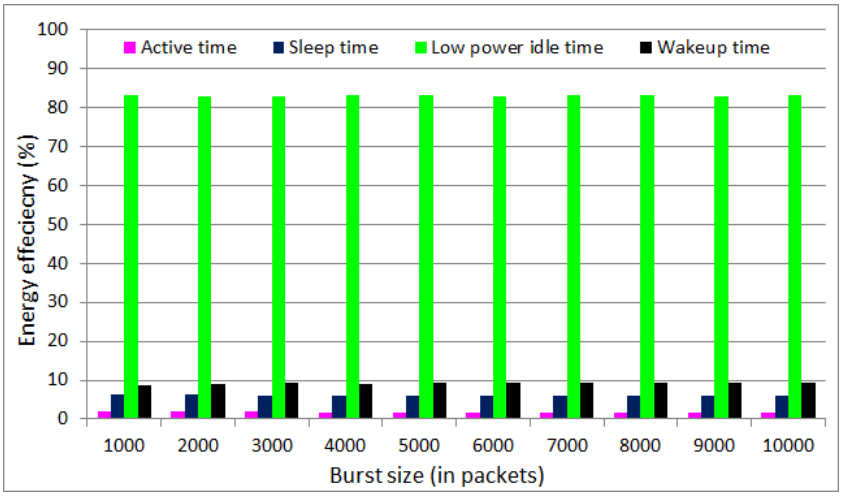

Figure 5.4: Energy Savings for Packet size of 500.

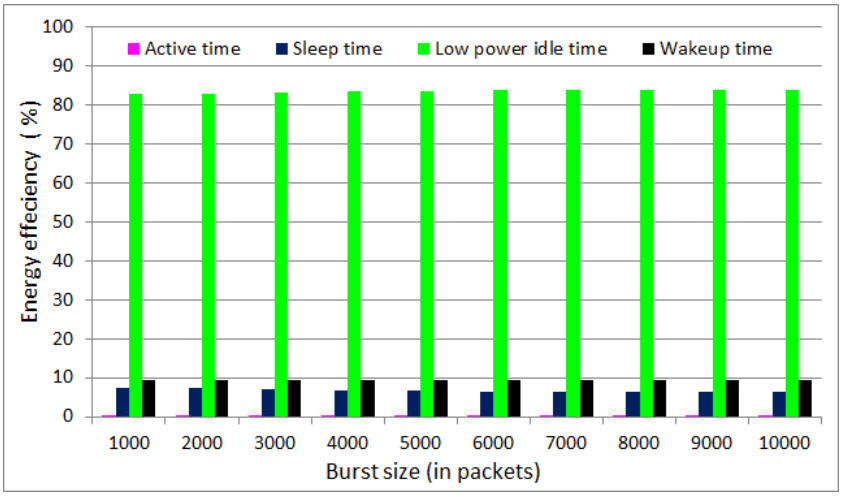

Figure 5.5: Energy Savings for Packet size of 100.

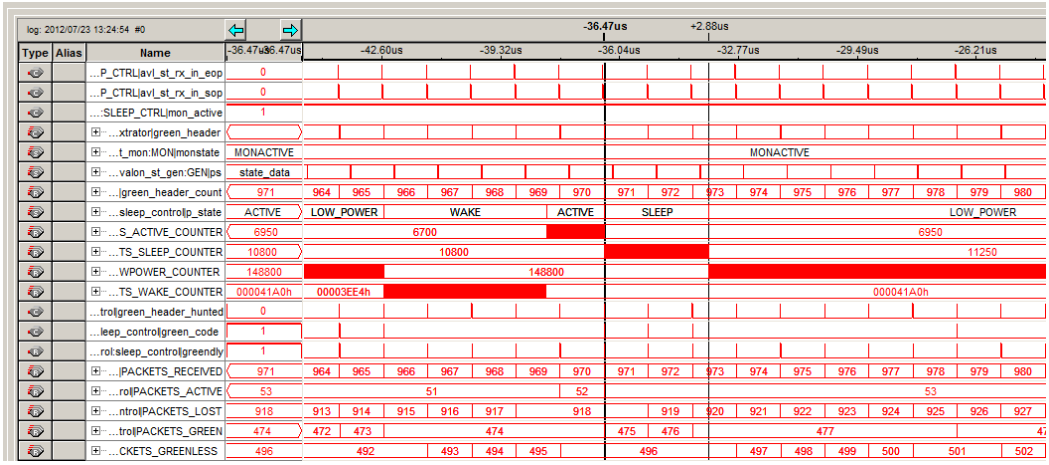

Figure 5.6: Signal tap result for Sleep time = 2.88 μ s.

