Winter 2006

Design and implementation of a 10 Gigabit Ethernet XAUI test systems

Meghana Reddy Kundoor
University of New Hampshire, Durham

DESIGN AND IMPLEMENTATION OF A 10 GIGABIT ETHERNET XAUI TEST SYSTEMS

BY

MEGHANA REDDY KUNDOOR

B.S.E.E. Osmania University, 2003

THESIS

Submitted to the University of New Hampshire

in Partial Fulfillment of

the Requirements for the Degree of

Master of Science

in

Electrical Engineering

December, 2006

This thesis has been examined and approved.

[illegible]

Thesis Advisor, Dr. Michael J. Carter, Associate Professor

[illegible]

Dr. Kuan Zhou, Assistant Professor

[illegible]

Dr. William H. Lenharth, Research Associate Professor

December 15, 2006

Date

ACKNOWLEDGEMENTS

First of all, I would like to thank my thesis adviser, Dr. Michael J. Carter for the guidance that he has given me during my thesis. Without Professor Carter's support and patience, this thesis work would not have been possible. I am grateful to Professor Kuan Zhou and Professor William H. Lenharth for serving on my committee.

I am also obliged to Mr. Bob Noseworthy, Technical Manager, IOL for his generous help in my research and career pursuit. I thank IOL staff members Mr. Dave Estes and Mr. Matt Plante for their insightful suggestions that greatly helped improve the quality of my thesis work.

Finally, I would like to thank my family. My parents have been a continuous source of support throughout my life, even when they are thousands of miles away. My husband Sriharsha, accompanied me throughout the course of my graduate study. There are no words to express my gratitude to him for all his support and care.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

DESIGN AND IMPLEMENTATION OF 10 A 10 GIGABIT ETHERNET XAUI TEST SYSTEM

by

Meghana R. Kundoor

University of New Hampshire, December, 2006

10 Gigabit Ethernet has been standardized (IEEE 802.3ae), and products based on this standard are being deployed to interconnect MANs, WANs, Storage Area Networks, and very high speed LANs. The XAUI portion of the standard is primarily concerned with short range (up to 50 cm) chip-to-chip communication across printed circuit board traces. The UNH-IOL 10 Gigabit Ethernet Consortium, an industry-supported organization, performs PHY layer testing on products using a test system that has been partially implemented on a Xilinx ML321 evaluation board using the Virtex II-Pro FPGA.

A new implementation of the 10 Gigabit Ethernet XAUI test system on the existing ML321 evaluation board is presented in this thesis. The new design removes a number of limitations present in the original Xilinx test system, and it adds new features to the existing transmit and receive sub-systems that enable test engineers to expand the range of test cases and analyze them while simultaneously increasing the speed of testing. The new test system also eliminates the need for expensive test instruments.

CHAPTER 1

INTRODUCTION

From its origin more than 25 years ago, Ethernet has evolved to meet the increasing demands of packet-based networks. Due to the low implementation cost, reliability, and relative simplicity of installation and maintenance, Ethernet's popularity has grown rapidly. Further, as the demand for ever-faster network speeds has increased, Ethernet has been adapted to handle these higher speeds. The IEEE 802.3 standard defines Ethernet at the physical and data link layers of the OSI network model. Today, virtually all companies, institutions, and other organizations are wired for these 10/100-Mbit/s versions of Ethernet. However, the world of Ethernet is constantly evolving and many organizations are looking to the benefits of 10 Gigabit Ethernet to support higher speeds across the entire network. Simply put, with its interoperability and scalability, 10 Gigabit Ethernet is an ideal solution for organizations with growing bandwidth needs. 10 Gigabit Ethernet provides organizations with a cost effective solution to increase productivity and to deliver new services. The primarily applications for this new technology will be found in data centers, high-end workstations, and web-enabling applications.

The 10 Gigabit Ethernet standard extends the IEEE 802.3ae standard protocols to a wire speed of 10 Gbps and expands the Ethernet application space to include WAN-compatible links. The 10 Gigabit Ethernet standard provides a significant increase in bandwidth while maintaining maximum compatibility with the installed base of 802.3 standard interfaces, protects previous investment in research and development, and retains the existing principles of network operation and management.

More background information about Ethernet and 10 Gigabit Ethernet is presented in Chapter 2.

1.1 Aim

The aim of this thesis was to improve the existing 10 Gigabit Ethernet XAUI test system, which is used for testing 10 Gigabit Ethernet devices at the UNH Interoperability Lab (IOL). Although the current system has enabled successful testing of vendor products, it requires several expensive test instruments in addition to a high speed FPGA-based platform. The FPGA portion of the test system is not easily reprogrammed to facilitate the addition of new test sequences or the use of triggered selection of special test sequences based on observed responses of the Device Under Test (DUT). There was also at one time the desire by the UNH IOL to license its 10 Gigabit Ethernet Test System to industry partners, and this goal motivated re-design of the test system to eliminate key elements of intellectual property not owned by UNH. Although this goal was subsequently dropped by IOL management, the need to improve the reliability, flexibility of re-programming, and automated test execution speed of the original test system remained. These objectives have been largely accomplished through the re-design of the FPGA portion of the test system while continuing to utilize the same hardware platform on which the current test system is implemented.

1.2 Interoperability Lab (IOL)

The University of New Hampshire Interoperability Laboratory (UNH-IOL) tests networking and data communications products. Since 1988, the laboratory has fostered multi-vendor interoperability while preparing UNH students for careers in the industry. The UNH-IOL offers collaborative testing programs in over 20 data networking and storage technologies. The 10 Gigabit Ethernet group is one such consortium. More information about the lab can be found at www.iol.unh.edu.

1.3 Organization of Thesis

The second chapter gives an introduction to Ethernet and the development of 10 Gigabit Ethernet. The objectives of 10 Gigabit Ethernet and the various implementations within the standard are also discussed in this chapter.

The third chapter discusses in detail the 10 Gigabit Ethernet Attachment Unit Interface (XAUI). This chapter also covers the different features of Xilinx's incorporated embedded PowerPC processor and 3.125Gbps RocketIO serial transceivers that were used for the development of the test system.

The fourth and fifth chapter discusses the implementation of the Transmit and Receive process. These chapters introduce the reader to the implementation details of the design. The chapters also present the implementation issues observed in the design. Finally, suggestions for future work are provided.

CHAPTER 2

BACKGROUND

This chapter gives an introduction to Ethernet and the development of 10 Gigabit Ethernet. It also discusses the various interfaces of the 10 Gigabit Ethernet standard.

2.1 Ethernet

The first experimental Ethernet system was developed in the early 1970s by Bob Metcalfe and David Boggs of the Xerox Palo Alto Research Center (PARC). It interconnected Xerox Alto computers and laser printers at a data transmission rate of 2.94 Mb/s. This data rate was chosen because it was derived from the system clock of the Alto computer. In the 1980's a group of vendors standardized the Ethernet system, which was known as the DEC-Intel-Xerox (DIX) standard. Ethernet was subsequently adopted by the Institute of Electrical and Electronics Engineers as a formal standard (IEEE 802.3). The first Ethernet standard was published in 1985 and was known as the "IEEE 802.3 Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method and Physical Layer Specifications". Its star-topology, twisted pair wiring form became the most widespread LAN technology in use from the 1990s to the present, largely replacing competing LAN standards such as coaxial cable Ethernet, token ring, FDDI, and ARCNET. Since then, the Ethernet System has evolved from 10 Mbps to 100 Mbps (Fast Ethernet) to 1000 Mbps (Gigabit Ethernet) and to the latest 10,000 Mbps 10 Gigabit

Ethernet). Ethernet has evolved from a traditional LAN technology to a LAN/MAN/WAN technology and remains one of the most popular networking standards.

2.2 10 Gigabit Ethernet Background

Gigabit Ethernet, a transmission technology based on the Ethernet frame format and protocol used in local area networks (LANs), provides a data rate of 1 billion bits per second (one gigabit). Gigabit Ethernet is defined in the IEEE 802.3 standard and is currently being used as the backbone in many enterprise networks. Gigabit Ethernet is carried primarily on optical fiber (with very short distances possible on copper media). Existing Ethernet LANs with 10 and 100 Mbps cards can feed into a Gigabit Ethernet backbone. An alternative technology that competes with Gigabit Ethernet is ATM. Gigabit Ethernet has lower cost of ownership when compared to alternative technologies like SONET or ATM. As the demand for high-speed networks continues to grow, the need for a faster Ethernet technology is apparent. In March 1999, a working group was formed by the IEEE 802.3 Higher Speed Study Group (HSSG) to develop a standard for 10-Gigabit Ethernet.

2.3 10 Gigabit Ethernet Standard

The 10 Gigabit Ethernet standard extends the IEEE 802.3ae standard protocols to a wire speed of 10 Gbps and expands the high speed Ethernet application space to include WAN-compatible links. The 10 Gigabit Ethernet standard provides a significant increase in bandwidth while maintaining maximum compatibility with the installed base of 802.3 standard interfaces, protects previous investment in research and development, and retains the existing principles of network operation and management. Under the Open Systems Interconnection (OSI) model, Ethernet is fundamentally a Layer 1 and 2 protocol. 10 Gigabit Ethernet retains key Ethernet architectural features, including the Medium Access Control (MAC) protocol, the Ethernet frame format, and the minimum and maximum frame size. Just as Gigabit Ethernet, both 1000BASE-X and 1000BASE-T, followed the standard Ethernet model, 10 Gigabit Ethernet continues the evolution of Ethernet in speed and distance, while retaining the same Ethernet architecture used in other Ethernet specifications.

The IEEE 802.3ae Task Force, the standards body governing the 10 Gigabit Ethernet standard, has set forth the following five criteria that the standard should meet:

a) Broad Market Potential: The standard must have broad market potential with multiple vendors and customers supporting it.

b) Compatibility: The standard must be compatible with IEEE Standard 802.3 and it should conform to the full-duplex operating mode of the 802.3 MAC (Media Access Control), suitably adapted for 10 Gb/s operation. Half duplex operation is not be supported in this standard. Hence there is no CSMA/CD.

c) Distinct Identity: The standard must be substantially different from other 802.3 specifications/solutions and it should have unique solutions for problems.

d) Technical Feasibility: The standard must have demonstrated technical feasibility so that it's a proven technology to instill confidence in its reliability.

e) Economic Feasibility: The standard must be economically viable so that the cost factor (Installation and Management costs) is reasonable for its performance.

In addition, the standards body had proposed the following requirements:

a) Two families of PHY layer specifications – a LAN PHY operating at 10 Gb/s, and a WAN PHY operating at a data rate compatible with STS-192c/SDH VC-4-64c (9.95328 Gb/s).

b) A mechanism to adapt the MAC data rate to the WAN data rate.

c) Various Physical layer specifications that support link distances of

At least 300m over installed MMF (Multi Mode Fiber).

At least 65m over MMF.

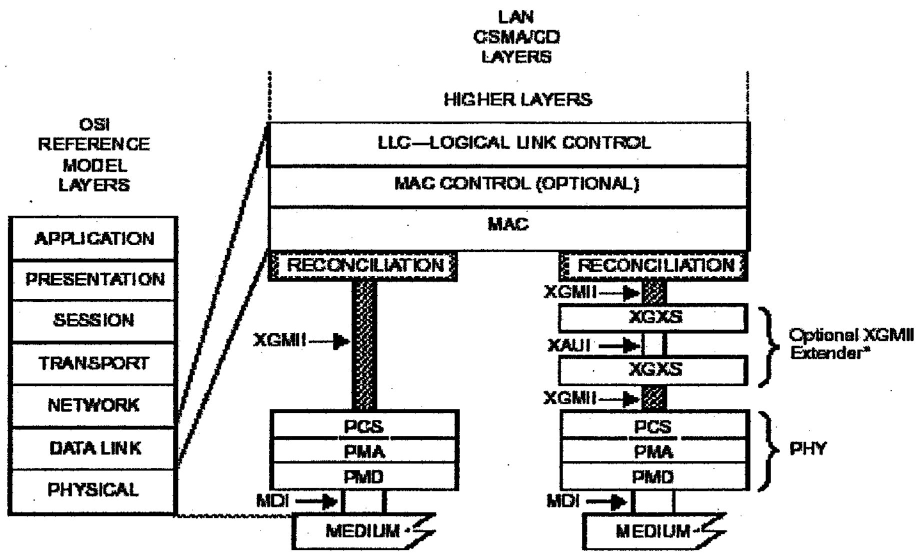
At least 2 Km over SMF (Single Mode Fiber).

