Feasibility of Serial ATA Cables for the Physical Link in High Performance Computing Clusters

William V. Kritikos

Submitted to the Department of Electrical Engineering & Computer Science and the Faculty of the Graduate School of the University of Kansas in partial fulfillment of the requirements for the degree of Master's of Science

Thesis Committee:

Dr. Perry Alexander: Chairperson

Dr. Ron Sass

Dr. David Andrews

Date Defended

2007/06/12

The Thesis Committee for William V. Kritikos certifies
that this is the approved version of the following thesis:

Feasibility of Serial ATA Cables for the Physical Link in High Performance Computing Clusters

Committee:

_____

Chairperson

_____

_____

_____

Date Approved

Abstract

A novel solution for inexpensive computing cluster networks is proposed and tested for feasibility. The network uses Serial ATA cables — commonly used in personal computers for connecting hard drives — as the physical media for connecting nodes in the computing cluster. The compute nodes, based on a Xilinx Platform FPGA, contain both microprocessors and high speed serial transceivers (that drive the SATA cables). If viable, the approach leads to a very cost-effective communication network. Experimental results do show that a computing cluster network based on SATA cables is feasible and that the cables provide adequately error-free transmission for lengths up to 10 meters.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Ever since the first electro-mechanical computers began to appear almost sixty years ago, there has a been a general push to build ever faster computers. Today, the importance of building the world's fastest computers cannot be understated. High-Performance Computing (HPC) research has had a significant impact on our nation and many segments of the economy now rely on our ability to "out-compute the competition." Indeed, the steady advancement of information technology — and HPC research in particular — has become vital to U.S. economic competitiveness, the advancement of science, and the security of the nation.

Over the past fifteen years, much of the HPC research has focused on improving the microprocessor because it emerged as the universal, basic building block of all parallel HPC systems. From highly customized machines (such as those built by Cray, SGI, and Sun) to Beowulf-class machines (commodity off-the-shelf components running Open Source software), all modern parallel system essentially consist of two parts. One part is a collection of (parallel) compute nodes which today are built around commodity microprocessors. The second part is a communication network that connects the nodes. It is worth noting that while the

very fastest computers are custom designs, most of the 500 fastest computers in the world leverage commodity nodes and interconnection networks. This speaks to the fact that speed alone does not drive the market and that HPC users are very sensitive to cost as well as speed. In other words, to all but the very largest compute centers, cost-effectiveness is an extremely important HPC metric.

For many HPC users, the availability of a cost-effective petascale[1] machine offers enormous potential. However — for power, space, cooling, and other reasons — it is not clear that the currently most cost-effective approach (commodity clusters) will scale to that speed. However the problems that could be solved are very significant. For example, biologists studying communities of microbes for bio-remediation are not able to culture individual microbes in the lab. Without the ability to isolate the individual species, one cannot sequence a species' genome. It has been proposed to gene sequence a whole community simultaneously; however, this approach leads to a computational problem 100× larger than the fastest computer available. Simply buying 100× more commodity off-the-shelf compute nodes, is not feasible. Clearly, an alternative approach is needed.

The Reconfigurable Computing Cluster (RCC) Project is investigating the feasibility of building cost-effective, petascale computers to support computational science. In the Reconfigurable Computing Cluster, the compute nodes are built from Platform FPGA[2] and a novel high-speed interconnection network. Unlike traditional commodity clusters, which rely on external network switches to connect the nodes, the RCC project has proposed to decompose the switch and spread it across the FPGAs. The main idea is that the distributed switching components

[1] a parallel machine that scales up to a peak of 10^{15} floating-point operations per second

[2] a reconfigurable integrated circuit capable of hosting an entire Linux-based system on a single chip

will use the Platform FPGA's on-chip high-speed transceivers to communicate with one another over low-cost, high-speed Serial ATA (SATA) connections.

This thesis is focused on the RCC network and, specifically, the potential value of using SATA connections. While the network components of a traditional commodity cluster typically account for about one-third of the overall cost, the proposed network would be a tiny fraction of a RCC's total budget. This savings comes from using a commodity technology (Serial ATA) in a a unique way. Specifically, SATA was designed to operate inside of chassis (which provides a certain degree of shielding) and over short distances (to connect disk peripherals to main-boards). In the RCC it is used to connect the compute nodes which are farther apart and outside of a single chassis. Consequently, the RCC communication network is much more susceptible to bit errors and one might anticipate that, to be reliable, one has to either slow down the transmission speed, increase transmission overhead, or both. *This leads to a fundamental question for the RCC Project: is the use of SATA links feasible for a HPC network?*

Clearly, an analytical study is not suitable; nor is a simulation likely provide a high quality, reliable results. To answer this question, it was necessary to build engineering prototypes and conduct physical experiments. The experiments described in this thesis were designed to evaluate the custom link layer and physical layer protocols under a variety of conditions. Two prototype network interface boards were designed and built for the experiments; the bit error ratio of several physical configurations was tested, and the effective bandwidth and latency of point-to-point links was analyzed. Tests included continuous streams of data as well as packet-based transmissions and measured hardware core-to-core performance.

The primary contributions of this thesis are:

- A custom PCB designed to interface 8 multi-gigabit transeivers on either Xilinx ML-310 or ML-410 development boards to Serial ATA connectors
- A modular bit error rate testing module which can test an arbitrary number of network links in parallel
- An interface over the PLB to the multi-gigabit transeivers
- Evidence that SATA cables can provide reliable transmission up to 10 m
- A reliable network foundation which will can be used as a basis for HPC clusters

The rest of this thesis is organized as follows. Chapter 2 provides background material including a survey the use of FPGAs in High-Performance Computing systems today, an overview of the Reconfigurable Computing Cluster, and the various commodity interconnection network technologies used in HPC systems today. In Chapter 3, the physical design of the prototype network interface cards are described. Based on those designs, the prototypes were manufactured. Experimental results are presented in Chapter 4. Chapter 5 summarizes the results and describes future work.

Chapter 2

Background

2.1 FPGA Based Computing Clusters

There a number of barriers with scaling today's technology up to a petaflop. These include a number of physical constraints (size, mass, power, cooling) and the cost is simply prohibitive. Medium-sized compute centers simply cannot afford the number standard microprocessors needed to deliver a petaflop.

A novel approach to petascale computing is to use a cluster of field programmable gate arrays (FPGAs). The Reconfigurable Computing Cluster project is currently investigating this approach. The FPGA-centric approach greatly increases the computational density of a computing cluster by providing the logic gates necessary to solve pieces of the computation with combinatorial logic. Along with providing domain-specific cluster computing architectures, the configurability of an FPGA allows researchers to explore alternative approaches to many aspects of the cluster (RAM organization, networking, etc.) while keeping costs on par with commodity off-the-shelf components.

2.1.1 Proposed Cluster Architecture

The proposed implementation of the SATA physical network was not developed to be a stand alone HPC network that could easily be implemented on any HPC computing cluster. The SATA physical network was developed to take advantage of HPC clusters which implement FPGAs for either network management, and possibly for the computation as well. The greatest advantage — in terms of low latency/high bandwidth communication between computation nodes — will be achieved with both the computation and networking communications are handled on a single FPGA.

This combination of computation and network communication is exactly what the Reconfigurable Computing Cluster (RCC) — currently under development at the University of North Carolina at Charlotte — is implementing. This cluster will be composed of 64 ML-410 FPGA platform development boards, and will rely on — if this thesis proves feasible — a SATA based communication network. The Virtex 4 FPGA on the ML-410 [1] board will provide both a platform for massively parallel computations and an on chip router to provide the necessary packet forwarding for the cluster. Figure 2.1 shows the planned FPGA architecture. This figure illustrates a possible advantage for bringing the low level network directly to a configurable computation element — there is no need for an external network router like what you would find in an Ethernet of InfiniBand HPC cluster.