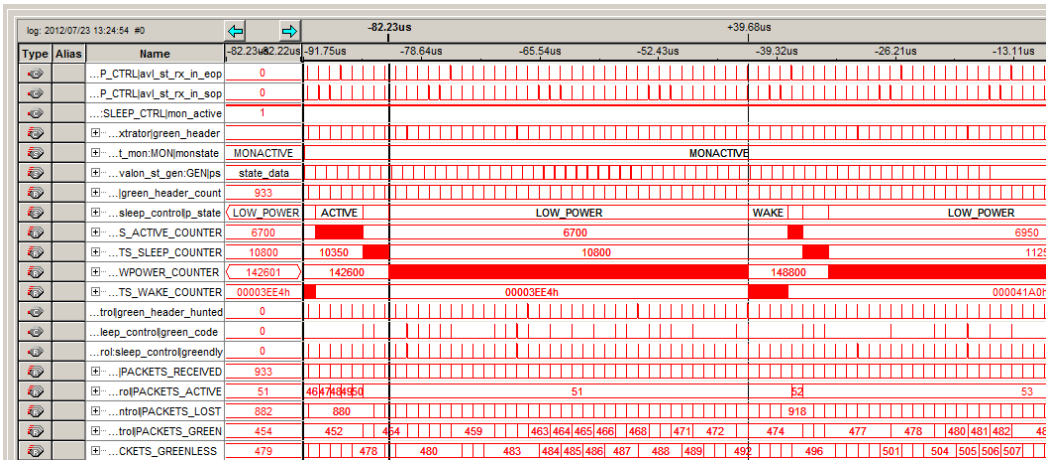

Figure 5.7: Signal tap result for Low power time = 39.68 μ s.

5.1.3 Sample results using SignalTap

The following section shows the Figure 5.6, Figure 5.7 and Figure 5.8, which show the SignalTap results for the sleep time, low power idle time and wake up times set during the thesis (in accordance with the IEEE 803.az standard).

5.1.4 Throughput Vs. packet size

Figure 5.9 shows the throughput observed in our implementation for varying packet size in the case of local loopback and internal loopback.

It can be observed that for larger packet size the throughput is higher. For larger packet size, the utilization of the link is higher because of the large number of payload. From the results it can be inferred that the gen-

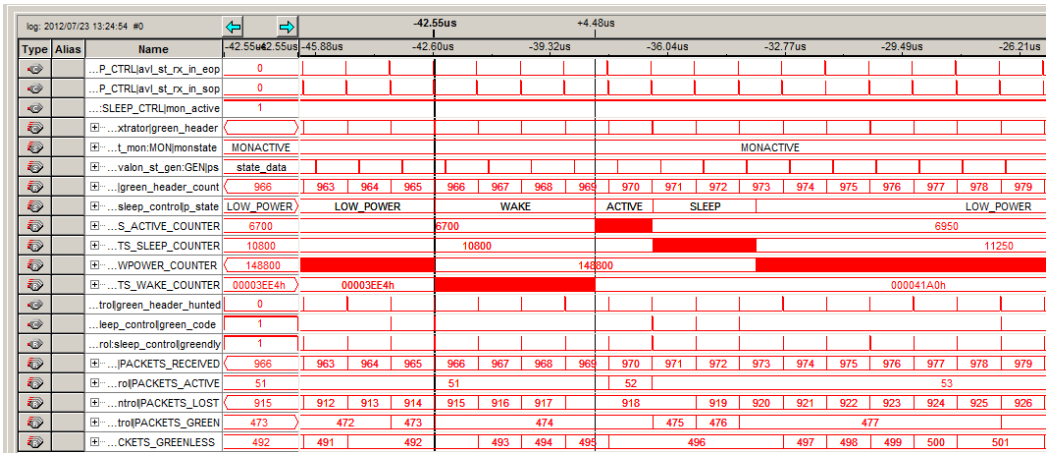

Figure 5.8: Signal tap result for wake up time = 4.48 μ s.