At least 10 Km over SMF.

At least 40 Km over SMF.

2.4 10 Gigabit Ethernet Architecture

At the physical layer (Layer 1), an Ethernet physical layer device (PHY) connects the optical or copper media to the MAC layer through a connectivity technology. Ethernet architecture further divides the physical layer into three sublayers: Physical Medium Dependent (PMD), Physical Medium Attachment (PMA), and Physical Coding Sublayer (PCS). PMDs provide the physical connection and signaling to the medium. The PCS performs encoding and decoding (e.g., 8B/10B) of the data streams to and from the MAC layer .The PMA serializes multi-bit code groups into a single bit stream and vice versa. The IEEE 802.3ae standard defines two PHY types: the LAN PHY and the WAN PHY. They provide the same functionality, except the WAN PHY has an extended feature in the PCS, that enables the SONET connection.



MAC = MEDIA ACCESS CONTROL
MDI = MEDIUM DEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PHY = PHYSICAL LAYER DEVICE
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT
XAUI = 10 GIGABIT ATTACHMENT UNIT INTERFACE
XGMII = 10 GIGABIT MEDIA INDEPENDENT INTERFACE
XGXS = XGMII EXTENDER SUBLAYER

Figure 2.1: 10 Gigabit Ethernet Architecture (Courtesy: IEEE 802.3ae Standard)

Figure 2.1 shows the 10 Gigabit Ethernet system as referenced to the seven-layered OSI model. This system consists of the following layers:

2.4.1 MAC (Media Access Control)

The MAC performs the important function of framing the data bytes coming from the upper layers. The MAC adds the CRC (Cyclic Redundancy Check) and sends the frame to the PHY layer. 10 Gigabit Ethernet uses the same frame format as that of 802.3 Ethernet frames to maintain compatibility.

Preamble	SFD	Destination MAC Address	Source MAC Address	Length or Type	VLAN Tag	MAC Data	Pad	FCS

Figure 2.2: Ethernet Frame Format

The Ethernet frame format is shown in Figure 2.2.The respective fields are defined as follows:

a) Preamble The Preamble pattern is seven bytes in length and each byte of the preamble is 10101010. It serves to give components in the network time to detect the presence of a signal and acquire data clock synchronization.

b) Start Frame Delimiter A sequence of 8 bits having the bit configuration 10101011 that indicates the start of the frame.

c) Destination & Source MAC Addresses: The Destination MAC Address field identifies the station or stations that are to receive the frame. The Source MAC Address identifies the station that originated the frame. These addresses are 6 bytes in length. A Destination Address may specify either an "individual address" destined for a single station, or a "multicast address" destined for a group of stations. A Destination Address of all 1's refers to all stations on the LAN and is called a "broadcast address". The source address identifies the source of the frame.

d) Length/Type: The length field is 2 bytes in length and this identifies the type and number of data bytes in the frame. If the value of this field is less than or equal to 1500, then the Length/Type field indicates the number of bytes in the subsequent MAC Client Data field. If the value of this field is greater than or equal to 1536, then the Length/Type field indicates the nature of the MAC client protocol (protocol type).

e) Data: This field contains the data transferred from the source station to the destination station or stations. The maximum size of this field is 1500 bytes. If the size of this field is less than 46 bytes, then use of the subsequent "Pad" field is necessary to bring the frame size up to the minimum length of 64. This field contains the actual payload, and the other fields are an overhead introduced by the Ethernet framing structure.

f) Pad: Extra data bytes are appended in this field to bring the frame length up to its minimum size. A minimum Ethernet frame size is 64 bytes.

Frame Check Sequence:

g) Frame Check Sequence (FCS): This field contains a 4-byte cyclical redundancy check (CRC) value THAT is used for error checking. When a source station assembles a MAC frame, it performs a CRC calculation on all the bits in the frame from the Destination MAC Address through the Pad fields (that is, all fields except the preamble, start frame delimiter, and frame check sequence). The source station stores the value in this field and transmits it as part of the frame. When the frame is received by the destination station, it performs an identical check. If the calculated value does not match the value in this field, the destination station assumes an error has occurred during transmission and discards the frame.

h) Inter-Frame Gap (IFG): Ethernet devices must allow a minimum idle period between transmission of frames known as the interframe gap (IFG) or interpacket gap (IPG). It provides a brief recovery time between frames to allow devices to prepare for reception of the next frame. The minimum interframe gap is 96 bit times, which is 9.6 nanoseconds for 10 Gbits/sec Ethernet.

i)VLAN Tagging :In 1998, the IEEE approved the 802.3ac standard that defines frame format extensions to support Virtual Local Area Network (VLAN) Tagging on Ethernet networks. The VLAN protocol permits insertion of an identifier, or "tag", into the Ethernet frame format to identify the VLAN to which the frame belongs. This provides various benefits such as easing network administration, allowing formation of work groups, enhancing network security, and providing a means of limiting broadcast domains. The 4-byte VLAN tag is inserted into the Ethernet frame between the Source MAC Address field and the Length/Type field. The first 2-bytes of the VLAN tag consist of the "802.1Q Tag Type" and are always set to a value of 0x8100. The 0x8100 value is actually a reserved Length/Type field assignment that indicates the presence of the VLAN tag, and signals that the traditional Length/Type field can be found at an offset of 4-bytes further into the frame. The last 2-bytes of the VLAN tag contain the User Priority Field that is used to assign a priority level to the Ethernet frame. The Canonical Format Indicator (CFI) is used in Ethernet frames to indicate the presence of a Routing Information Field (RIF) and VLAN Identifier (VID) which uniquely identifies the VLAN to which the Ethernet frame it belongs.

The original Ethernet standards defined the minimum frame size as 64-bytes and the maximum as 1518-bytes. These numbers include all bytes from the Destination MAC Address field through the Frame Check Sequence field. The Preamble and Start Frame Delimiter fields are not included when quoting the size of a frame. The IEEE 802.3ac standard released in 1998 extended the maximum allowable frame size to 1522-bytes to allow a "VLAN tag" to be inserted into the Ethernet frame format.

2.4.2 RS (Reconciliation Sublayer)

The 10 Gigabit Media Independent Interface (Clause 46) provides an interconnection between the Media Access Control (MAC) sublayer and Physical Layer entities (PHY). This XGMII supports 10 Gb/s operation through its 32-bit-wide transmit and receive data paths. The Reconciliation Sublayer provides a mapping between the signals provided at the XGMII and the MAC/PLS service definition. The RS converts the MAC data stream into the parallel data paths of XGMII (10 Gigabit Media Independent Interface) and vice-versa. The RS also maps the signals between XGMII and the MAC.

2.4.3 XGMII (10 Gigabit Media Independent Interface)

The XGMII provides the interface between the RS and the PHY layers. The XGMII is divided into four lanes with each lane comprising 8 bits of data (TXD), 1 bit control (TXC), and a common clock signal (TCLK) for all the lanes. The control line determines if the information being sent is a special character or ordinary data. If TXC is 1, then the information conveyed by the TXD line is a special character. If TXC is 0, then the information conveyed by the TXD line is data. The XGMII has RXD, RXC, and RCLK signals on the receive side similar to the TXD, TXC and TCLK signals on the transmit side. Table 2.1 shows the Transmit and Receive Lane associations.

Lanes	TXD/RXD (Data)	TXC/RXC (Control)	Clock
0	<7:0>	Control Bit0	TCLK/RCLK
1	<15:8>	Control Bit1	TCLK/RCLK
2	<23:16>	Control Bit2	TCLK/RCLK
3	<31:24>	Control Bit3	TCLK/RCLK

Table 2.1: Transmit and Receive Lane Associations

The XGMII can be optionally extended using an XGMII extender known as XAUI (10 Gigabit Attachment Unit Interface). XAUI can be used to drive signals over longer circuit board trace lengths and it also reduces the number of required signal traces when compared to XGMII. The positioning of XAUI within the 10 Gigabit Ethernet architecture is shown in Figure 2.1. The XAUI architecture is explained in greater detail in the next chapter.

2.4.4 PCS (Physical Coding Sublayer)

The PCS performs the important jobs of encoding the XGMII characters and passing the encoded stream to the PMA (Physical Medium Attachment) on the transmit side, as well as decoding the PMA bit stream and passing the decoded output to the XGMII on the receive side. There are three types of PCS defined in the10 Gigabit Ethernet Architecture:

a) 8B/10B PCS: This form of PCS is referred to as the 10GBASE-X PCS, which provides services to the XGMII that include:

1) Encoding of 32 XGMII data bits and 4 XGMII control bits into four parallel lanes with each lane conveying 10-bit code groups.

2) Decoding of 10-bit code groups on four parallel PMA lanes into 8 XGMII data bits and 1 XGMII control bit per lane. Hence, this PCS is basically an 8 to 10-bit converter and vice-versa.

3) Conversion of the XGMII idle sequence into a pseudo-random pattern so as to reduce electromagnetic interference.

4) Synchronization of code groups on each lane to determine word boundaries.

5) De-skew of received code groups to eliminate the lane-to-lane skew problem.

6) Clock rate compensation to compensate for difference in clock rates.

Table 2.2 below shows the relationship between XGMII and PCS-Code Groups on the transmit side, and it also applies to the receive side except that transmission is replaced by reception.

TXC	TXD	PCS Code Group	Description
0	00-FF	/D/	Data Transmission
1	07	/K/	XGMII Idle
1	9C	/Q/	Sequence
1	FB	/S/	Start
1	FD	/T/	Terminate
1	FE	/E/	Error

Table 2.2: XGMII to PCS Code-Group Mapping

b) 64B/66B PCS: This form of PCS is referred to as the 10GBASE-R PCS. The services provided by this PCS include:

1) Encoding of 64 XGMII data bits into 66-bit PCS output, and this 66-bit output is transferred in 16-bit blocks to the PMA for transmission and decoding of 66-bit inputs from the PMA into 64-bit XGMII output.

2) Clock rate compensation between the MAC and PMD (Physical Medium Dependent) when connected to the WAN PMD through insertion or deletion of idle characters.

The 10GBASE-R PCS could be connected either to a LAN PMD or WAN PMD. The LAN PMD operates at 10 Gb/s whereas the WAN PMD operates at the SONET compatible rate of 9.95328 Gb/s. The 10GBASE-R PCS connects to the WAN PMD through the WIS (WAN Interface Sublayer) and serial PMA layer.