Bringing the switch closer to the computational elements will reduce the latency required to move data from a computational elements to the network interface. The on chip network switch can also be used — with further research — as a mechanism for moving data between computational elements on the same FPGA instead of the traditional system bus model that has been prevalent in

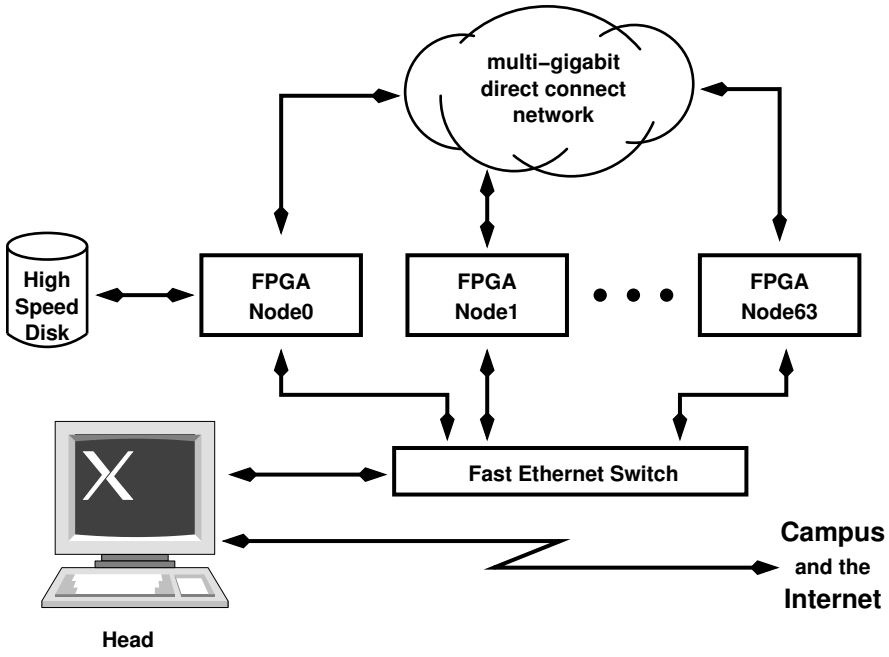

Figure 2.1. Reconfigurable Computing Cluster Block Diagram

Xilinx FPGAs.

2.1.2 BEE2 Cluster

At first glance, the BEE2 cluster under development at UC-Berkeley is very similar to the RCC cluster [2]. Both clusters use Xilinx FPGAs as the computational elements, and both use the Rocket IO Multi-Gigabit Transceivers (MGTs) on these FPGAs as the transceivers in the communication network. However, the BEE2 cluster uses InfiniBand as the physical communication media. Also, while there are on chip switches on some of the FPGAs, these cross bar switches are only serving to move data between 4 FPGAs in a single computation node. The routing between computation nodes is still done with a traditional InfiniBand switch [2].

The BEE2 module as built uses 16 of the 20 Rocket IO MGTs on the Xilinx

V2P70 FPGA to drive four 4-lane InfiniBand connectors for each "User FPGA" on the module. The other four MGTs were connected to SATA connectors as a test platform.

2.2 High performance computing networks

2.2.1 Gigabit Ethernet

IEEE 802.3ab, also known as 1000Base-T or GigE, is often used in HPC clusters made from commodity parts. GigE network interface cards are commodity parts. 1000Base-T uses all 4 twisted pairs in an Ethernet cable simultaneously to transmit 8 user data bits at 125 MBaud. This is physically done by using 4D Trellis Coding to convert the 8 bits into 12 bits or 4 sets of 3 bits each. Each of those 4 sets is then encoded with a 5 level pulse amplitude modulation (PAM) scheme and sent over the 4 twisted pairs of the cable [3].

Xilinx transceivers can be used to directly drive a single pair in an Ethernet cable. A Xilinx characterization report [4] showed that a single pair in a CAT6 cable — currently the highest rated Ethernet Cable — can only support bit rates of 2.5 Gbps on a 14 foot (4.2 meter) cable. This low bit rate and short maximum distance makes Ethernet cable such as CAT5e or CAT6 unsuitable for the RCC.

2.2.2 InfiniBand

InfiniBand (IB) is a complete I/O fabric consisting of host channel adapters (HCA) and switches [5]. IB is a good alternative to GigE for HPC clusters as it offers very high bandwidth and significant hardware support for inter-process message passing. Each IB channel operates at a bit rate of 2.5 Gbps.

Typical HCAs will aggregate 4 of these single channels to make a single 10

Gbps link from the HCA to the IB switch. The HCA also negotiates with the software process to allocate a dedicated section of user addressable memory to allow the user to directly send and receive messages without operating system intervention. This system is referred to as Remote DMA, as a user process on a node can directly access memory on another user process on a different node through IB.

Xilinx has also characterized the performance of IB cable with its transceivers [6]. This report showed that IB cable can support reliable communications on 10 meter cable at data rates up to 3.125 Gbps [6]. This result shows that IB has no technical limitations which would preclude it from use in the RCC.

The greatest disadvantage to IB is its cost. A standard 4-channel 1 meter IB is currently selling for approximately $100 as of May 2007. Furthermore, because only 8 transceivers are available for networking on each node in the RCC, each node could only talk to 2 other nodes while still fully utilizing the expensive IB cable. A more dense interconnection network could only be achieved with IB cables if some of the channels in the 4x cables were left "dark".

2.2.3 Serial ATA

No examples of high performance computing networks can be found in the literature which use Serial ATA cables as the primary interconnect. However, there are two bodies of knowledge which suggest that this is possible.

First is the high similarity between single lane InfiniBand cables and Serial ATA cables. Both cables have 7 conductors: 3 ground conductors,and 2 signal pairs for differential bidirectional communication [7] [8] [9]. Both IB and SATA signal pairs are specified to have a 100Ω differential characteristic impedance.

Both IB and SATA cables have similar propagation delay of 4.2 ns per meter approximately 2 dB per meter of attenuation at 4 GHz [8] [9].

Xilinx has also characterized the performance of SATA cables with their transceivers [10]. This report shows that full speed 3.125 Gbps data transmission is possible on SATA cables up to 7.5 meters in length. 10 meter cables can be used if the bit rate is slowed to 2.5 Gbps.

The similarity of IB and SATA cables along with the results of the Xilinx Characterization reports suggest that SATA cables can work. However it does not prove that SATA cables will work in the RCC. The networking adapter board which is needed to connect the transceivers with SATA connectors could significantly reduce the performance. For a cable to be considered as feasible to use in a HPC cluster then the interface between the cable and the network transceiver must also be feasible. A significant portion of the engineering work in this thesis went into the interface between the SATA connectors and the transceivers on the FPGA.

2.3 High Performance Network Metrics

2.3.1 Bit Error Ratio

Comparison of the error rates between networks is frequently done by comparing the ratio of incorrect bits sent over a network to correct bits sent in some period of time. This ratio is defined as the bit error ratio. A bit error ratio (BER) is usually given as a fraction of incorrect bits to correct bits. For example if 1 bit was incorrect in a stream of 10^{12} then the BER would be given as $\frac{1}{10^{12}}$ or 1×10^{-12}. Scientific notation is often used when the BER is a very small number — which is desirable.

2.3.2 Achievable Bandwidth

The metric for network bandwidth is defined as amount of user data that can be transferred from one user application to another user application through the network in a fixed amount of time. This definition takes into account the inefficiencies of the encoding schemes at the transceiver level, as well as the overhead associated with packetizing the data, and header information which must be sent with every packet.

2.3.3 Error Correction

The bit error ratio also affects the achievable bandwidth between FPGA cores. Whenever a bit error occurs, the packet in which that error occurred must either be corrected or discarded. There are significant trade-offs involved when determining if a network should correct or discard bad packets. As long as the bit errors are very infrequent then the cost of dropping a packet with an error and resending will be less than the overhead of inserting error correction codes into each packet.

2.4 Electrical Fundamentals

2.4.1 Edge coupled microstrip transmission lines

Every wire, even if it was not intended to be, is a transmission line. The traces on the networking board must be carefully designed to give the correct characteristic differential impedance of 100Ω for the frequencies present in the MGT signal. The characteristic impedance of a transmission line is simply the ratio of the voltage and current waves traveling on that line at any given point, with units of Ohms.