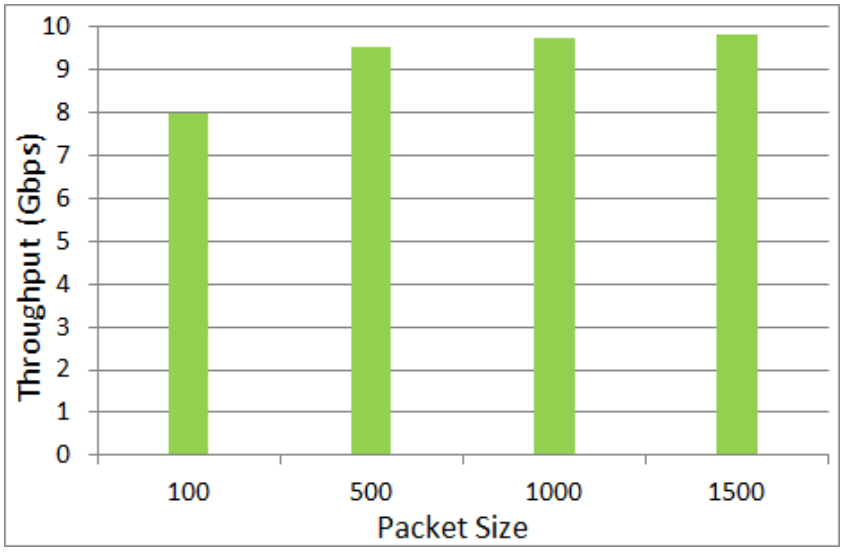

Figure 5.9: Throughput Vs Packet size for different configuration.

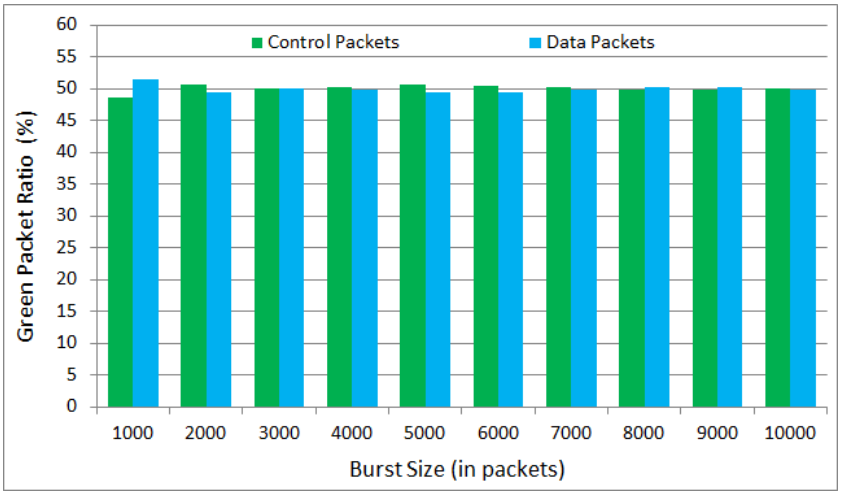

Figure 5.10: Green header generator consistency.

erator implemented in the design is working perfectly.

For the packet size of 1500 bytes, the design operates at around 9.8 Gbps. When decreasing the packet size to 1000, the throughput reduces to 9.6 Gbps. The least observed throughput in our design is 8 Gpbs corresponding to a packet size of 100 bytes.

5.1.5 Green header generator consistency

Figure 5.10 shows the results of the consistency of the green header generator for a packet size of 1500 bytes implemented in our design. For this implementation we use a LFSR (Linear Feedback Shift Register), that acts as a random number generator, by generating new values at the output based on the previous values. From the results, it can be observed that the generator generates approximately the same number of packets with the control and data in the green header.

Figure 5.10 shows the variation green packet ratio, which is the ratio of the number of control/ data packets generated to the total packets generated for varying burst sizes. The number of control and data packets generated are almost equal. This infers that the random number implemented for the design is working to the expected level.

5.1.6 Inference

As described under Figure 4.7, the states are governed by the timeouts which decide when the transition to different states occurs.

In this design, the time out are based on the IEEE 803.az standard.