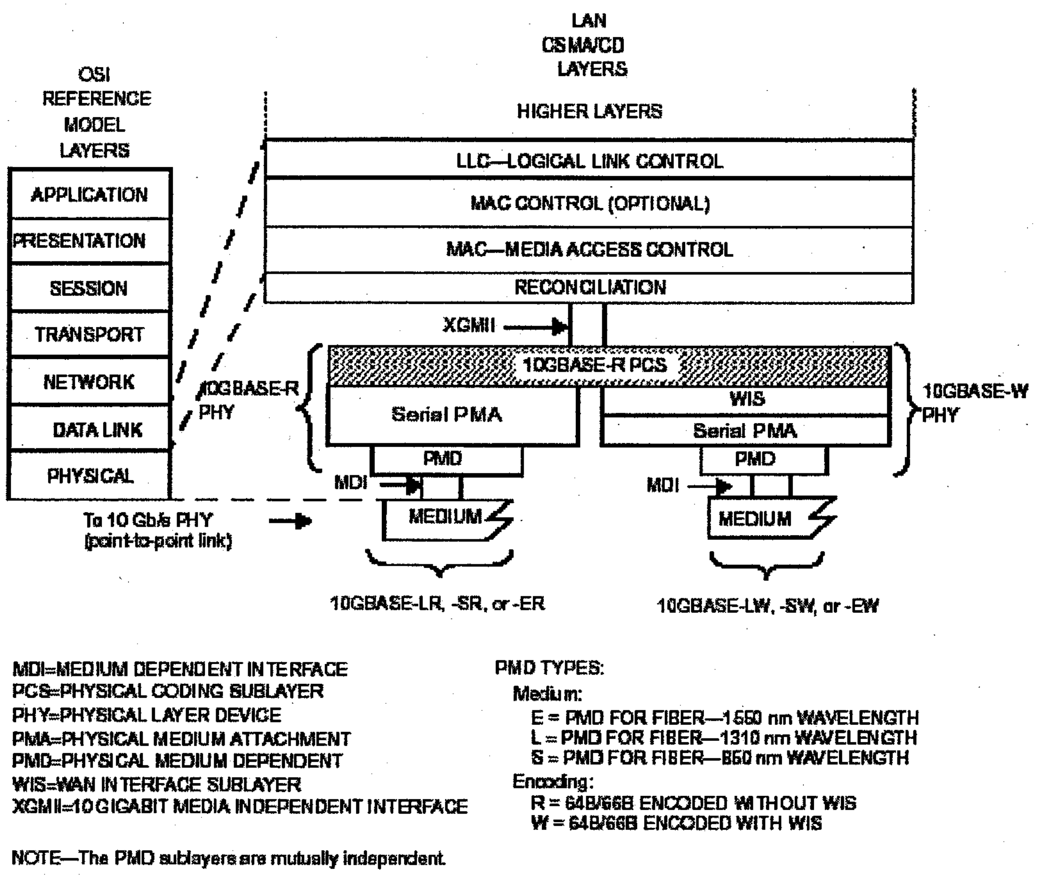

LAN CSMA/CD LAYERS
OSI REFERENCE MODEL LAYERS
HIGHER LAYERS
LLC—LOGICAL LINK CONTROL
MAC CONTROL (OPTIONAL)
MAC—MEDIA ACCESS CONTROL
RECONCILIATION
APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
XGMII
10GBASE-R PCS
10GBASE-R PHY
Serial PMA
WIS
Serial PMA
10GBASE-W PHY
PMD
PMD
MDI
MDI
MEDIUM
MEDIUM
To 10 Gb/s PHY (point-to-point link)
10GBASE-LR, -SR, or -ER
10GBASE-LW, -SW, or -EW
MDI=MEDIUM DEPENDENT INTERFACE
PCS=PHYSICAL CODING SUBLAYER
PHY=PHYSICAL LAYER DEVICE
PMA=PHYSICAL MEDIUM ATTACHMENT
PMD=PHYSICAL MEDIUM DEPENDENT
WIS=WAN INTERFACE SUBLAYER
XGMII=10 GIGABIT MEDIA INDEPENDENT INTERFACE
PMD TYPES:
Medium:
E = PMD FOR FIBER—1550 nm WAVELENGTH
L = PMD FOR FIBER—1310 nm WAVELENGTH
S = PMD FOR FIBER—850 nm WAVELENGTH
Encoding:
R = 64B/66B ENCODED WITHOUT WIS
W = 64B/66B ENCODED WITH WIS
NOTE—The PMD sublayers are mutually independent.

Figure 2.3: 10GBASE-R Implementations.
(Courtesy: IEEE 802.3ae Standard)

c) The term 10GBASE-W refers to a specific family of physical layer implementations based upon STS-192c/SDH VC-4-64c encapsulation of 64B/66B encoded data. The 10GBASE-W family of physical layer implementations is composed of 10GBASE-SW, 10GBASE-LW, and 10GBASE-EW. All 10GBASE-R and 10GBASE-W PHY devices share a common PCS specification (Clause 49). The 10GBASE-W PHY devices also require the use of the WAN Interface Sublayer, (WIS) (Clause 50). Specifications of each physical layer device are contained in Clause 52 and Clause 53 and Clause 54. The WIS provides a 10GBASE-W device with the capability to transmit and receive IEEE 802.3 MAC frames within the payload envelope of a SONET frame. Figure 2.3 shows the 10GBASE-W PCS architecture.

2.4.5 PMA (Physical Medium Attachment) Sublayer

The PMA sublayer provides a medium independent interface for the PCS to support a wide range of physical media. The PMA functions vary depending on the type of PCS, but in general the functions performed by the PMA include:

a) Mapping of data streams between the PCS and PMD and vice-versa, and between the WIS and the PMD layers for a 10GBASE-W implementation.

b) Serialization of parallel data streams from the PCS to be presented to the serial PMD, and deserialization of received PMD data streams to be presented to the PCS.

c) Clock recovery from the received data stream.

2.4.6 PMD (Physical Medium Dependent) Sublayer

The PMD sublayer connects the PMA to the medium, and it is a medium dependent interface. The PMD functions and types vary depending on the type of PCS and PMA, but in general the functions performed by the PMD include transmission and reception of the serial bit stream on the underlying physical medium and an optional loop-back function at the PMD service interface.

The PMD types vary depending on the type of physical medium used and these are shown in Table 2.3 along with the associated PCS versions.

PMD	Description	Associated PCS
10GBASE-SR	850nm serial LAN PHY	64/66B
10GBASE-LR	1310nm serial LAN PHY	64/66B
10GBASE-ER	1550nm serial LAN PHY	64/66B
10GBASE-SW	850nm serial WAN PHY	64/66B
10GBASE-LW	1310nm serial WAN PHY	64/66B
10GBASE-EW	1550nm serial WAN PHY	64/66B
10GBASE-LX4	1310nm parallel LAN PHY	8/10B

Table 2.3: Physical Media Dependent Types

In this chapter the background information about Ethernet, 10 Gigabit Ethernet and 10Gigabit Ethernet's interfaces were discussed. The 10 Gigabit Ethernet XAUI system architecture and Xilinx's Virtex 2 pro FF672 features will be presented in chapter3.

CHAPTER 3

XAUI SYSTEM ARCHITECTURE

In this chapter, the 10 Gigabit Ethernet XAUI system is reviewed. This chapter also includes the different features of Xilinx's embedded PowerPC processor and 3.125 Gbps RocketIO serial transceivers that were used for the development of the test system

The XAUI system consists of an XGXS (XGMII Extender Sublayer) at the RS end (DTE XGXS) and an XGXS at the PHY end (PHY XGXS) separated by a XGMII extender known as XAUI, the purpose of which is to extend the operational distance of the XGMII and also reduce the number of interface signals. The DTE XGXS acts as an 8B/10B encoder in one direction and as a decoder in the other direction, and the same is true of the PHY XGXS. The 8B/10B encoding/decoding functionality is specified in clause 48 of the IEEE 802.3ae standard-10GBASE-X PCS. The XGXS blocks are separated by XAUI data paths spanning up to approximately 50 cm. The XAUI interface is optional if implemented, should be implemented as a chip-to-chip interface with traces on a Printed Circuit Board (PCB) and possibly a pluggable module or backplane connector.

The XAUI system's architecture is shown in Figure 3.1.

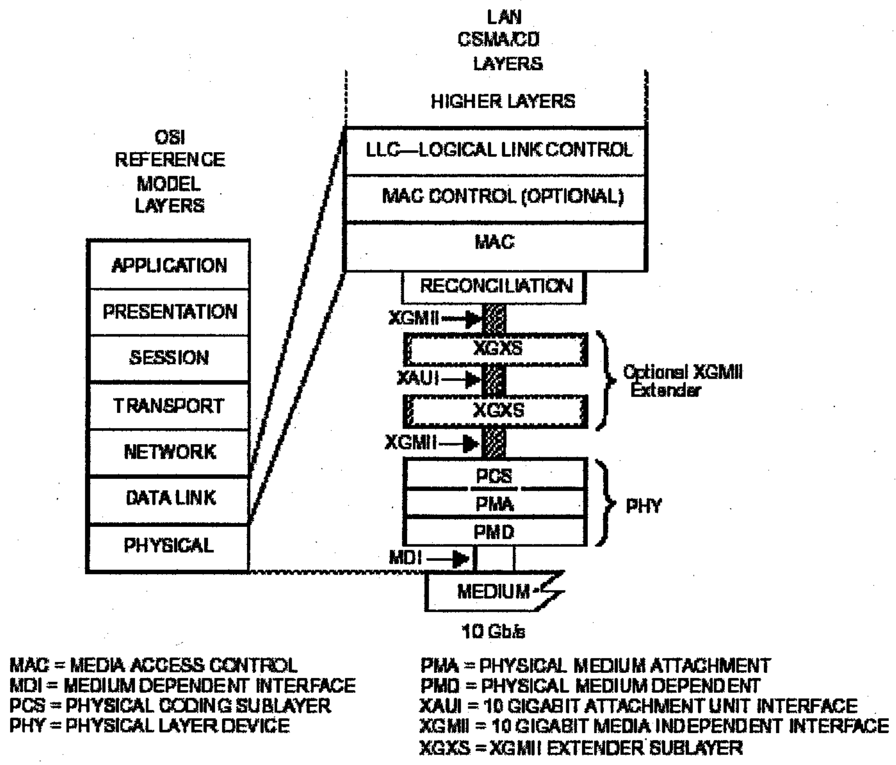

LAN CSMA/CD LAYERS
HIGHER LAYERS
LLC—LOGICAL LINK CONTROL
MAC CONTROL (OPTIONAL)
MAC
RECONCILIATION
OSI REFERENCE MODEL LAYERS
APPLICATION
PRESENTATION
SESSION
TRANSPORT
NETWORK
DATA LINK
PHYSICAL
XGMII
XGXS
XAUI
XGXS
XGMII
Optional XGMII Extender
PCS
PMA
PMD
PHY
MDI
MEDIUM
10 Gb/s
MAC = MEDIA ACCESS CONTROL
MDI = MEDIUM DEPENDENT INTERFACE
PCS = PHYSICAL CODING SUBLAYER
PHY = PHYSICAL LAYER DEVICE
PMA = PHYSICAL MEDIUM ATTACHMENT
PMD = PHYSICAL MEDIUM DEPENDENT
XAUI = 10 GIGABIT ATTACHMENT UNIT INTERFACE
XGMII = 10 GIGABIT MEDIA INDEPENDENT INTERFACE
XGXS = XGMII EXTENDER SUBLAYER

Figure 3.1: XAUI System Architecture
(Courtesy: IEEE 802.3ae Standard)

3.1 XGXS Architecture

The XGXS blocks in the transmit and receive directions are shown in Figure 3.2 for a single lane. The XAUI system comprises of four such lanes. The XGXS in the XAUI Transmit direction encodes the 8-bit input to a 10-bit wide output word and sends it to the Transmit block, which serializes the 10-bit input into a 1-bit output. The inputs to the XGXS are the XGMII signals denoted by TXD<7:0> (Data), TXC<0> (Control) and TCLK (Clock). Convercely, in the XAUI Receive direction, the Receive block deserializes the 1-bit input into a 10-bit output word and feeds it to the Synchronize block on the receive side. The XGXS first synchronizes the incoming bit stream to determine code group boundaries, then de-skews the code groups across the lanes and finally decodes the 10-bit input into an equivalent 8-bit output word and feeds it to the XGMII. The Synchronization and Deskew Processes are explained in detail in section 3.2.