The signaling used in the MGTs is called differential signaling, meaning that there is a positive and negative signal line. If you wanted to send a 1 down the differential line, the positive signal would have the voltage corresponding to a 1 on it, while the negative line would have the voltage corresponding to 0. This is called differential signaling. The differential impedance of 100Ω is achieved when both the positive and negative signal lines are behaving as 50Ω transmission line when a properly biased differential signal is applied. The positive and negative signal transmission lines are place a specific distance from each other to achieve this characteristic impedance.

Chapter 3

Implementation

The Reconfigurable Computing Cluster (RCC) has been designed to be a test bed for implementing a multiple novel ideas which will determine the feasibility of using FPGAs in order to scale to a petaflop cluster [11]. In order to evaluate using SATA connections in a HPC communications network two prototype designs have been created to measure its performance under a variety of conditions. The focus of this thesis is to analyze the applicability of using SATA connectors in the network.

3.1 Overview

The Virtex series of FPGAs come with embedded, hard silicon, Multi-Gigabit Transceivers (MGTs), also known as RocketIO transceivers, and are used for the network in the RCC project. The transceivers on the standard Virtex series operate at a maximum bit rate of 3.125 Gbps. Further details are available in Rocket IO Transceiver Users Guide [12] [13].

A prototype networking board has been built to handle the physical connection

between the networking cables and the aforementioned transceivers on the FPGA. In specific board interfaces between the Z-Dok+ connector on an ML-310 [14] or ML-410 [1] FPGA development board and the SATA or external SATA (eSATA) cables. As part of this thesis work, it has been found that controlled impedance manufacturing is necessary to have reliable communication. Two versions of the prototype board have been designed and built with a third currently in development that will be used in the RCC which will draw heavily from this thesis work.

The primary innovation for this cluster is using SATA cables as the physical networking link. The SATA standard supports bit rates of up to 3 Gbps in each direction, over a 100Ω differential pair, for distances up to one meter on a single SATA cable [15]. This one meter limit on cable length is not due to a limitation in the cables. Rather it is a limitation of the standard SATA transceivers found on PC chip sets and hard drives. It has been shown [10] that reliable communications over SATA cables can be achieved using MGTs with up to ten meter long SATA cables.

3.2 Implementation Details

3.2.1 Rocket IO Multi Gigabit Transceivers

As previously stated, the Rocket IO MGTs [12] [13] form a core component of the RCC network. These are highly configurable transceivers which interface quite well with the logic on the Xilinx FPGA, as well as standard impedance networking cables, such as InfiniBand and in this case SATA. The core of the Rocket IO transceiver is the serialize-deserialize (SERDES). The SERDES is responsible for taking the parallel data from the logic, and then serializing it with a significantly

faster clock to be sent to the network link. for the RCC, the MGTs take in parallel data sixteen bits wide clocked at 156.25 MHz. The clock is then internally multiplied by a factor of twenty to achieve the bit rate of 3.125 Gbps. The factor of twenty is needed because 8B/10B encoding is used to maintain clock synchronization between the two nodes communicating in the link. Using this method, the sixteen bits of data coming into the MGT are sent over the SATA link as twenty bits. Figure 3.2.1 shows the architecture of the MGT.

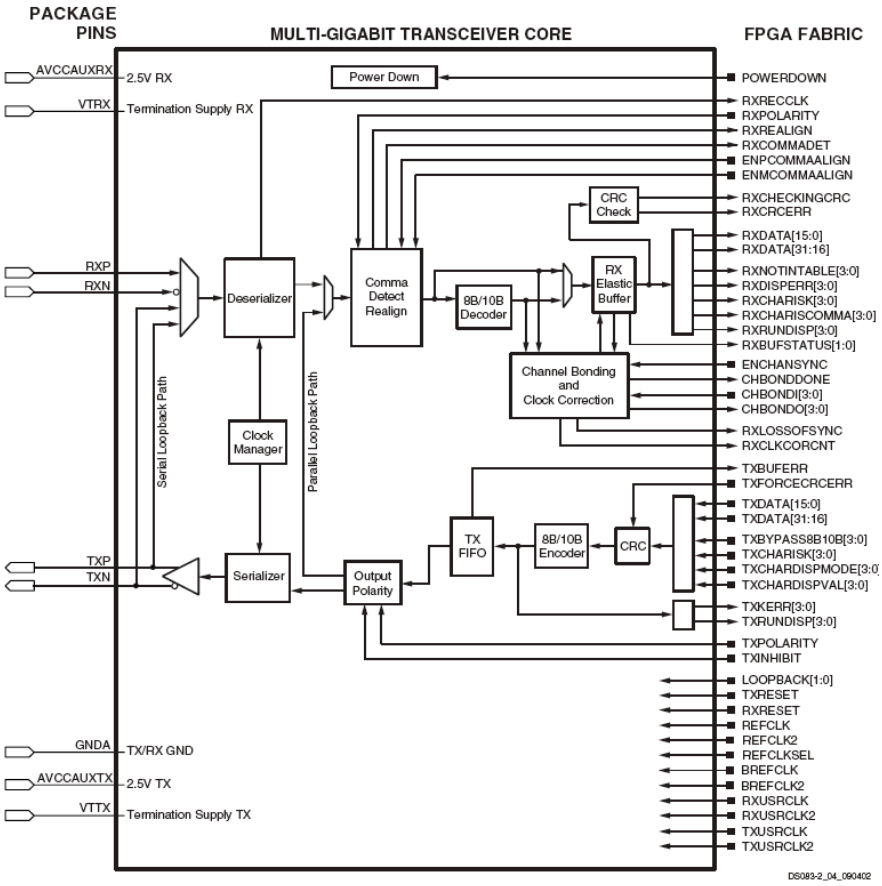

Figure 3.1. Rocket IO Transceiver Block Diagram
Source : Xilinx

3.2.2 Aurora

Aurora is a link layer protocol provided freely by Xilinx [16]. Its use in the RCC is primarily for framing the data and inserting periodic clock correction sequences. Aurora provides these services, as well as channel bonding and support for streaming data if necessary. Channel bonding essentially takes two or more single SATA links between two nodes and concatenates them together, making a single higher bandwidth channel. It is possible to use channel bonding in the cluster; however, the ML-310 and ML-410 boards only provide access to eight MGTs off the board through the Z-Dok+ connector which severely limits the ability to implement channel bonding or interesting network architectures. For example, one interesting external network architecture to study — a 3d torus — requires 6 channels per node. There are not enough extra MGTs available to support bonding channels while still being able to support the desired 3d torus architecture.

3.2.3 Transmission Line Design

The transmission line used in this design is an edge coupled microstrip line. The primary constraints for a transmission line design is range of frequencies where a certain characteristic impedance must hold. The MGT signals have a rise time of 120 ps [17]. This rise time will result in signals which have frequency components from 0 to 4.2 GHz [18]. The MGT can drive either 100Ω or 150Ω differential transmission lines. 100Ω was selected due to the SATA cables and Z-Dok+ connector both being designed for 100Ω systems.

The ideal geometry for the edge coupled micro strip transmission line is too complicated to solve with a simple equation. The ADS suite is used [19] to find the

ideal trace geometry which will show a 100Ω differential impedance for frequencies up to 4.2 GHz. An additional constraint is that the transmission lines must be thin enough to route through the tight spaces on the Z-Dok+ connector. The final geometry given by ADS is shown in Figure 3.2. The board dielectric used in this design is standard FR4 material with a rel permittivity (ϵr) of 4.3.

3.2.3.1 Board Stack Up

Figure 3.2 is a cross section of the second revision of the networking board. Two layers of MGT signals were required to route all eight MGTs from the Z-Dok+ connector. Note that the thickness between the microstrip transmission lines and the ground plane was specified as five mils, but was actually built as twenty mils in the first revision board.