Sleep time $T_s = 2.88\mu s$
Low power idle time $T_{lp} = 39.68\mu s$
Wake time $T_w = 4.48\mu s$

So, if the ONU requires to complete one sleep request, it has to go through Sleep, Low power and Wake states. This implies that each sleep request from the OLT results in the ONU sleeping for the low power idle time and the overhead time (sleep time + wake up time).

In this design,

Total sleep time = Actual sleep time + Overhead associated with transitions
= Low power idle time + Sleep time + Wakeup time
$= 39.68\mu s + 2.88\mu s + 4.48\mu s$
$= 47.04\mu s$

So the overall percentage of the actual sleep time in a sleep request
= (Low power idle time / Total sleep time)*100
$= (39.68\mu s/47.04\mu s) * 100$
$= 84.35\%$

Similarly the overall percentage of the sleep time and the wake time is 6.1% and 9.5% respectively. The results shown Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5 indicates the percentage of time spent in Low Power Idle state to be higher than the percentage of time spent in Sleep, Active and Wake state. This is primarily because of the timeouts provided in the design. The timeouts force the ONU to remain in a state until the timers (modeled using counters) expire. Since the low power time occupies a large percentage of the time, hence the higher energy savings.

The above results confirms that the designed FSM switches successfully between the states: Active, Sleep, Low Power Idle and Wake. Through the results, the functionality of the timeouts used in the design for different states has also been verified. The results obtained through implementation comply with the evaluated percentage of time spent in various states through timeout.

Analytical evaluation of energy savings

The following section provides an analytical evaluation for the amount of energy savings that can be achieved for this implementation:

P_a : Power consumed by the ONU during Active state
P_{lpi} : Power consumed by the ONU during Low Power Idle state
T_a : Time spent in Active state
T_s : Time spent in Sleep state
T_{lpi} : Time spent in Low Power Idle state
T_w : Time spent in wake state

Let ONU_{time} refer to the full cycle time which includes the time spent in all the four states (Active, Sleep, Low Power Idle, Wake).

$$ONU_{cycletime} = \Sigma ONU_{timespentinindividualstates} = T_a + T_s + T_{lpi} + T_w$$

Let the energy consumed by the ONU with sleep be denoted by E_s and the one without Sleep technique be E_{ws}.

When the sleep mode is not implemented, the power consumed by the ONU per unit time (P_a) is the power consumed during the Active state and is the same for all the states. So the resulting energy consumption of the ONU can be calculated from the following formula.

$$E_{ws} = P_a * (T_a + T_s + T_{lpi} + T_w)$$

The energy consumed by the ONU when it is has the Sleep technique implemented in it, is given by the following equation.

$$E_{ws} = P_a * (T_a + T_s + T_w) + P_{lpi} * T_{lpi}$$

Where: P_a is the power consumed by the ONU per unit time during the Active, Sleep and Wake state. P_s Is the power power consumed per unit time by the ONU in the Low Power Idle state.

So now we can calculate the energy savings by the following formula.

Energy Savings =$((E_a - E_b)/E_a) * 100$

Energy Savings =$((P_a - P_{lpi}) * Tl_{pi}/(P_a * (T_a + T_s + T_{lpi} + T_w))) * 100$

Now let us consider that the power consumed by the ONU during the Active state is ten times the power consumed during Low Power Idle state (i.e., $Pa = 10 * P_{lpi}$). Hence substituting this in the above equation, the resulting Energy Savings is given by,

Energy Savings =$(0.9 * Tl_{pi}/(T_a + T_s + T_{lpi} + T_w)) * 100$

By substituting the values of the $T_a = 1\mu s$; $T_s = 2.88\mu s$; $T_{lpi} = 39.68\mu s$; $T_w = 4.48\mu s$, the resulting Energy Savings is around 75%.