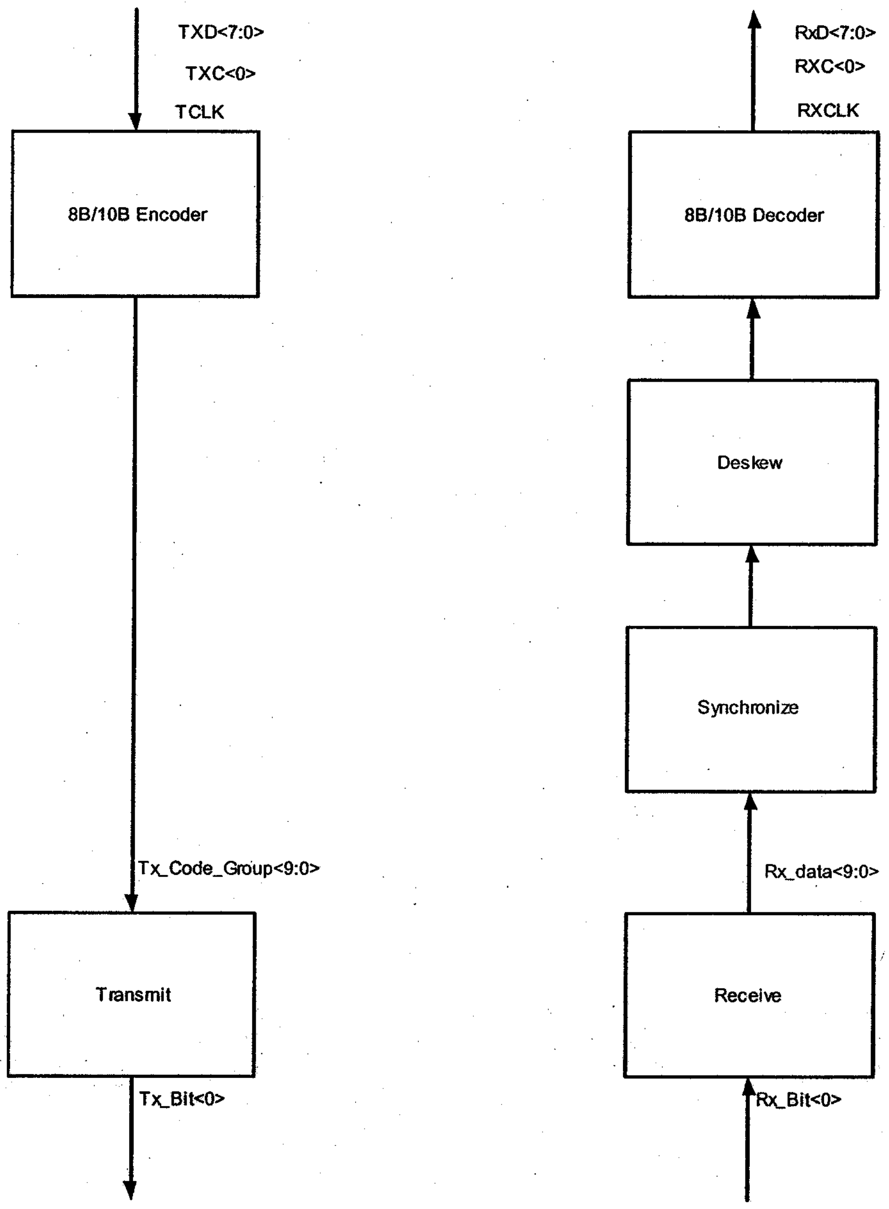


Figure 3.2: XGXS Architecture

In effect, the serializer on the Transmit block is a parallel-to-serial converter and the deserializer on the Receive block is a serial-to-parallel converter. This serial-parallel-serial conversion results in two synchronization issues, as described below, on the Receive side:

a) Synchronization within a lane: As the 10-bit words are continuously converted to a serial stream, it is very difficult for the receiver to find the word boundaries within the lane. This process is referenced as Synchronize in Figure 3.2.

b) Synchronization between lanes: The XGXS blocks operate in parallel on 4 lanes. Synchronization between lanes is necessary to correctly interpret code groups across the 4 lanes. Hence, the receiver should deskew code groups across the four lanes. This process is referenced as Deskew in Figure 3.2.

Since most of the complexity of the XGXS interface resides in these synchronization functions, the details of these processes are next described in section 3.2.

3.2 Synchronization Processes

Full synchronization on the receive side of XGXS requires two phases - synchronization within lanes and across lanes

1) Synchronization within lanes is carried out independently and in parallel. This process identifies the code group boundaries and eliminates bit misalignment within lanes. The in-lane synchronization process works by detecting a special pattern within the code groups and locates the code group boundaries using this special pattern. The special pattern used for the code group alignment is the Comma pattern. Comma is a 7-bit code group, which can be either 0011111 or 110000 depending on the running disparity (see section 3.5 for definition of running disparity).

For example, if the serial bit stream is 11**0011111**0/1100110011/00..., then code group alignment takes place because of the comma pattern's appearance, and the new 10-bit code groups are **0011111**011/0011001100/... with the comma pattern being highlighted in bold. This process is implemented as a pattern detector that detects the comma pattern and sends control signals to the Multiplexer, which multiplexes the 10-bit output in accordance with this control signal. The implementation details are described in Chapter 5.

2) The "synchronization across lanes" process makes sure that the lane-to-lane skew is eliminated. Skew may be introduced by both active and passive elements of a 10GBASE-X link. The alignment process waits for the first observed /A/ pattern on any lane. This process identifies one /A/ in each of the four lanes, then it de-skews the four lanes starting from this /A/ pattern.

For example, consider the following case

Time	Lane0	Lane1	Lane2	Lane3
T		A		
T+1		K		A
T+2			A	K
T+3	A		K	
T+4	K			

After the De-skew process, the code groups appear as shown after time delay (t).

Time	Lane0	Lane1	Lane2	Lane3
T+t	A	A	A	A
T+1+t	K	K	K	K
T+2+t				
T+3+t				
T+4+t				

Hence, this process makes sure that the lane-to-lane skew is eliminated.

This process is implemented by writing the code groups into a RAM, and the addresses of the /A/ patterns across individual lanes are noted. Once an /A/ pattern is detected in all four lanes, the read pointers across the four lanes point to the locations of /A/ patterns in the individual lanes. The RAM read operations begin from these new locations, thus de-skewing the lanes. The implementation details are described in Chapter 5.

3.3 8B/10B PCS Coding

This clause of the PCS specification consists of mapping each 8-bit input into a unique 10-bit output based on the current running disparity, which results in increased transition density to assist the receiver to recover the clock. Running disparity (RD) is a measure of 1's and 0's in the 10-bit output. RD is negative if the 10-bit output has more 0's than 1's and positive if the 10-bit output has more 1's than 0's.

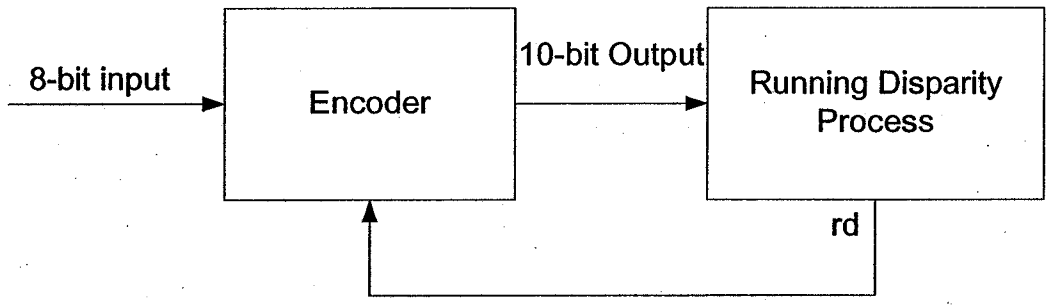


Figure 3.3: 8-10 Bit Mapping

Figure 3.3 illustrates the 8 to 10-bit mapping. In this process, every 8-bit input is mapped into a unique 10-bit output based on the current running disparity. The 10-bit output thus determined is used for calculating the new running disparity, which in turn is used for encoding the next 8-bit input. Thus every 8-bit input assumes one of two possible 10-bit outputs based on the current running disparity. This process utilizes two lookup tables.

Data code groups lookup table: This consists of 256 different data code groups ranging from 00(Hex) to FF (Hex). Data code groups are mapped to their corresponding 10-bit outputs from this table.

Special code groups lookup table: This consists of 12 different special code groups. Some of the code groups relevant to the PCS are presented below.

K28.5: PCS Idle Character (/K/).

K28.3: PCS Idle Character (/A/).

K28.0: PCS Idle Character (/R/).

K28.4: PCS Sequence Character (/Q/).

K27.7: PCS Start Character (/S/).

K29.7: PCS Terminate Character (/T/).

These code groups are explained in detailed in section 3.4.

3.4 PCS Code Groups

The code groups used by the 10GBASE-X PCS are:

1) Idle (||I||): An idle column, (column refers to the code group transmitted on each lane during a single group interval) is transmitted across four lanes whenever the XGMII is idle, and this condition is denoted by TXD=07 & TXC=1. A sequence of ||I|| columns consist of consecutive transmissions of ||K||, ||A||, or ||R||, where ||K|| represents a column of /K/ code groups, whereas /K/ represents K28.5 in a single lane. This notational convention applies to all other code groups. The Idle columns are

a) ||K||(Sync code): The /K/ code group is used for synchronization. The /K/ code group contains a comma pattern which identifies a code group boundary. If this pattern is detected, then code group alignment takes place in accordance with the new boundary.

b) ||A||(Align code): The /A/ pattern is used for alignment across lanes, which eliminates the skew between lanes introduced by both active and passive elements of a 10GBASE-X link. This /A/ pattern is initiated simultaneously on all four lanes by the transmitter so that there is minimal initial skew. This pattern doesn't occur in any other ordered set.

c) ||R||(Skip code): This code group is used for clock rate compensation when operating on multiple clock domains. Clock rate compensation is achieved by insertion or deletion of the Skip code groups.

||I|| can be transmitted as ||K||, ||A|| or ||R||. The rules for the idle sequencing are

a) ||I|| sequencing follows the ||T||-Terminate code group (see item 4 below).

b) ||A|| is sent every 16 to 31 columns.

c) The first ||I|| following ||T|| alternates between ||K|| or ||A||.

d) ||R|| is always sent as the second ||I|| following ||T||.

e) If ||A|| is not sent, then ||K|| or ||R|| is sent based on a random integer generator.

2) Data (/D/): The /D/ character conveys an octet of data sent by the XGMII, and the incoming character from the XGMII is encoded as /D/ whenever the TXC line of XGMII is 0.

3) Start (||S||): The start or ||S|| ordered set directly maps to the XGMII start control character in lane 0 followed by three data characters in lanes 1 through 3. ||S|| indicates to the PCS that a packet has been initiated.

4) Terminate (||T||): The ||T|| ordered set directly maps to the XGMII terminate control character in any lane preceded by data characters and followed by idle characters if Terminate is not in lane 3.

5) Error (/E/): The Error code group is directly mapped to the XGMII Error control character. All invalid code groups are mapped to the /E/ code group.

6) Sequence (||Q||): The sequence or ||Q|| ordered set directly maps to the XGMII sequence control character on lane 0 followed by three data characters in lanes 1 through 3. ||Q|| indicates to the PCS that a link status message has been initiated. Sequence ordered sets are always sent over the PMA service interface immediately following the ||A|| ordered set.

3.5 The XAUI Test System

The new Test System design was implemented on the Xilinx ML321 evaluation board using a Virtex-II Pro FPGA (XC2VP7-FF672). The speed grade of the system is -6.The Test System also utilizes the embedded PowerPC processor, General Purpose Input and Output (GPIO) and UART interfaces on the FPGA. The FPGA is programmed primarily in Verilog HDL (the core FPGA fabric was coded in Verilog, and the PowerPC

Peripherals were coded in VHDL). Xilinx ISE and EDK 8.2i design environments were used for developing the test system. Expanded details on the Test System design are provided in chapters 4 and 5.

The following sections describe the features of the evaluation board that were used in developing the Test System.

3.6 Xilinx 10Gigabit Ethernet Logic

The PCS layer connects between the Xilinx RocketPHY™ 10Gbps transceivers and Xilinx LogicCORE™ XAUI core. The PMA is implemented by RocketPHY transceivers and the MAC layer is implemented by the Xilinx LogicCORE™ MAC. The Digital Clock Manager Core (DCM) was used for the clock synthesis, coarse or fine phase adjustments and synchronization.

3.6.1 Xilinx RocketPHY™-10Gbps Transceivers

The RocketIO transceivers are based on Mindspeed's SkyRail™ technology. The transceiver module is designed to operate at any serial bit rate in the range of 600 Mb/s to 3.125 Gb/s per channel. Each serial link operates at 3.125 Gb/s to accommodate both data and the overhead associated with 8B/10B coding. The operating frequency is implied by the received data, the reference clock applied, and the SERDES_10B (serializer/deserializer) attribute set by the user.