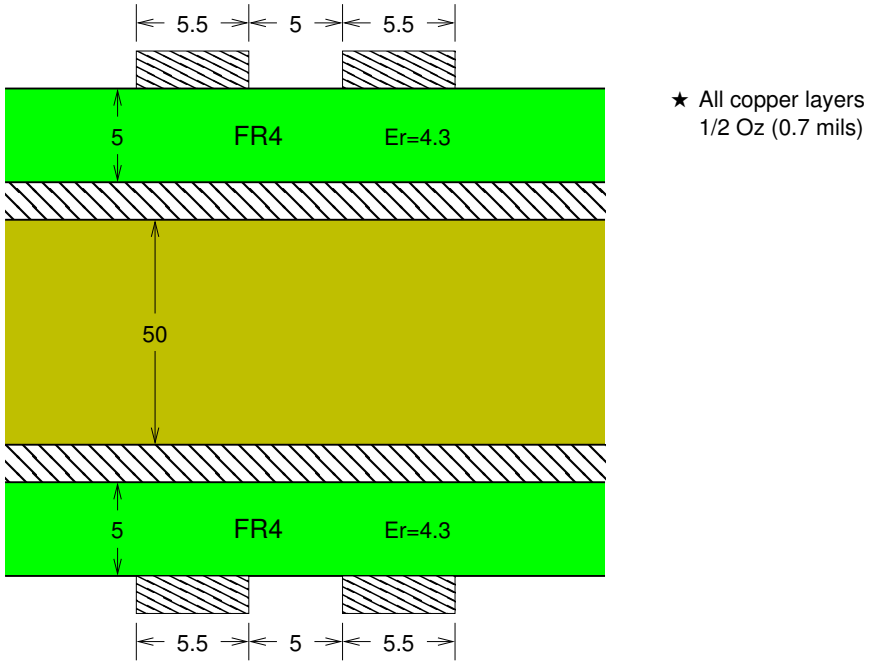

Figure 3.2. Board Cross-section

3.2.4 Connectors

This section details all of the connectors used in the design of the networking board. The connectors play a vital role in the board's functionally and must be given significant attention.

3.2.4.1 Z-Dok+ Connector

The Z-Dok+ connector (Figure 3.3) is manufactured by Tyco Electronics and is rated for bit rates up to 6.25 Gbps [20]. This connector maintains the 100Ω differential pairs through the connector, allowing it to reach these high speeds. Xilinx has used the Z-Dok+ connector on the ML-310 and ML-410 development boards to allow its users access to the MGT signals. Figure 3.4 shows how the spacing between the pins is very tight, and how the traces must be routed.

3.2.4.2 SATA and eSATA Connectors

There are many options when choosing a SATA or eSATA connector, including PCB mounting (through hole or surface mount) and direction of cable. The following figures show the various SATA and eSATA connectors that were used in the two revisions of the networking board. The vertical SATA connector is Molex part number 67800-8101. The surface mount SATA connector is Molex part number 67490-1220. The surface mount eSATA connector is Molex part number 47082-1000.

Figure 3.5 shows the SATA connector used in the first revision board. It used through hole pin mounting, and straight cable orientation. The second revision of the networking board uses a hybrid footprint which will work with either the right angle surface mount SATA or eSATA connectors shown in Figure 3.6 and

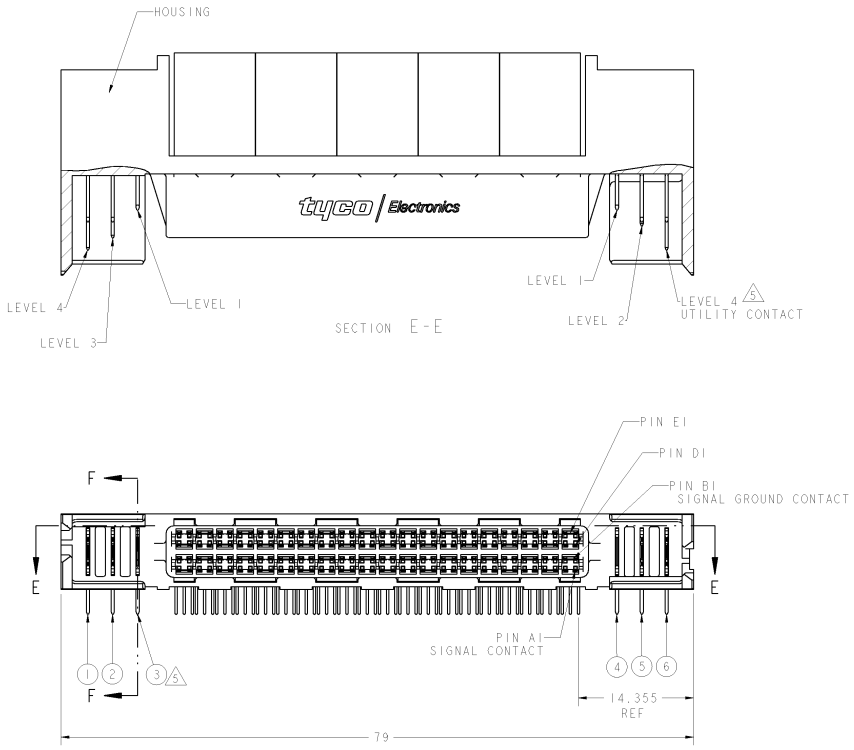

Figure 3.3. Z-Dok+ Connector Drawing
Source : Tyco Electronics

Figure 3.7. The hybrid footprint allows relatively quick change out from SATA to eSATA, so that the relative benefits of the shielded eSATA cable can be measured with the same board.

3.2.5 First Revision Networking Board

The first attempt at an adapter board to interface between the Z-Dok+ connector on the ML-310 board and a SATA network is shown in Figure 3.8 and Figure 3.9. This board has several fatal flaws which lead to very poor performance.