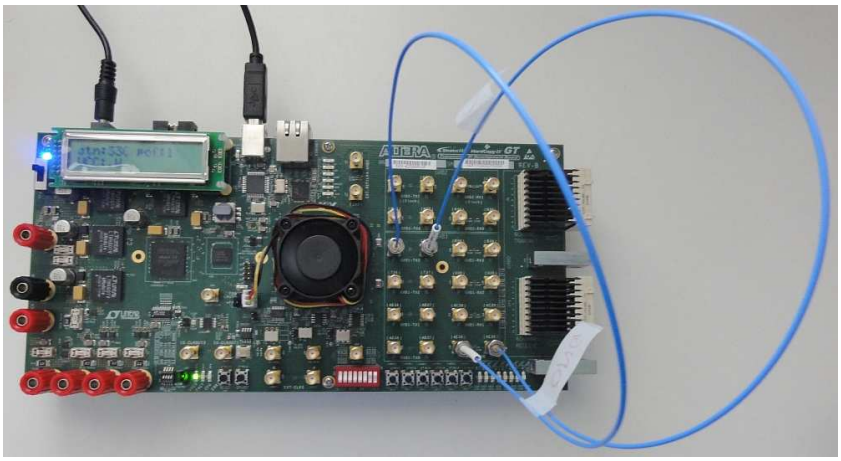

Figure 5.11: Local loopback.

5.2 Local Loopback

Local loopback is an other mode that is available in all transceiver configurations. In this mode the physical coding sublayer (PCS) and physical media attachment (PMA) blocks in the transmitter and receiver channels. Hence the data is allowed to pass through the transceivers at the transmitter and the receiver channel though the coaxial cables as shown below in the Figure 5.11. Since the design utilizes the transceivers supporting high data rates, the low voltage differential signaling (LVDS) standard is used to reduce errors. A more detailed information about the loopbacks can be found in [29]. The test is performed and the results are extracted through the TCL command line interface provided in System Console.

5.2.1 Experimental setup

The following description explains how the local loopback (Tx-to-Rx) was performed. The system can be configured to send 1000 packets with one iteration having a data length of 1500 bytes each. The procedure for running the tests for the local loopback are similar to the internal loopback. Numerous iterations are run for the different packet lengths and bursts. For configuring the local loopback mode and running the test, the following steps were followed:

(1) The FPGA is programmed with the .sof file corresponding to our design.

(2) The transceiver tool kit is opened and the TCL transcript window is used.

(3) The path corresponding to the source TCL files is provided using the following command.
cd <path>

(4) The specific test file is sourced using the following command.
source <testfile.tcl>

(5) The local loopback required is chosen from the TCL interface by typing in the command line.

(6) To run the local loopback test (and start both the Packet Monitor and the Packet Generator) the following command is typed in the command prompt.
TEST LOCALLB <BURST SIZE> <ITERATION>

The tests can be configured to run for different packet sizes by changing the packet size variable in the parameter.tcl, which is found inside the folder containing all the script files.

5.2.2 Results

The results for the local loopback are exactly similar to those obtained for the internal loopback. The experiments were repeated and found that all the parameters taken into account resembled the results of the internal loopback.

5.2.3 Inference

From the results it can be observed that the considered parameters are not varying and resemble the results of the internal loopback. This implies that the design is working fine at different loopback points, so it can be inferred that the data that is transmitted by the transceivers (through LVDS) from the OLT is received at the ONU through the coaxial cables without any errors.

5.3 Design implementation with exponential inter-arrival time

In this section the results of the exponential inter-arrival time implementation has been presented. As shown in Figure 4.5, the implementation utilizes a read only memory (ROM) initialized with memory information file (also refereed to as .mif) which stores the exponential inter-arrival times in