The RocketIO transceiver comprises of the Physical Medium Attachment (PMA) and Physical Coding Sublayer (PCS). The PMA contains the serializer/deserializer

(SERDES), TX and RX buffers, clock generator, and clock recovery circuitry. The PCS contains the 8B/10B encoder/decoder and the elastic buffer supporting channel bonding and clock correction. The PCS also handles Cyclic Redundancy Check (CRC).

3.6.2 PowerPC Interfaces

The PowerPC 405 processor provides a set of interfaces that support the attachment of Xilinx Cores and user logic. The following are the list of interfaces that are available in the Virtex-ii Pro device family.

a) Processor local bus interface (PLB)

b) Device control register interface(DCR)

c) Clock and power management interface

d) JTAG port interface

e) On-chip interrupt controller interface

f) On-chip memory controller interface

The PowerPC interfaces used in the design are described below:

1) JTAG Port: The JTAG port interface supports the attachment of external tools. The JTAG interface was used to download the firmware for the Xilinx ML321.

2) Processor Local Bus: The processor local bus (PLB) interface provides a 32-bit address and three 64-bit data buses attached to the instruction-cache and data-cache units. The on-chip processor local bus (OPB) provides a 32-bit address bus and a 32-bit data bus. PLB is faster than OPB, but consumes more FPGA resources. The *Logic Analyzer* and *Patram* peripherals implemented in the Test System are accessed using the OPB.

.3) On-Chip Memory Controller: An on-chip memory (OCM) interface supports the attachment of additional memory to the instruction and data caches that can be accessed at performance levels matching the cache arrays. The On-Chip Memory (OCM) controller serves as a dedicated interface between the FPGA block RAMs and the OCM signals contained within the embedded PowerPC 405 core. The data-side OCM (DSOCM) utilizes the Block RAMs dual-port feature to enable both read and write data transfer between processor and FPGA. The instruction-side OCM (ISOCM) is used for the storage of interrupt service routines. The DSOCM was used to write the received data to the *Logic Analyzer* Block RAM and then transfer the data to PC for analysis.

In this chapter, the 10 Gigabit Ethernet XAUI system was reviewed. Xilinx Logic cores that enabled the re-design of 10 Gigabit Ethernet test systems were also reviewed. Chapter 4 presents the implementation details of the Transmit side of the Test System.

CHAPTER 4

PCS TRANSMIT SYSTEM

The transmit side of the 10 Gigabit Ethernet system is discussed in this chapter. Implementation details of the transmit system are also presented.

The 8B/10B encoder implements the transmit finite state machine specified in clause 48 of the IEEE 802.3ae standard. The main function of the encoder is to encode each 8-bit input vector into an associated 10-bit output. The encoding process consists of randomizing the XGMII idle stream to reduce electromagnetic interference and encoding the XGMII data characters based upon a lookup table specified in the IEEE standard. There are two lookup tables: one for encoding the XGMII data stream, and the other for encoding the XGMII special code groups, the distinction between them being identified by the TXC value.

The encoder accepts the 8-bit input word and maps it into a unique 10-bit output word based on the current running disparity. The 10-bit output is used to find the new running disparity, which is used as the current disparity for the next 8-bit code group.

Running disparity is a measure of the number of 1's and 0's in a 10-bit code group. Running disparity for a code group is calculated on the basis of sub-blocks, where the first six bits form one sub-block and the next four bits form another sub-block. Running disparity at the end of a code group is applied as the disparity at the beginning of the 6-bit sub-block. Running disparity at the end of a 6-bit sub-block is the running disparity at the beginning of the 4-bit sub-block. Running disparity at the end of a 4-bit sub-block is the running disparity at the end of a code-group. The running disparity for a sub-block is calculated as follows:

1) Running disparity at the end of any sub-block is positive if the sub-block contains more ones than zeros. It is also positive at the end of the 6-bit sub-block if the sub-block is 000111, and it is positive at the end of the 4-bit sub-block if the 4-bit sub-block is 0011.

2) Running disparity at the end of any sub-block is negative if the sub-block contains more zeros than ones. It is also negative at the end of the 6-bit sub-block if the sub-block is 111000, and it is negative at the end of the 4-bit sub-block if the 4-bit sub-block is 1100.

3) Otherwise, the running disparity at the end of the sub-block is the same as at the beginning of the sub-block.

For example if the received 10-bit code group is 1111000110 and assuming that the initial running disparity at the beginning of the code group is negative, then the running disparity for this code group is calculated as follows:

$$- 111100 + 0110 +$$

The 10-bit code group is split into a 6-bit sub-block (111100) and a 4-bit sub-block (0110). Since the 6-bit sub-block contains more 1's than 0's, the running disparity at the end of this block will be positive. This positive value is used as input to the 4-bit sub-block and since this sub-block has an equal number of 1's and 0's, the running disparity at the end of this block will be same as its beginning, this being positive. Hence, the running disparity at the end of the code group is positive.

4.1 PCS Transmit Implementation

The system is designed around the Xilinx ML321 Virtex2 Pro evaluation platform. The system is designed to interface with a PC running customized LabView software to upload /download data patterns, as well as control the board. I/O between the ML321 and the PC is performed via the serial port of the ML321 board. This serial port is also used to interface with the embedded Power PC processor.

Figure 4.1 illustrates the test setup employed throughout the testing process. The XAUI interface of the Device Under Test (DUT) was used to provide access to the DUT in all test cases. Control access to the DUT was provided via a serial port interface. The test system consists of the PC, Lecroy SDA 6000 (DSO), Xilinx ML321 and the Spirent Smartbits interfaces. The test setup described in this chapter is greatly simplified in the next chapter. The Lecroy SDA 6000 (DSO) is replaced by a Logic Analyzer built into the FPGA improve the overall reliability of the XAUI test system. The Spirent Smartbits was also removed from the system.

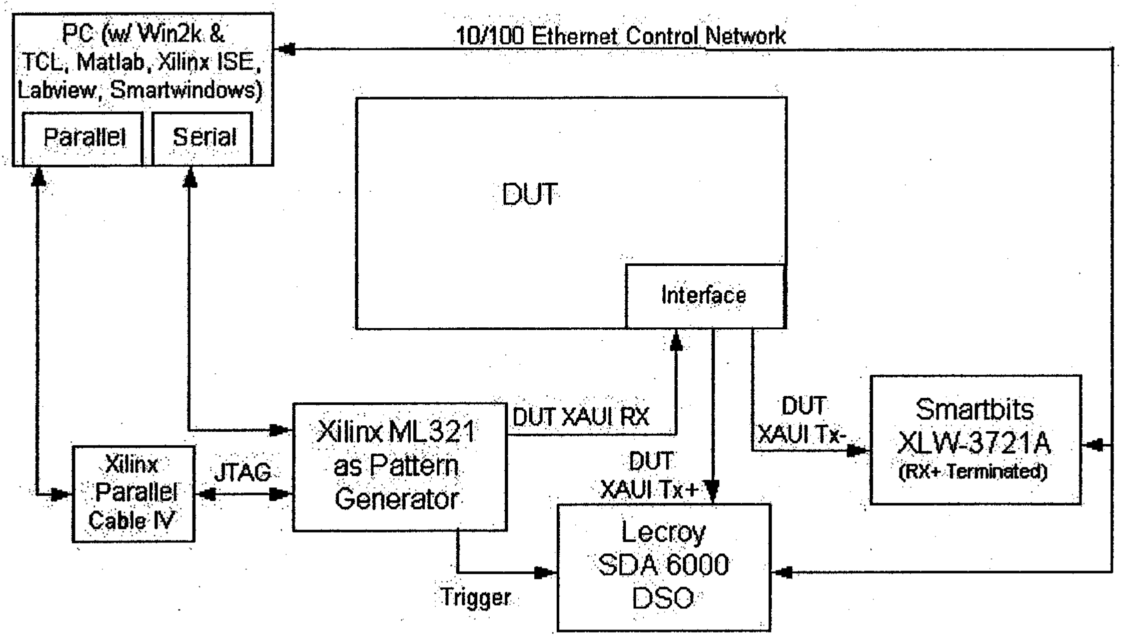

PC (w/ Win2k & TCL, Matlab, Xilinx ISE, Labview, Smartwindows)
Parallel
Serial
10/100 Ethernet Control Network
DUT
Interface
DUT XAUI RX
DUT XAUI Tx-
DUT XAUI Tx+
Smartbits XLW-3721A (RX+ Terminated)
Xilinx ML321 as Pattern Generator
Xilinx Parallel Cable IV
JTAG
Lecroy SDA 6000 DSO
Trigger

Figure 4.1: Original Design of XAUI Test System (IOL)

1) XAUI signaling is generated by the "Xilinx ML321" Testing Station. This system continuously sends valid (though not truly randomized) XAUI Idle when not instructed to send a programmable transmit pattern. When the transmit pattern is sent, a trigger signal is sent to the DSO to capture the response of the DUT.

2) Labview software controls the generation of the test vectors and the programming of the ML321. Labview and Matlab software control the downloading and analysis of the signaling captured on the Lecroy SDA 6000 (DSO).

3) The 4-channel XAUI signaling from the DUT is captured single-endedly on a Lecroy SDA 6000 (DSO). The positive signals (Tx+) are driven into the 50 ohm inputs of the DSO, while the negative signals (Tx-) are driven into negative receive signals of the

Smartbits XLW-3721A interface are terminated with 50ohm terminations on the positive receive channels of the Smartbits.

4) A PC is used for four purposes:

a) via the network to control the DSO and download and process the waveforms in Matlab.
b) via the USB/JTAG interface to download the firmware for the Xilinx ML321
c) via a serial port to control transmissions from the Xilinx ML321

The Virtex2 Pro FPGA is programmed primarily in Verilog HDL (the core FPGA fabric is Verilog, and the PowerPC Peripherals are in VHDL), using the Xilinx ISE and EDK 8.2i design environments. The XGMII interface is divided in to Transmit and Receive functions for ease of presentation. Figure 4.2 illustrates the Transmit side of the system. The Receive side and the PowerPC interface are illustrated in chapter 5.

The Transmit Path module implements transmit related functionality on the FPGA. The functionality is implemented primarily using the Transmit RAM. Two kinds of RAMs could be implemented in the Transmit system. The Primary RAM contains "Looping "and "One-shot" pattern memories. The Secondary RAM contains only "One-Shot" patterns. Only Primary Ram has been implemented at the time of this writing. The Secondary RAM can be implemented in the future as the testing requirements dictate. The Transmit RAMs are shown in Figure 4.3.