The most significant flaw is that the distance between the ground plane and

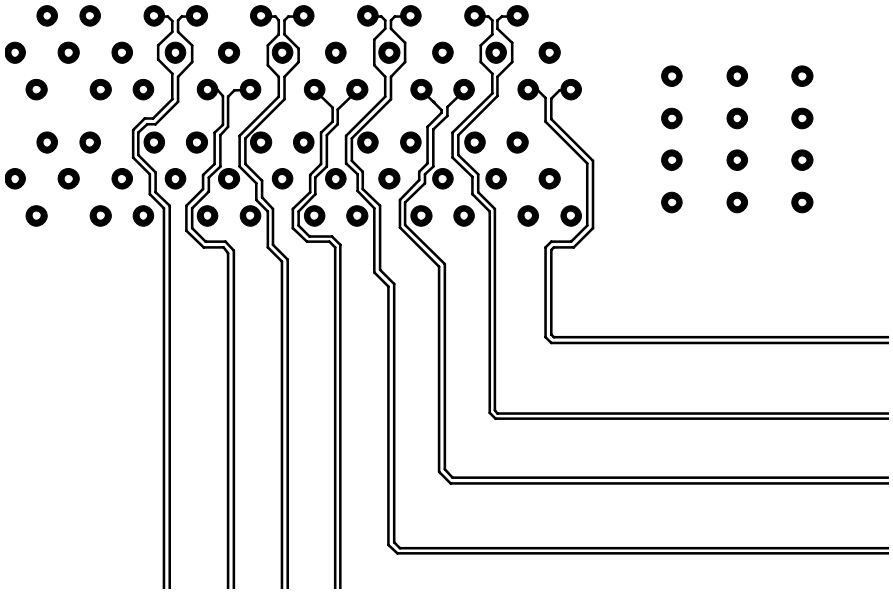

Figure 3.4. Z-Dok+ Trace Routing

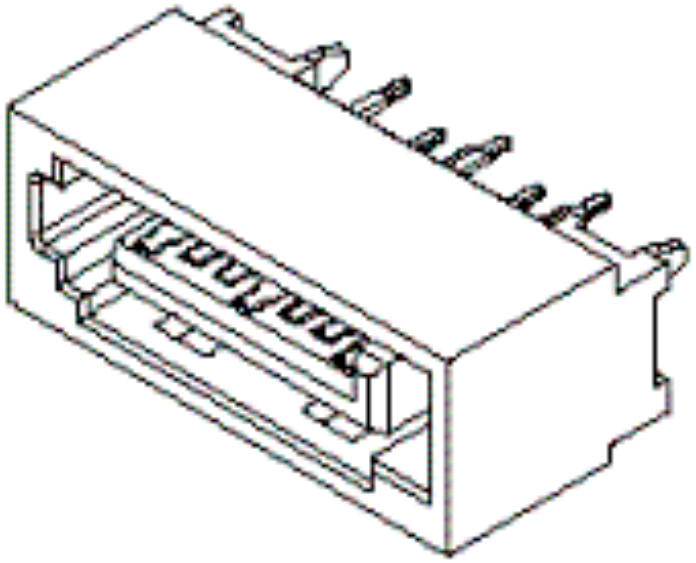

Figure 3.5. Through Hole Mount SATA connector
Source : Molex Incorporated

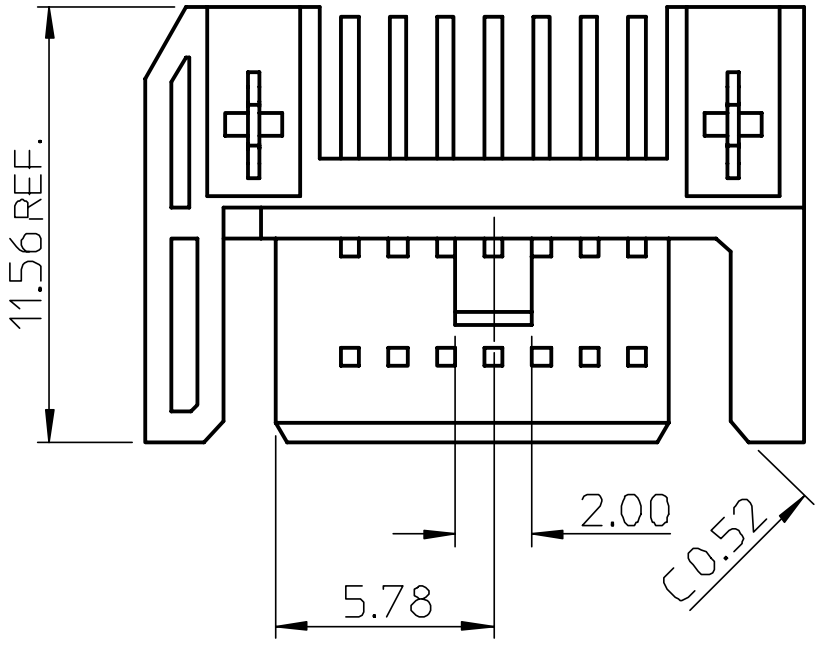

Figure 3.6. Surface Mount SATA connector
Source : Molex Incorporated

the transmission lines was not built as specified. This was not the fault of the manufacturer, but of the design. The boards were ordered using the cheapest manufacturing technology available, which does not allow for custom thickness between copper planes, only twenty mils. The traces were designed to work with a five mil dielectric. The trace thickness on the board could have been adjusted to make the 100Ω transmission line with the standard thickness dielectric. However, this would have made the transmission line traces too wide to route through the Z-Dok+ connector.

A second flaw in the design is the 90 degree bends on the traces. The thickness of the trace increases by a factor of $\sqrt{2}$ around these corners [18]. This will significantly change the impedance at the corner, creating an impedance mismatch

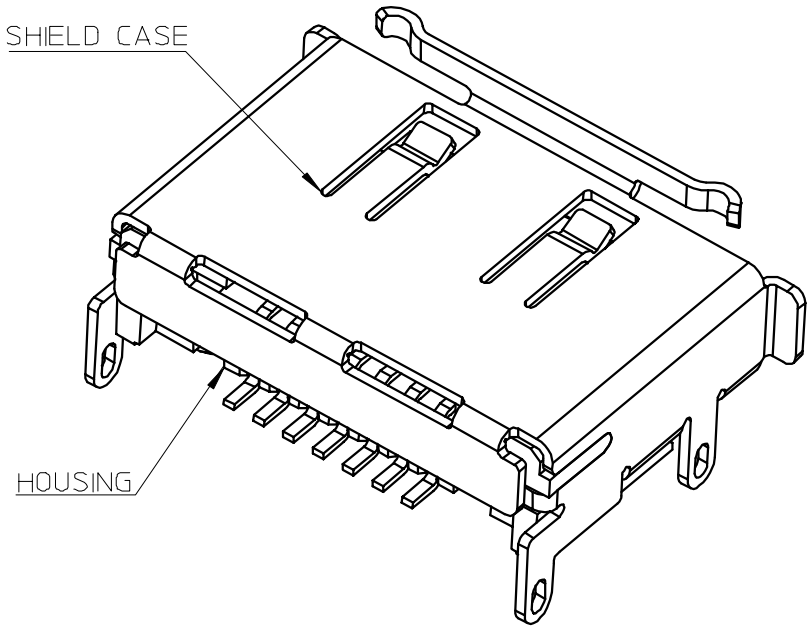

Figure 3.7. Surface Mount External SATA connector
Source : Molex Incorporated

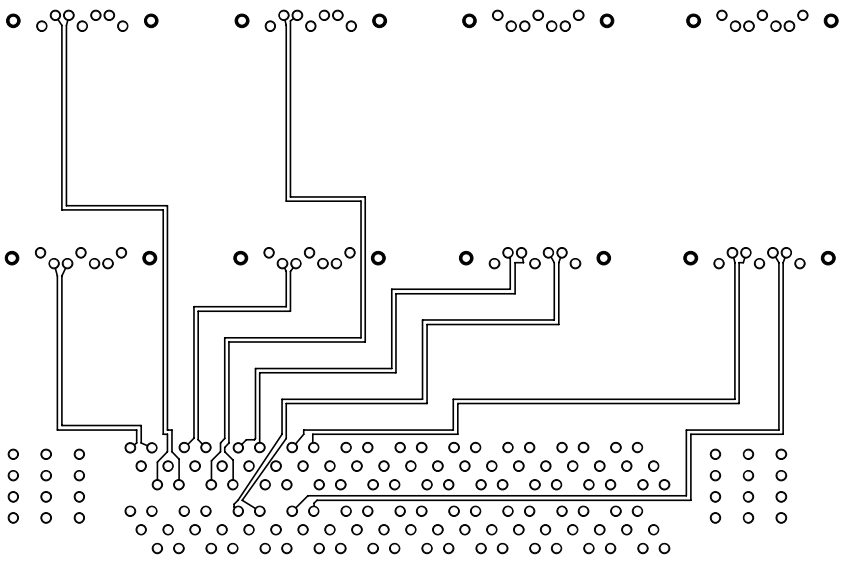

Figure 3.8. First Revision Board Top Layer

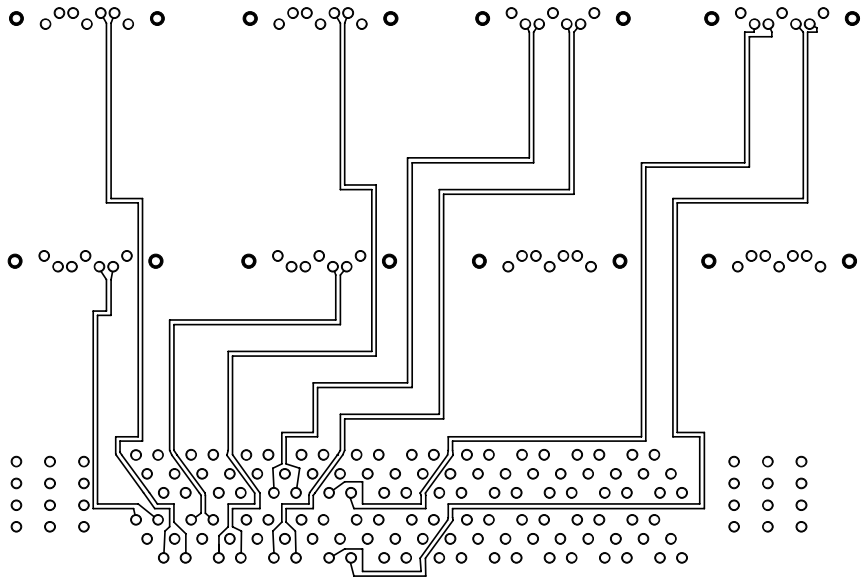

Figure 3.9. First Revision Board Bottom Layer

at that point. This can lead to reflections on the line.