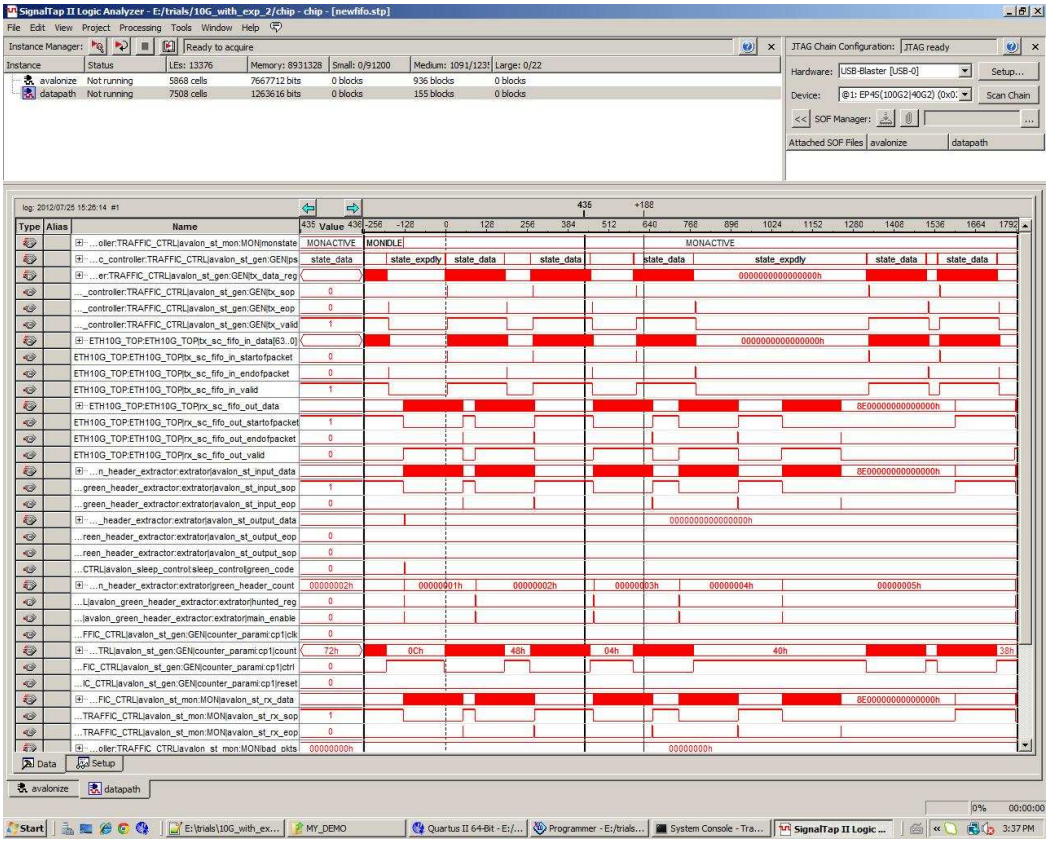

Figure 5.12: Representation of implemented exponential distribution using SignalTap.

number of cycles. This ROM file is then instantiated in the traffic generator module, with some significant modification to accommodate the new state and its functionalities.

Figure 5.12 shows the result obtained for the exponential inter-arrival time implementation through Signal Tap II logic analyzer tool in Quartus 2.

It can be seen that the above design has two Signal Tap instances. The "avalonize" is used to check the general working of the state machine by tapping the Sleep controller and associated module and some external signals. The "datapath" instance is used to check the flow of data through various modules at the same instant of time. Through this instance, implementation of the exponential inter-arrival time in the design can be verified.

It can be observed that at the transmitter side, the implementation works perfectly. Each packet lasts for $1.26\mu s$ with a clear definition of the Start of Packet (SOP) and the End of Packet (EOP). The structure of the packet is consistent in the Transmitter FIFO, but as it goes through the lower layers of the Ethernet subsystem there are some inconsistencies observed. One

of them is the SOP signal extends for the inter-arrival time and some minor inconsistencies with the data. This requires further analysis as this can be due to incompatibility of the design to adapt to modification at lower levels, even though modifications are allowed at the higher levels of abstraction.

Sampling frequency must be optimized to have the best results in Signal Tap. But this has to be chosen on what is the result we are interested to view as different modules work at different clock speeds.

Chapter 6

Conclusions

Exponential increase in the bandwidth requirements along with power consumption in telecommunication networks is posing a great challenge to the industry for which they need to come up with innovative solution to tackle these issues. These pressing issues have triggered wide range of research activities in both, industry and academia, to explore different approaches for reducing energy consumption. Lower energy consumption in access networks is of prime importance mainly because the subscriber base is expected to increase rapidly, and also as their demand for more bandwidth is ever increasing. The main goal of this thesis was to find and implement a solution to reduce the energy consumption in PONs.

This thesis proposed and implemented a sleep mode technique in the data link layer for Ethernet PONs in Altera Stratix 4GT FPGA. The sleep technique is based on the FSM implementation written in Verilog HDL. The FSM switches the ONU "on" and "off" based on the sleep request encoded explicitly through out green header implementation in the frames generated at the OLT. The time spent by the ONU in the low power state (where least energy is consumed as the ONU sleeps), sleep and wake transitions are all based on the values defined as per the IEEE 803.az standard.