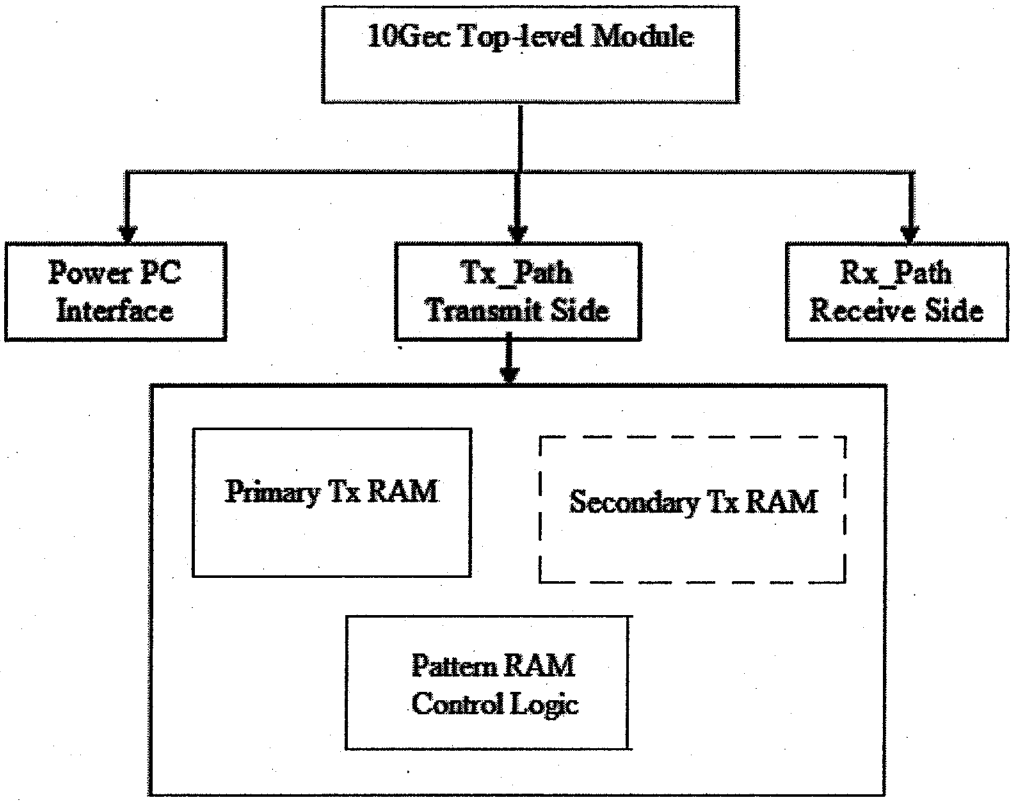


Figure 4.2: Transmit System

The primary RAM consists of a "looping" and "one-shot" sections. The looping pattern allows for a repetitive idle pattern or specific data patterns. Constant Idle is transmitted through the looping pattern to the DUT. The constant idle pattern helps the DUT to acquire synchronization and alignment, thus reducing the risk of Local and Remote faults. The "one-shot" pattern is used for transmission of a specific data pattern. The one shot pattern is the test sequence. "one-shot" pattern can be transmitted when the tester desires (via a signal from the PowerPC control interface). The end of the looping pattern is indicated by Flag bits. The most significant data bit stored is the Flag bit. When the Flag bit is seen, the system exits the "one-shot" pattern and the RAM pointer returns to the beginning of the RAM (looping sequence).

while in the looping pattern if the flag bit is seen, the RAM address pointer returns to the beginning of the RAM. If the fire_one_shot signal is high, when the end of the looping pattern is reached, the address pointer jumps to the address of the beginning of the one-shot sequence, and the RAM is read from the one-shot section until the flag bit is observed again.

The secondary RAM is an entire one-shot pattern store. It can be configured to transmit on manual intervention from the PowerPC control interface, or can be configured to transmit upon reception of a frame. Once a flag bit is seen, the RAM pointer returns to the primary RAM. The Secondary RAM has not been implemented yet. This can be designed in the future as per the requirements of the test suite development.

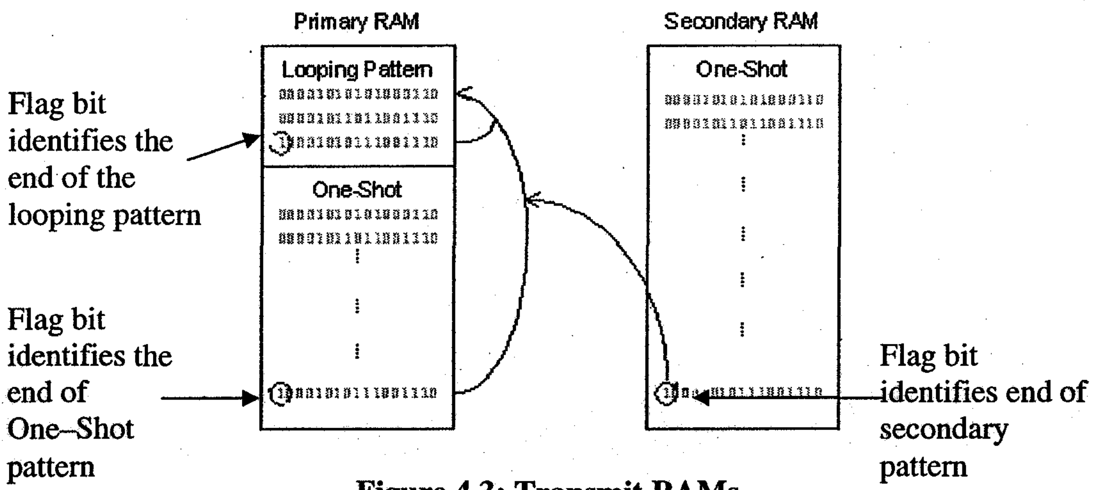

Primary RAM
Secondary RAM
Looping Pattern
One-Shot
One-Shot
Flag bit identifies the end of the looping pattern
Flag bit identifies the end of One–Shot pattern
Flag bit identifies end of secondary pattern

Figure 4.3: Transmit RAMs

The Transmit RAM is a dual-port RAM. The dual-port RAM allows the tester to update the RAM contents on the fly without reprogramming the FPGA itsself. The desired transmit patterns are transmitted to the board over the serial interface, using the Xmodem protocol. Xmodem is a half-duplex communication protocol. It breaks up the data into series of packets for transmission. The receiver, either acknowledges (ACK) or not acknowledges (NAK) the packets. The receiver determines whether the packet is correctly received or not based on the CRC. The Xmodem protocol was available for the Gigabit Média Independent Interface (GMII). The protocol was adopted for use in the 10 Giga bit XAUI testing system. The PowerPC receives the data packets, and updates the pattern RAM through the *patram* peripheral on the on-chip peripheral bus (OPB). The *patram* peripheral, which is 96 bits wide with 16 bits RAM address, was created using EDK 8.2i, to enable the data transfer using Xmodem protocol. The *patram* uses dual port Block RAM for receiving the data on the FPGA from the Power PC and transmitting the same to DUT. The older version of the code did not use Xmodem protocol and *patram* module for transmission. Test data had to be downloaded on to the board for every test case using the Xilinx parallel cable IV interface. The new design helps save the test engineer downloading time. This is especially useful when performing large automated test suites.

One of the test cases involves transmitting skewed frames to the DUT. The XGXS blocks operate in parallel on 4 lanes. Synchronization between these lanes is necessary to correctly interpret code groups across the 4 lanes. Hence, the receiver should deskew code groups across the four lanes. This process is referenced as deskew. This is explained in detail in Chapter 3. To check if the DUT detects the frames with skew, the test system is developed to introduce skew randomly in all the four lanes. The amount of skew required on each lane can be changed by the user on the fly. The Power PC also assists in transmitting frames with skew. The Power PC transmits the skew value via GPIO to the fabric. In the fabric the skew value is interpreted as the delay in the number of transmitted bits for a particular lane for which skew is introduced. If the delay is more than 20 times the bit delay, each 20 times the bit delay is approximated as a one word delay. Hence the skewed frame with a particular word and bit delay can be transmitted to the DUT for testing the Deskew functionality of the DUT. Xilinx presented the skew code to IOL in VHDL. It has been rewritten in Verilog to suit the IOL test requirements. The Transmit side of the code has been verified on the ML321 board. The design works without any errors. The Transmit side of the system was simulated functionally using Modelsim. There were no verification errors. It was successfully used to perform PCS and MAC testing on a vendor's DUT.

4.2 Future Developments

In future HWICAP (Hardware Internal Configuration Access Port) Logic Core can be added to the Transmit system. This module enables the PowerPC to read and write to the FPGA configuration memory through the ICAP (Internal Configuration Access Port). The user can apply four levels of programmable pre-emphasis values during the circuit operation to overcome losses and attenuations in the channel. Variable trigger conditions can also be applied to the built-in Logic Analyzer Module at run time. The Logic Analyzer module is explained in depth in Chapter 5. Dynamic reconfiguration with out any glitches cannot be implemented in the design, if the design uses LUTs (Look Up Tables) or SRL16 (16 bits Serial Shift Register) primitives. The SRL16 has been used in the design for calculating the word delay. In order to use the HWICAP Logic Core, the Transmit System has to be redesigned in the future.

In this chapter background information about Transmit side of the 10 Gigabit Ethernet XAUI Test System was presented. Also the various modules of the Transmit test system were discussed. The 10 Gigabit Ethernet XAUI Receive system architecture and implementation is presented in Chapter 5.

CHAPTER 5

PCS RECEIVE SYSTEM

This chapter presents the implementation details of different modules of the receive system. The new modules that are implemented on the Receive side of the XAUI test system are channel bonding, IPG count, receive triggers and Logic Analyzer. Each of these modules is explained in depth in this chapter.

<u>5.1 8B/10B PCS Receive Process Overview</u>

The receiver section of the Xilinx's Logic Core accepts the 8b/10b encoded low voltage differential serial data from the DUT. The clock recovery circuit locks to the data stream and extracts the bit rate clock, which is 3.125 Gb/s per channel.

The recovered clock (RXRECCLK) from the clock recovery circuit reflects the data rate of the incoming data. The FPGA core consumes the data at a different rate this clock is referred to as RXUSRCLK. Since the two clocks have different sources there is always a difference in the clock rates. The difference is accommodated in the receive buffer. The buffer is always half full when the clocks are properly matched. If the RXUSRCLK is faster than the RXRECCLK, the clock correction logic corrects it by reading a repeatable byte sequence, and if it is slower the clock correction logic corrects this by removing a byte sequence that need not appear in the final FPGA core byte stream. This sequence is generally the Idle sequence comprising of /R/(skip),/K/(comma) and /A/ align characters.

The serial data is then clocked to serial-to-parallel shift registers. The 10 bit parallel data is then multiplexed and fed into 8B/10B decoders. The data is then synchronized through the detection of the incoming K28.5 synchronization pattern. The synchronization pattern or comma was explained in chapter 3. The decoder then generates a synchronization signal indicating alignment of the data to their 10 bit boundaries. The 10 bit data is then converted to 8 bit data while also removing the control words. The decoder also signals reception of various special characters and errors.

1) RXCHARISK: If the RXCHARISK port is asserted High, it indicates that the received byte of data is a control (K) character. Otherwise, the received byte of data is a data character.

2) RXRUNDISP: The RXRUNDISP port indicates the disparity of the received byte is either negative or positive. RXRUNDISP asserted High indicates positive disparity.

3) RXCOMMADET: The comma detect signal RXCOMMADET registers a comma on the receipt of any plus-comma, minus-comma, or both. RXCHARISCOMMA allows the decoder to detect the three defined commas (K28.1, K28.5, and K28.7) as plus-comma, minus-comma, or both.

4) RXDISPERR: The decoder separately detects both "disparity errors" and "out-of-band" errors. A "disparity error" occurs when a 10-bit character is received that exists within the 8B/10B table, but has an incorrect disparity. An "out-of-band" error occurs

when a 10-bit character is received that does not exist within the 8B/10B table. The Receive system data flow is represented in figure 5.1

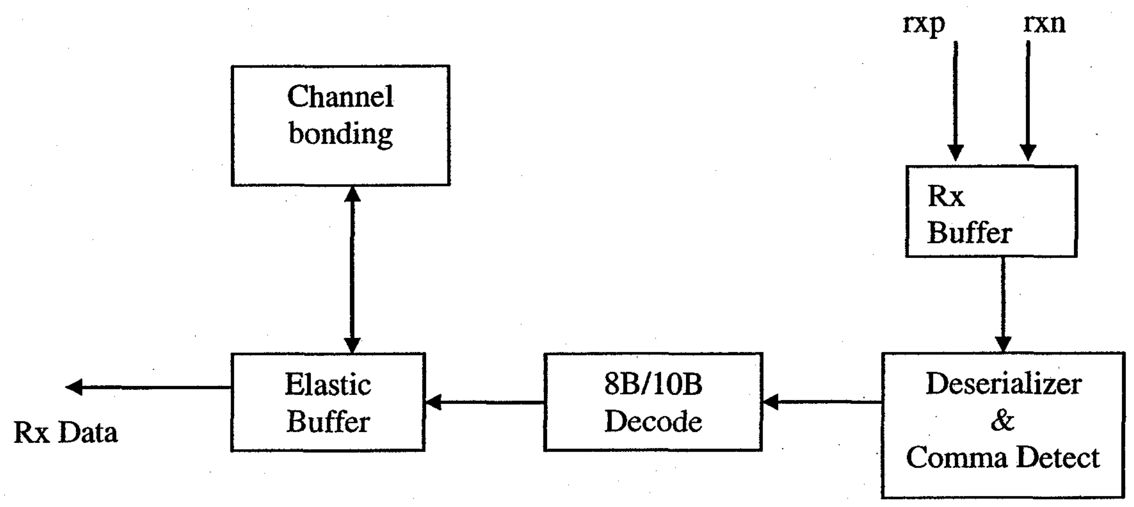


Figure 5.1: Rx Data Flow

The differential serial ports RXN, RXP, TXN and TXP are connected directly to the external ports. For the implementation of the Receive System, 8B/10B decoding was enabled by setting the RX_DECODE_USE attribute to TRUE.