The third flaw of the design was the breakout or fan out of the traces from the Z-Dok+ connector. The first design did this in a rather *ad hoc* method which lead to differential trace pairs not coming together as soon as possible as shown in Figure 3.9. This again will lead to impedance mismatches and reflections on the transmission lines. Preliminary tests of this board showed a Bit Error ratio of $1 \times 10^{-}6$, which is unacceptable. A second revision of the board was needed to address these problems.

3.2.6 Second Revision Networking Board

A second revision of the networking board was built after it was discovered that the first version was giving very poor bit error ratio performance. The top and bottom layer of the second revision board are shown in Figure 3.10 and

Figure 3.11. The second revision uses 45 degree bends, a five mil thick dielectric, and a much more organized fan out from the Z-Dok+ connector. The Xilinx user guides for the RocketIO MGTs on the Virtex-II Pro and Virtex-4 FPGAs provide a very good reference for specific PCB design guidelines regarding the transmission lines [12] [13], including trace length matching, routing, signal coupling, and via dimensions.

The second revision of the board was also necessary to match the chassis which had then been chosen for the RCC. The SATA network is routed through the front of the chassis, which required the use of a slightly larger board, and right angle SATA connectors. The second revision board also added several debugging ports to the board. These ports interfaced a SATA connector to four SMA connectors. The initial design was to use a network analyzer to study the transmission lines if the this board design also failed. However, the analysis would either require a four port network analyzer with a bandwidth up to 4.2 GHz, or a balun with bandwidth to 4.2 GHz. Both of these tools were unavailable so a direct measurement of the transmission line performance was not possible. We can only infer if the transmission lines are within specification if the bit error ratio is sufficiently low.

The board was manufactured by Hughes Circuits and assembled in the Instrument Design Laboratory at KU. Hughes Circuits etched the board using a controlled impedance process which periodically measures the impedance of the line during the etching process. The final impedance for all traces on the board was measured by Hughes as 97Ω.

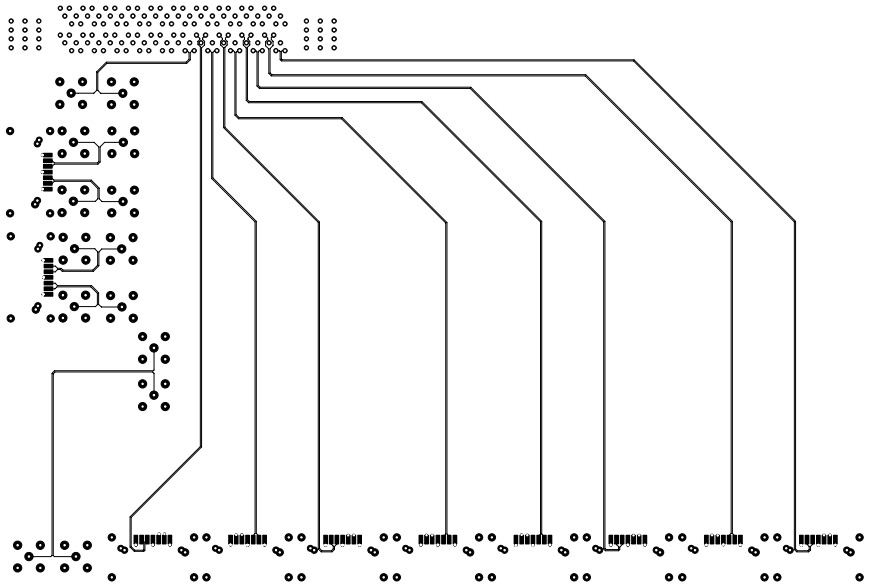

Figure 3.10. Second Revision Board Top Layer

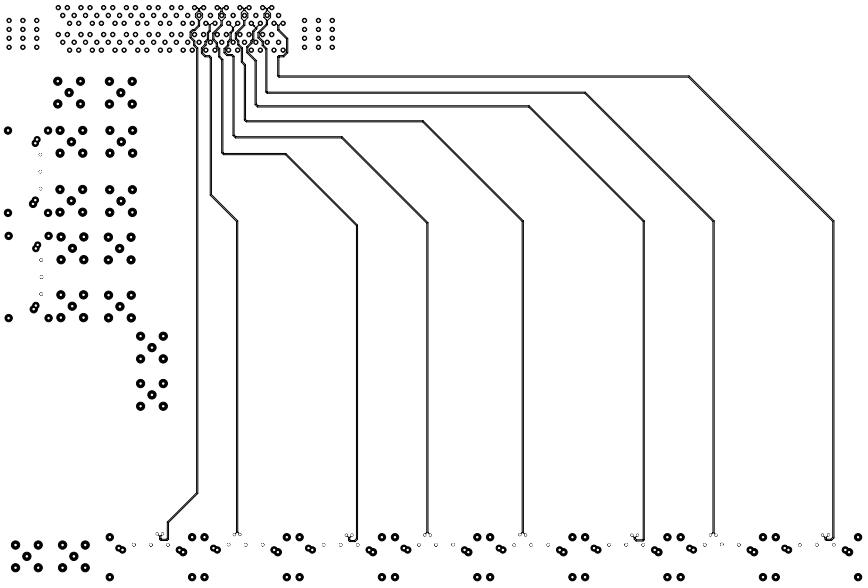

Figure 3.11. Second Revision Board Bottom Layer

Chapter 4

Evaluation

Ideally, the best way to evaluate the network performance is by using real applications; however, since we need to decide if SATA is feasible *before* we build the cluster, we need artificial metrics. A common metric for communication networks is to measure the number of bit errors. The most common method in HPC community is to use what are termed "micro-benchmarks" [5] that test bandwidth and latency of the message passing interfaces. We can directly measure the number of bit errors and use that, along with other measurements, to calculate the bandwidth and latency of a SATA based network.

This chapter details a number of experiments designed to test the feasibility of the SATA network. In section 4.1 we describe the experimental set-up for measuring bit errors; in section 4.2 we present the results from those experiments; in 4.3 we describe the latency and bandwidth for both a packet-based and stream-based protocol.

Bandwidth and latency measurements will be taken for both streaming and framing based networks. These numbers will show the characteristics of the network at the link layer. Applications on top of this network will see greater latency

and lower bandwidth than these numbers will show.

4.1 Bit Error Ratio Test

Xilinx has developed several bit error rate testers which can be implemented on an FPGA with MGT transceivers. [21] and [22] provide a bit error ratio tester (BERT) for the Xilnx ML-32x development boards. However, the BERT implemented in these application notes is specific to the ML32X board and the V2P50 or V2P70 FPGA. The Xilinx BERT implemented many features which made the process of porting the Xilinx BERT from the V2P50/V2P70 on the ML32x to the V2P30 on the ML310 very difficult, including partial reconfiguration of the RocketIO transievers to change the drive settings. The Xilinx BERT also only tests 1 channel at a time (2 transievers), which is not enough to test for crosstalk between channels. These limitions led to the development of a custom bit error ratio tester. The implemenation of that BERT is described in the following sections. The core of this BERT, in common with the Xilinx BERT, is a pseudo random data stream which is sent from 1 transeiver, over a cable under test, and recieved on a second transceiver. However, this BERT is modularized in such a way that all transcievers on an FPGA can be used to both send a pseudo random data stream and to count the number of errors it recieves on a second pseudo random data stream. This will allow the simultaneous test of all 8 MGTs present on the V2P30 to test for problems which may arrise from crosstalk between channels.