The analyses of the results show that the implemented sleep mode technique provides substantial energy savings. The implementation was performed on the testbed setup in two different modes: local loopback and internal loopback. Similar results were obtained for both the configurations. For the implementation with standard values as defined in IEEE 803.az, a sleep efficiency of 75% is achieved when the power during active state is ten times the power at low power state. Additionally, a negative exponential interarrival module was developed. This is integrated in the generator to emulate the inter packet arrival times. So that packet arrivals can be modeled with various statistical distributions.

6.0.1 Future scope

New designs can be implemented for various statistical distributions referring to our negative exponential module developed during this thesis. Buffers will be of prime importance in the future designs mainly because:

- For the ONU to have low power consumption, it has to remain in the sleep mode as long as possible. The longer the better. But for guaranteed quality of service (QOS), the buffers should be large enough to store all the packets which arrive even when the ONU is sleeping.
- They buffers should be intelligently managed both in terms of buffer size and the number of buffers. This in itself is an interesting task which requires buffer optimization mainly due to the limited resources available in the FPGA.
- In the real time scenario, implementation of the buffer(s) would be inevitable because:
  - Numbers of users are ever increasing.
  - Traffic is increasing.
  - Bandwidth requirements are rising.
  - QOS has to be guaranteed.
- Taking all the above into considerations, a dynamic sleep protocol is required. This should not only manage the buffers, but also should synchronize with both the OLT and ONU.

Bibliography

[1] C. Lange, D. Kosiankowski, R. Weidmann, and A. Gladisch, "Energy consumption of telecommunication networks and related improvement options," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, pp. 285 –295, march-april 2011.

[2] R. Tucker, "Green optical communications—Part II: Energy limitations in networks," *Selected Topics in Quantum Electronics, IEEE Journal of*, vol. 17, pp. 261 –274, march-april 2011.

[3] S.-W. Wong, L. Valcarenghi, S.-H. Yen, D. R. Campelo, S. Yamashita, and L. Kazovsky, "Sleep mode for energy saving PONs: Advantages and Drawbacks," in *GLOBECOM Workshops, 2009 IEEE*, pp. 1–6, 2009.

[4] B. Skubic and D. Hood, "Evaluation of ONU power saving modes for gigabit-capable passive optical networks," *Network, IEEE*, vol. 25, pp. 20–24, march-april 2011.

[5] L. Shi, B. Mukherjee, and S.-S. Lee, "Energy-efficient PON with sleep-mode ONU: progress, challenges, and solutions," *Network, IEEE*, vol. 26, pp. 36 –41, march-april 2012.

[6] R. Kubo, J.-i. Kani, H. Ujikawa, T. Sakamoto, Y. Fujimoto, N. Yoshimoto, and H. Hadama, "Study and demonstration of sleep and adaptive link rate control mechanisms for energy efficient 10G-EPON," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 2, pp. 716 –729, september 2010.

[7] M. Fiammengo, A. Lindström, P. Monti, L. Wosinska, and B. Skubic, "Experimental evaluation of cyclic sleep with adaptable sleep period length for PON," in *37th European Conference and Exposition on Optical Communications*, p. We.8.C.3, Optical Society of America, 2011.

[8] X. Jin and J. Tang, "First experimental demonstration of real-time optical OFDMA PONs with colorless ONUs and adaptive DBA," in *Optical Fiber Communication Conference and Exposition (OFC/NFOEC), 2012 and the National Fiber Optic Engineers Conference*, pp. 1 –3, march 2012.

[9] R. Scott, W. Cong, V. Hernandez, K. Li, B. Kolner, J. Heritage, and S. Yoo, "An eight-user time-slotted spects O-CDMA testbed: demonstration and simulations," *Lightwave Technology, Journal of*, vol. 23, pp. 3232 – 3240, oct. 2005.

[10] L. G. Kazovsky, N. Cheng, W.-T. Shaw, D. Gutierrez, and S.-W. Wong, *Broadband Optical Access Networks: Emerging Technologies and Optical-Wireless Convergence*. Hoboken, New Jersey: John Wiley & Sons, 2011.