The use of multiple transceivers in parallel for higher data rates results in splitting words of data. Each byte is sent over a separate channel or transceiver. Due to variations in transmission delay, the FPGA core might not assemble these bytes correctly into words. This misalignment is referred to as lane-to-lane skew. The system can tolerate up to 84 UI (29.9ns) of skew. To correct this misalignment, the data stream includes special characters called Alignment character "A". Each receiver recognizes the "A" character as a channel bonding character. Each transceiver remembers the location of "A" in its buffer. One transceiver designated as Master instructs all the transceivers to align to the channel bonding character. The following is an example for a channel bonding sequence

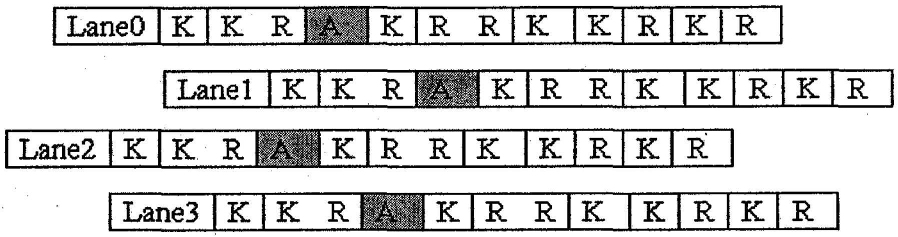

Lane0 K K R A K R R K K R K R
Lane1 K K R A K R R K K R K R
Lane2 K K R A K R R K K R K R
Lane3 K K R A K R R K K R K R

Figure 5.2: Lane Skew at the receiver input

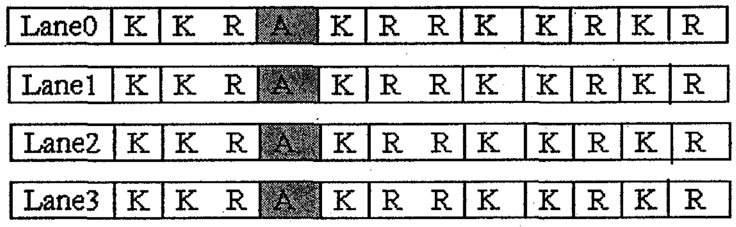

Lane0 K K R A K R R K K R K R
Lane1 K K R A K R R K K R K R
Lane2 K K R A K R R K K R K R
Lane3 K K R A K R R K K R K R

Figure 5.3: Lanes after performing Channel bonding

5.2 Implementation of the Receive System

The earlier Receive System developed by Xilinx had very limited features. When the transmit pattern was transmitted to the DUT, a trigger signal was sent to the Lecroy SDA 6000 (DSO) to capture the 4-channel XAUI signaling from the DUT single-endedly. There were two types of trigger signals: Trigger on Zap and Trigger on Frames. Trigger on Zap is a trigger signal from the test system user. The trigger signal is sent via GPIO from the PowerPC to capture the data. The Trigger on Frames signal would trigger the DSO whenever it saw a start character on the received data.

In the new Receive system the features like channel bonding, Inter Frame Gap (IFG) calculation, and on-board Logic Analyzer were implemented. Each of these modules is described in this chapter. Figure 5.2 gives an overview of the Receive System architecture.

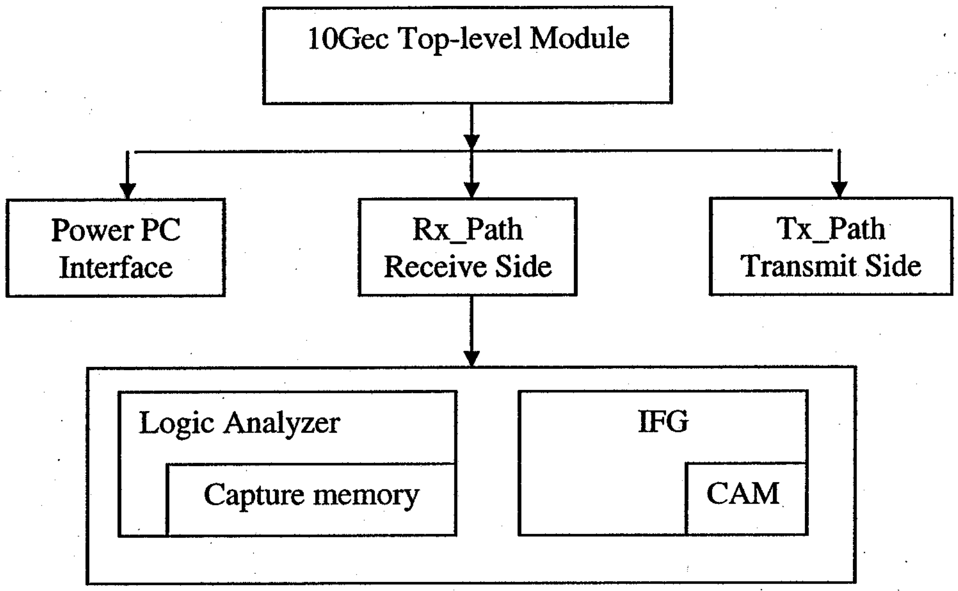


Figure 5.4 Receiver Architecture

5.2.1 Channel Bonding

The channel bonding feature was implemented on the Receive system to enable calculation of inter-frame gap and also to enable the implementation of logic analyzer on the board.

The Xilinx provided channel-bonding feature can be implemented only if the CLK_CORRECT_USE attribute is set to true. This attribute enables the elastic buffer to repeat or skip the clock correction sequences (Idle) to compensate for differences between the clock recovered from serial data and the reference clocks. As a result the apparent IFG from the DUT might be effectively increased or decreased for the clock correction process. Hence, it would be impossible to determine the exact true IFG between the frames. If the channel bonding feature provided by Xilinx is set to FALSE, the clock correction can be disabled. The RXRECCLK can be used to drive the Receive logic in the fabric. Thus, there will be no problem of data overflow. However the channel bonding has to be implemented by design changes.

The channel bonding feature was implemented using 4 different FIFOs. The data is transmitted and received as 2 byte data. Each channel receives 2 bytes of data. The use of multiple transceivers in parallel for higher data rates results in splitting words of data. Each byte is sent over a separate channel or transceiver. The first byte of data is followed by the fifth byte of data and the second byte of data is followed by sixth byte of data and so on. Fig.5.2.2 shows the data sequence on each channel.

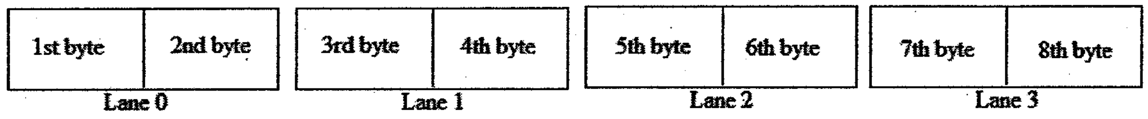


Figure 5.5 Receive Data Sequence

The data has to be arranged in the proper sequence before writing to the Inter-Frame Gap module and the Logic Analyzer module. For the proper alignment of the data one byte wide FIFOs were used. The data rate has to be doubled. Data has to be written into the FIFO at both the rising and falling edges of the clock. Since the board is operating at very high speed, there is always a chance of signal deskew. To avoid the clock-to-signal skew problem, the Digital Clock Manager (DCM) feature provided by Xilinx is used. The DCM module is used to synthesize clock0 and clock 180. Data (8 bits) are written in to the FIFO at the rising edges of both clock0 and clock 180. Data (16 bits) is read at only clk0.

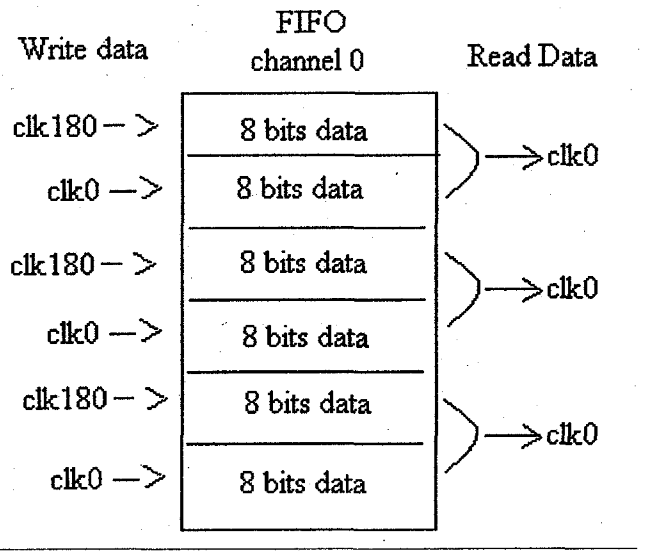


Figure 5.6 Channel 0 FIFO

If the "A" (alignment) character is detected on channel 0 (master), it sends a signal to the FIFOs (slaves) on the other channels, to check f

or "A". When the alignment character is found on all four channels, the address of the location is stored. The read pointer now jumps to this location and reading of the data begins. After performing the channel bonding, the data is sent to the Inter-Frame Gap module. Functional simulation of channel bonding was performed using the Modelsim and Verilog test benches. The design was verified to be functionally correct.

5.2.2 Inter Frame Gap Module

The Inter-Frame Gap module was implemented to calculate the IPG (inter-packet gap) between the received frames. This module was implemented using a finite state machine and CAM (content accessible memory). The Block Select RAM primitive with data width 16 bits and depth 16 words was used to generate a CAM using Xilinx's core generator. The contents for the memory are specified in a text file. The text file contains different combinations of Start and Terminate data. During implementation the values will be embedded in the EDIF netlist. A match signal is asserted whenever the data on the input bus matches the data in one of the locations in the CAM. The FSM (finite state machine) checks if the data is a Terminate character or a Start character. If the data is a terminate character, it starts incrementing the counter until it sees a next match for Start character. The inter packet gap is determined by the counter value. After seeing the match for the start character a second counter starts incrementing until it sees a terminate character. The value of the second counters gives the frame size. Error patterns can also be written to the memory contents of the CAM. Whenever the input data matches the error contents, a flag-bit is set high to indicate error. This module was verified functionally using Modelsim and Verilog test benches. The module was embedded into the previous test system and was verified on the ML321 board.

5.2.3 Receiver Logic Analyzer

The Lecroy SDA6000, which was earlier used in the IOL test setup to capture the received data from the DUT, is now replaced with Logic Analyzer module. The Smart Bits interface has also been removed from the test system. The logic analyzer is implemented using a Block RAM which is 96 bits wide. The depth of the Block RAM is 4096 words. The logic analyzer is controlled via the PowerPC interface. The Logic Analyzer Finite State Machine (LASFSM) captures the received data and control signals from the XGMII's receiver interface.The LAFSM must be armed via the PowerPC interface before data capture will start. Once it is armed, it starts capturing the data. The LAFSM captures data until the capture RAM is filled, or when the user lowers the *arm* signal. Figure 5.5 shows the new setup for the XAUI test system.