4.1.1 Pseudo Random Noise Generator

A common data set to use when testing is a string of pseudo random bits. These strings of bits are commonly generated by a linear feedback shift register

(LFSR). A linear feedback shift register (LFSR) is a shift register whose input bit is a linear function of its previous state [21]. The PRNG generator used in this experiment has its input bit be the exclusive or of its 14 and 15th bit in its current state. Figure 4.1 shows the LFSR implementation used in this test. This method of PRNG generation is a common method for testing bit error ratio's in networking.

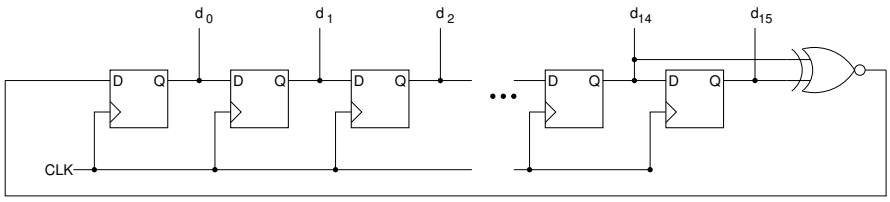

Figure 4.1. LFSR Implementation

An Aurora core was set up to provide a simple streaming interface and clock correction sequences for this BER test. The Aurora cores were set up to take in 16 bits of data every 6.4 ns ($\frac{1}{156.25MHz}$), and then pass the data on, along with all of the necessary control signals, to the MGTs. There were no bonded channels in this test.

A LFSR implemented on the FPGA fabric available on the Virtex2 Pro FPGA can not run at 3.125 GHz, which would be required to directly generate the bit stream with 1 LFSR and only taking the 1 output bit, which is the standard operation. The FPGA logic simply can not run at that high of clock rate. To solve this problem, two solutions were possible: have a dedicated LFSR generating a data stream for each of the 16 bit wide input bus, or use 16 bits from the internal state of the shift register in parallel to feed the 16 bit input bus. The first option would generate a more random bit stream, however the extra overhead of initializing each bit independently, and keeping track of 16 separate generating

and following LFSRs would be more work than is necessary. The second option has a disadvantage that each 16 bit word transmitted in the network is very similar to the previous word, it is just the previous word shifted by 1 place, and 1 new bit is inserted to the sequence. However, this option is much simpler to manage - only 1 LFSR is needed on the transmitting and receiving side.

There are some doubts if this method of parallel output from the LFSR generates the same quality bit stream as a single LFSR running at the much higher clock rate. However, if you consider the output of the serial LFSR as a very long stream of bits, then the output of the parallel LFSR is simply a sliding window along this stream. This window moves 1 bit every clock cycle input to the parallel LFSR. The output of the parallel LFSR is just 16 bits from the long sequence of the pseudo random string that the serial LFSR is generating.

4.1.2 Bit Error Ratio Tester

The LFSR described in the previous section forms the core of the of the bit error ratio tester implemented for this experiment. There are two of these LFSRs for each direction for each network link tested. One LFSR generates the test pattern and sends it to the transceiver via Aurora. The second LFSR receives the beginning of the pattern stream, latches it into its shift register, and the begins to generate the same pattern. Because each LFSR has an identical feedback circuit, and that they are now synchronized to the same place in the pattern, the output from the LFSR on the receiving side, and the output from the Aurora core, also on the receiving side, should then be identical. A simple error counter counts the number of times these two values were not identical. This value is read out either by LEDs on the ML-310 board, or though the debugging interface Chipscope.

Figure 4.2 shows the architecture of the BER core used in these tests.

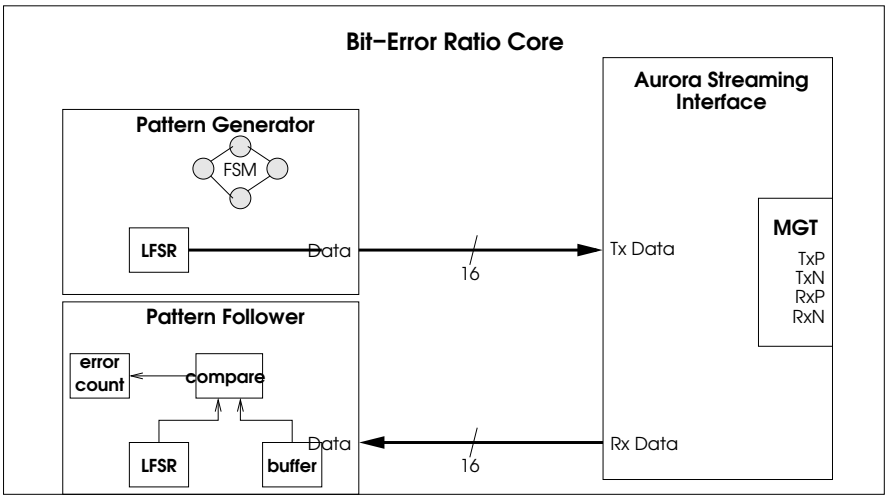

Figure 4.2. BER Core Implementation

The output of the BER simply states the number of errors it has encountered so far.

4.2 Bit Error Ratio Results

4.2.1 Testing Parameters

There are many options available when configuring the Aurora cores. Each aurora core is configured to take in 16 bits of data on every reference clock cycle. The reference clock for a 3.125 Gbps channel is 156.25 MHz, and 125 MHz for a 2.5 Gbps channel. The termination voltage was set to 2.5 V, which is stated as the ideal termination voltage for the case of two identical transceivers communicating with each other [12]. The differential output voltage was set to 800 mV for all tests. All BER tests have 2 transceivers active - both are sending and receiving a test stream. All tests were ran for 1 hour.

4.2.2 Length Tests

The following table shows the minimum (best) BER which was achievable for various lengths of SATA cables. Notice that the transceivers were able to run at the full bit rate for cables up to 5 meters in length. The transceivers had to be slowed down to 2.5 Gbps for 7.5 and 10 meter cables.

Table 4.1. Bit Error Ratio Verses Cable Length

Length (meters)	Pre-emphasis	Bit Rate (Gbps)	Errors	BER
0.5	1	3.125	0	4.44444E-14
1	2	3.125	0	4.44444E-14
2.5	2	3.125	0	4.44444E-14
5	3	3.125	0	4.44444E-14
7.5	3	2.5	0	5.55556E-14
10	3	2.5	0	5.55556E-14

4.2.3 Crosstalk Tests

Crosstalk can be a very serious issue, which could degrade our BER for a heavily loaded cluster. This could certainly be an issue with for SATA cables, which are not well shielded. The primary mechanism for crosstalk in this system is the mutual capacitance and inductance between separate channels.

The crosstalk test setup is very similar in setup to the previous length test. The differences are that only 1 meter cables were tested, as they will be the only cables needed for dense network architectures like a 3d-torus. In this test, all 8 transceivers are used on 1 ML-310. 4 1-meter SATA cables are used to connect the 8 transceivers. The 4 cables were arranged in parallel to maximize the capacitive and inductive coupling. These tests were ran for 3 days.

The first crosstalk test was ran with the pre-emphasis setting of 2. 106 errors were found during the course of this test, however all 106 errors occurred on 1

Table 4.2. Crosstalk test for 1 meter Cables

Length (meters)	Pre-emphasis	Bit Rate (Gbps)	Errors	BER
1	2	3.125	106	1.65123E-14
1	3	3.125	0	4.62963E-16

channel. This error prone channel is the channel which has the longest traces on the board. There can be two possible explanations for this result: crosstalk or dielectric losses. Two tests were ran to determine the mechanism.

First, the crosstalk test was reran with the pre-emphasis setting of 3. This test was successful which suggests that board loss was the primary loss mechanism. If crosstalk was the mechanism then increasing the signal strength of both the transmitter and receiver would not have an effect.