[11] Xilinx, "What is a fpga?." `www.xilinx.com/fpga/index.htm`.

[12] M. Marsan, A. Anta, V. Mancuso, B. Rengarajan, P. Vasallo, and G. Rizzo, "A simple analytical model for energy efficient ethernet," *Communications Letters, IEEE*, vol. 15, pp. 773 –775, july 2011.

[13] "IEEE standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks–specific requirements part 3: Carrier sense multiple access with collision detection (csma/cd) access method and physical layer specifications amendment 5: Media access control parameters, physical layers, and management parameters for energy-efficient ethernet," *IEEE Std 802.3az-2010 (Amendment to IEEE Std 802.3-2008)*, pp. 1–302, 27 2010.

[14] "GPON power conservation." ITU-T G-series Recommendations - Supplement 45 (G.sup45), 2009.

[15] E. Trojer and P. Eriksson, "Power saving modes for GPON and VDSL," *in Proc. 13th Euro. Conf. Netw. and Optical Commun. (NOC), Austria*, June 30-July 3 2008.

[16] "Code of conduct on energy consumption of broadband equipment," *European Commission Directorate General Joint Research Center (JRC) Institute for Energy, Renewable Energy Unit*, February 10 2011.

[17] K. Christensen, P. Reviriego, B. Nordman, M. Bennett, M. Mostowfi, and J. Maestro, "IEEE 802.3az: the road to energy efficient ethernet," *Communications Magazine, IEEE*, vol. 48, pp. 50 –56, november 2010.

[18] S.-H. Yen, S.-W. Wong, S. Das, N. Cheng, J. Cho, S. Yamashita, O. Solgaard, and L. Kazovsky, "Photonic components for future fiber access networks," *Selected Areas in Communications, IEEE Journal on*, vol. 28, pp. 928 –935, aug. 2010.

[19] L. Valcarenghi, M. Presi, G. Contestabile, P. Castoldi, and E. Ciaramella, "Impact of modulation formats on ONU energy saving," in *Optical Communication (ECOC), 2010 36th European Conference and Exhibition on*, pp. 1 –3, sept. 2010.

[20] W. Freude, M. Roeger, G. Boettger, M. Dreschmann, M. Huebner, C. Klamouris, A. Bett, J. Becker, and J. Leuthold, "Optically powered fiber networks," *Opt. Express*, vol. 16, pp. 21821–21834, Dec 2008.

[21] J. Mandin, "Epon power saving via sleep mode," *IEEE P802.3av 10G-EPON Task Force Meeting*, September 2008.

[22] J. Mandin and L. Khermosh, "Power saving in IEEE 802-style networks," *US Patent Application*, vol. Us201000118753a1, May 13 2010.

[23] PMC-Sierra, "Second generation GPON ONT System-on-a-Chip (SoC)." `http://www.pmc-sierra.com/products/details/pas7401.htm`.

[24] J. Zhang and N. Ansari, "Toward energy-efficient 1g-epon and 10g-epon with sleep-aware mac control and scheduling," *Communications Magazine, IEEE*, vol. 49, pp. s33 –s38, february 2011.

[25] Altera, "10-gbps ethernet mac megacore function." `http://www.altera.com/products/ip/iup/ethernet/m-alt-10gbps-ethernet-mac.html`.

[26] Altera, "10gbase-r phy megacore function." `http://www.altera.com/products/ip/iup/ethernet/m-alt-10gbase-r-pcs.html`.

[27] Altera, "Stratix iv gt edition reference manual." `http://www.altera.com/literature/manual/rm_sivgt_si_dev_board.pdf`.

[28] A. M. Law, *Simulation Modeling and Analysis*. New York: McGrawHill International, 2011.

[29] Altera, "Transceiver architecture in stratix iv devices." `http://www.altera.com/literature/hb/stratix-iv/stx4_siv52001.pdf`.

List of Publications

- Neelakandan Manihatty Bojan, Dung Pham Van, Luca Valcarenghi, and Piero Castoldi, "An Energy Efficient ONU Implementation", Sustain IT 2012: The Second IFIP Conference on Sustainable Internet and ICT for Sustainability, October 4-5, 2012 - Pisa, Italy.

Appendix

All the source code for the design implemented is located on JEAN1 server under the folder E:\Neels\Source_code.