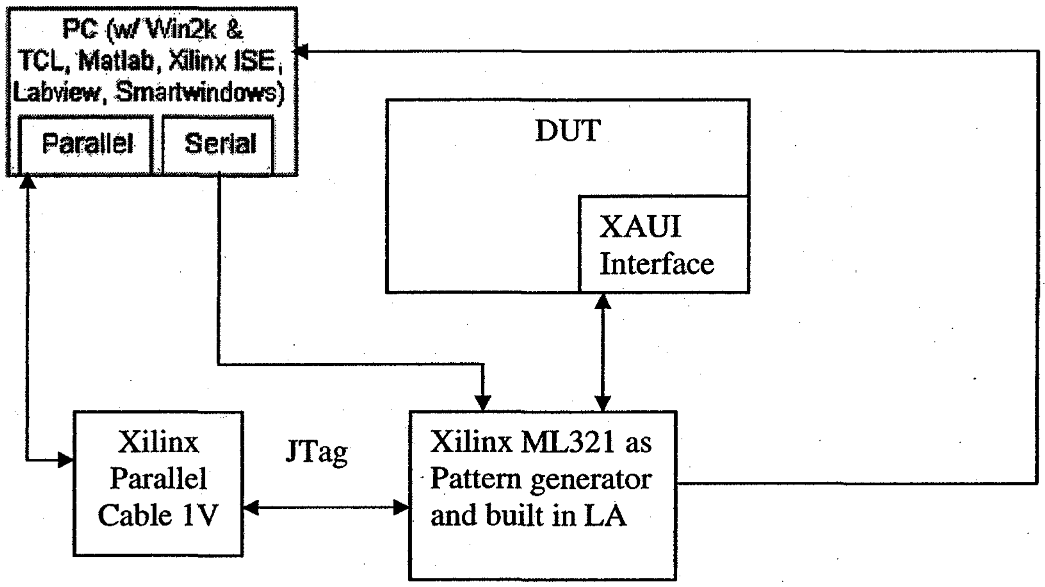


Figure 5.7: 10Gec Test system

After the capture has been stopped, the trig_addr signal indicates the final location where the capture RAM was written. The PowerPC then waits for the user to download the captured data via the serial interface using the Xmodem protocol. The PPC interfaces to the capture RAM via the Logic_Analyzer OBP peripheral. Data is requested via the peripheral from address 0x0000, to the trig_addr. Once 128 bytes of data have been acquired by the peripheral, the PowerPC transmits 1 packet, waits for a positive ACK, then proceeds to transmit the next 128 bytes of data. Once all packets have been transmitted, the PowerPC pulses the trig_ack signal, which resets the LAFSM back to the idle state. The logic analyzer peripheral was designed using EDK8.2i. The peripheral is made of 4 slave registers, each 32 bits wide, used for capturing data and other control signals. The design has been verified functionally. However, the design could not be verified on the evaluation board due to time constraints.

5.3 Future Developments

The Logic Analyzer module captures the data upon the reception of only two kinds of trigger signals, Trigger on Start and Trigger on Zap. More arm signals such as Trigger on Errors, Trigger on Runts or Trigger on Invalid characters can be added in future. These trigger signals were not added to the present module due to time constraints.

LIST OF REFERENCES

[1] Xilinx Virtex 2 Pro datasheet.
http://direct.xilinx.com/bvdocs/publications/ds083.pdf

[2] Xilinx. ML300 evaluation board.
http://www.xilinx.com/products/boards/ml300/index.htm

[3] Xilinx. RocketIO BERT Reference Design User Guide.
http://www.xilinx.com/bvdocs/userguides/ug064.pdf.

[4] IEEE P802.3ae 10 Gigabit Ethernet Task Force,
http://grouper.ieee.org/groups/802/3/ae/

[6] 10 - Gigabit Ethernet - a whatis definition,
http://whatis.techtarget.com/definition/0,289893,sid9_gci535824,00.html

[7] 10 Gigabit Ethernet Tutorial,
http://www.ieee802.org/3/tutorial/nov99/index.html

[8] UG018 PowerPC 405 Processor Block Reference Guide
www.xilinx.com/bvdocs/userguides/ug018.pdf

[9] Embedded design Examples
www.xilinx.com/ise/embedded/edk_examples.htm

[10] XAPP268 "Dynamic Phase Alignment" v1.2
www.xilinx.com/bvdocs/appnotes/xapp268.pdf

[11] System Clock Management Simplified with Virtex-II Pro FPGAs
By: Chris Ebeling, Thane Koontz, Ralf Krueger, and Anil Telikepalli
direct.xilinx.com/bvdocs/whitepapers/wp190.pdf

[12] Connecting the PowerPC Processor to Hardware - PowerPC Example B
www.hunteng.co.uk/pdfs/tutor/powerpc_exB.pdf

[13] Connecting the PowerPC Processor to Hardware - PowerPC Example A
www.hunteng.co.uk/pdfs/tutor/powerpc_exA.pdf

[14] Clock Domain Crossing
www.cadence.com/whitepapers/cdc_wp.pdf

[15] High-Speed Clock Architecture for DDR designs using local inversion
www.xilinx.com/bvdocs/appnotes/xapp685.pdf

[16] RTL Register-Based Memory Implementations
www.actel.com/documents/RTL_Memory_AN.pdf

[17] Xilinx XAPP806 Determining the Optimal DCM Phase Shift for the DDR
direct.xilinx.com/bvdocs/appnotes/xapp806.pdf

[18] Asynchronous & Synchronous Reset Design Techniques - Part Deux
www.klabs.org/richcontent/General_Application_Notes/reset_sync_async_v2.pdf

[19] Simulation and Synthesis Techniques for Asynchronous FIFO Design
www.sunburst-design.com/papers/CummingsSNUG2002SJ_FIFO1.pdf

[20] IBM, On-chip pheripheral bus, architecture specifications version 2.1, 2001.

[21] The Fundamentals of Efficient Synthesizable Finite State Machines
www.sunburst-design.com/papers/CummingsICU2002_FSMFundamentals.pdf

[22] Introduction to Verilog
www.inf.ed.ac.uk/teaching/courses/cd/VerilogTutorial.pdf

[23] Xilinx Application Note XAPP201 An Overview of Multiple CAM
www.xilinx.com/bvdocs/appnotes/xapp201.pdf

APPENDIX A

SYSTEM CONSTRAINTS FILE

```
Release 8.2i Par G.38
Copyright (c) 1995-2004 Xilinx, Inc.  All rights reserved.

MEGHANA-DOOMBOX::  Wed Nov 01 14:49:19 2006


par -w -ol high system_map.ncd system.ncd system.pcf


Constraints file: system.pcf

Loading device database for application Par from file "system_map.ncd".
   "system" is an NCD, version 2.38, device xc2vp7, package ff672, speed -6
Loading device for application Par from file '2vp7.nph' in environment
c:/Xilinx8.2i.
Device speed data version:  PRODUCTION 1.90 2004-11-02.


Resolved that IOB <sys_clk_pin> must be placed at site AC13.
Resolved that IOB <fpga_0_RS232_RX_pin> must be placed at site AA15.
Resolved that IOB <fpga_0_RS232_TX_pin> must be placed at site AC15.
Resolved that IOB <sys_rst_pin> must be placed at site AB12.
Resolved that IOB <fpga_0_RS232_req_to_send_pin> must be placed at site AB15.
Resolved that IOB <dcm_0_lock_pin> must be placed at site AC19.


Device utilization summary:

   Number of External IOBs          102 out of 396    25%
      Number of LOCed External IOBs    6 out of 102     5%

   Number of PPC405s                  1 out of 1     100%
   Number of RAMB16s                 32 out of 44     72%
   Number of SLICEs                1403 out of 4928   28%

   Number of BUFGMUXs                 1 out of 16      6%
   Number of DCMs                     1 out of 4      25%
   Number of JTAGPPCs                 1 out of 1     100%


Overall effort level (-ol):  High (set by user)
Placer effort level (-pl):   High (set by user)
Placer cost table entry (-t): 1
Router effort level (-rl):   High (set by user)

Starting initial Timing Analysis.  REAL time: 2 secs
Finished initial Timing Analysis.  REAL time: 5 secs


Phase 1.1
Phase 1.1 (Checksum:98ea76) REAL time: 5 secs

Phase 2.2

Phase 2.2 (Checksum:1312cfe) REAL time: 10 secs

Phase 3.3
Phase 3.3 (Checksum:1c9c37d) REAL time: 10 secs

Phase 4.5
Phase 4.5 (Checksum:26259fc) REAL time: 10 secs

Phase 5.8
................
...........
................
...........
...........
Phase 5.8 (Checksum:ce808c) REAL time: 33 secs

Phase 6.5
Phase 6.5 (Checksum:39386fa) REAL time: 33 secs

Phase 7.18
Phase 7.18 (Checksum:42c1d79) REAL time: 46 secs

Phase 8.24
Phase 8.24 (Checksum:4c4b3f8) REAL time: 46 secs

Phase 9.27
Phase 9.27 (Checksum:55d4a77) REAL time: 47 secs

Writing design to file system.ncd.

Total REAL time to Placer completion: 47 secs
Total CPU time to Placer completion: 47 secs
```

```
Phase 1: 12187 unrouted;       REAL time: 48 secs

Phase 2: 9553 unrouted;      REAL time: 51 secs

Phase 3: 2445 unrouted;      REAL time: 1 mins

Phase 4: 2445 unrouted; (0)      REAL time: 1 mins

Phase 5: 2445 unrouted; (0)      REAL time: 1 mins 1 secs

Phase 6: 2445 unrouted; (0)      REAL time: 1 mins 1 secs

Phase 7: 0 unrouted; (0)      REAL time: 1 mins 13 secs

Total REAL time to Router completion: 1 mins 15 secs
Total CPU time to Router completion: 1 mins 14 secs

Generating "par" statistics.

*************************
Generating Clock Report
*************************

+------------------------+----------+------+------+------------+-------------+
|      Clock Net         | Resource |Locked|Fanout|Net Skew(ns)|Max Delay(ns)|
+------------------------+----------+------+------+------------+-------------+
|  dsocm_porta_BRAM_Clk | BUFGMUX7S| No   | 1227 |  0.174     |  1.311      |
+------------------------+----------+------+------+------------+-------------+
| jtagppc_0_0_JTGC405TCK |  Local   |      |    1 |  0.000     |  2.026      |
+------------------------+----------+------+------+------------+-------------+


  The Delay Summary Report

  The SCORE FOR THIS DESIGN is: 218


The NUMBER OF SIGNALS NOT COMPLETELY ROUTED for this design is: 0

  The AVERAGE CONNECTION DELAY for this design is:        1.247
  The MAXIMUM PIN DELAY IS:                               8.002
  The AVERAGE CONNECTION DELAY on the 10 WORST NETS is:   4.661

  Listing Pin Delays by value: (nsec)

   d < 2.00  < d < 4.00  < d < 6.00  < d < 8.00  < d < 9.00  d >= 9.00
  --------- --------- --------- --------- --------- ---------
    10328      1777       74        6         2         0

Timing Score: 0

Asterisk (*) preceding a constraint indicates it was not met.
  This may be due to a setup or hold violation.

------------------------------------------------------------------------------
 Constraint                          | Requested  | Actual    | Logic
                                |         |          | Levels
------------------------------------------------------------------------------
 NET "dcm_clk_s" PERIOD =  20 nS   HIGH 50 | N/A        | N/A        | N/A
 .000000 %                          |         |          |
------------------------------------------------------------------------------
 PERIOD analysis for net "dcm_0/dcm_0/CLK0 | 20.000ns  | 11.518ns  | 5
 _BUF" derived from  NET "dcm_clk_s" PERIO |          |          |
 D =  20 nS   HIGH 50.000000 %             |          |          |
------------------------------------------------------------------------------


All constraints were met.
INFO:Timing:2761 - N/A entries in the Constraints list may indicate that the
  constraint does not cover any paths or that it has no requested value.
Generating Pad Report.

All signals are completely routed.

Total REAL time to PAR completion: 1 mins 17 secs
Total CPU time to PAR completion: 1 mins 16 secs

Peak Memory Usage:  147 MB

Placement: Completed - No errors found.
Routing: Completed - No errors found.
Timing: Completed - No errors found.

Writing design to file system.ncd.


PAR done.
```