Second, the length test was ran again, but only using the channel which showed the problem with the pre-emphasis setting of 2. This test had 1 errors in 30 minutes while all of the other channels are idle. The error is that the transmitted hex string "5860" is always received as "58F9". This shows that dielectric losses, and not crosstalk were the sources of the errors in Table 4.2.

4.2.4 Board to Board Tests

So far all of the tests have been completed on a single ML-310. However, in a real cluster a node would never be connected to itself except for testing. Tests must be done to see how well nodes can communicate with other nodes. While all nodes will have the similar 156.25 MHz reference clocks, these clocks are not perfect and may drift, causing bit errors.

There are several mechanisms allready in place to correct for drifting clocks. Each MGT extracts a reference clock from the bit stream it receives. 8B/10B encoding is used to ensure that sufficient bit transitions are present in the stream

to allow for clock extraction. The received data as it is presented from Aurora is clocked into a register using the received data clock, not the local reference clock. Also, clock correction sequences are transmitted every 10,000 cycles to further ensure proper clock synchronization between nodes.

The experimental setup for the board to board test is identical to that of the crosstalk test, except that 2 ML-310s are used instead of 1. All 8 transceivers are active on both boards. 8 1-meter SATA cables are used. This test was ran for 24 hours.

Table 4.3. Board to Board tests for 1 meter Cables

Length (meters)	Pre-emphasis	Bit Rate (Gbps)	Errors	BER
1	3	3.125	0	3.08642E-16

4.3 Latency and Bandwidth Tests

The BER of a network link is independent of the application running through it. However, the latency and bandwidth which the application will see is dependent on the application itself, and the inherit latency and bandwidth of the network. The network latency and bandwidth was tested using one stream, and one frame based example application. The bandwidth and latency of the network itself, not the application is reported below. This is done by measuring these metrics at the input of the Aurora core, before the data has been processed by the application.

The streaming application used in these tests is simply the BER test. It utilizes the stream based Aurora core. In this case the latency is measured by comparing data streams being transmitted and received by a single BER core in loop-back mode by counting the number of clock cycle difference between a single word in the the sent and recieved data stream.

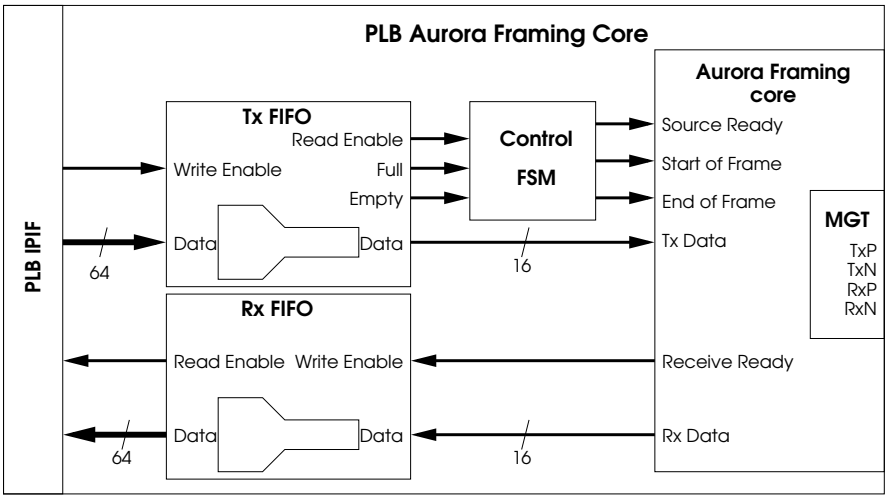

Figure 4.3. Example Framing Application

The framing application used in these tests is an example of how a software application would send data to the MGT. The block diagram for this application can be seen in Figure 4.3. In this example the CPU fills a the FIFO over the PLB. The framing Aurora core begins to send the data when the TX FIFO is full. The receiving core then pushes data into the RX FIFO as it is received. The CPU can then read out the contents of the RX FIFO via the PLB. Latency is defined in this system as the number of clock cycles it takes from when the transmitting Aurora core asserts the "start of frame" signal to when the receiving core sees that same "start of frame" signal.

4.3.1 Latency

The latency of the streaming Aurora core was found to be 38 clock cycles, where the clock frequency was 156.25 MHz. The framing Aurora core shows 40 clock cycles of latency. These measurements were completed using the serial loop-back on a single transceiver and Chipscope to directly measure the latency. The

loop-back path includes the latency for all of the aurora streaming logic, as well as all of the MGT transceiver latency. See Figure 3.2.1 for the diagram of the MGT transceiver and serial loop-back path. The cable delay for a 1 meter internal SATA cable is 4.9 ns per meter [7].

Table 4.4. Point to Point Latency

Mode	Aurora → Aurora (ns)	1 meter Cable (ns)	Total (ns)
Streaming	243.2	4.9	248.1
Framing	256	4.9	260.9

4.3.2 Bandwidth

The effective bandwidth of a network link is defined as the raw bit rate of the link multiplied by the efficiency of that link. The efficiency of a link is defined as the ratio of correct user data sent to all data sent over the link. The efficiency of the link is the product of efficiencies for each component of the network.

4.3.2.1 Framing Efficiency

In finding the effective bandwidth of the framing aurora link, four efficiencies were used: each characterizes different parts of the system. The efficiencies used are for the framing aurora module, the header efficiency, the packet error and retransmit efficiency, and the 8B/10B encoding efficiency.

The efficiency of the Aurora framing module is driven by the overhead of sending the start of frame, and end of frame bytes down the link, as well as the bytes inserted for clock correction. This efficiency is given in the following equation, where m is the length, in bytes, of the entire packet, including the header. Four byte overhead is due to sending the 2 byte "start of frame" and 2

byte "end of frame" for every frame on the link. The $12 \times m/9988$ factor represents the overhead of sending 12 clock correction bytes every 10,000 bytes sent.

Framing Aurora Efficiency: $m/(m + 4 + 12 \times m/9988)$

The efficiency of the header is defined as $n/n + h$, where n is the length of the user data to be sent in the packet, and h is the length in bytes of the header that must be sent with the data for control data, such as the sender, receiver, and check-sum for this packet.

4.3.2.2 Streaming Efficiency

The streaming aurora core does not send start or end of frame signals. It does however, still send 12 clock correction bytes every 10,000 Bytes. This means that on a streaming interface, 9988 user data bytes are sent for every 10,000 bytes which are actually sent over the network.

Streaming Aurora Efficiency: $\frac{9988}{10000}$

4.3.2.3 8B/10B Encoding Efficiency

Both framing and streaming Aurora cores use 8B/10B encoding. 8B/10B encoding takes 8 user bits and then encodes them into a 10 bit sequence. It is frequently used in high speed communication systems to ensure a sufficient number of bit transitions to keep the transmitter and receiver synchronized. The efficiency of 8B/10B encoding is 8/10.

4.3.2.4 Packet Retransmit Efficiency

The efficiency of retransmitting a packet that was lost due to a bit error is defined as the number of packets that are sent over a link by that same number,

plus the number of packets that will need to be retransmitted because of a bit error. This assumes that whenever a packet is effected by a bit error, the entire packet will be thrown away, and that the new packet will be retransmitted perfectly.

The bit errors are assumed to be randomly spaced - not tightly clumped. When an error occurs in a packet the entire packet will be dropped. The average number of good packets transmitted for every 1 bad packet can be found by 1/(Packet length in bits × BER), assuming perfect retransmission of the bad packet. For the BER found above in the cross talk test of 4.6E-16, this ratio of good packets to bad packets is 4.1E9.

4.3.2.5 Bandwidth Calculations

Table 4.5 shows the calculations completed to find the bandwidth of for both streaming and framing based networks.

Table 4.5. Bandwidth Calculations

User Data Length (n) =	8160 Bytes
Header Length (h) =	32 Bytes
Packet Length (m) =	8192 Bytes
Framing Aurora Efficiency =	$\frac{8160}{8160+4+12\times 8160/9988}$
=	0.9983
Streaming Aurora Efficiency =	$\frac{9988}{10000}$
=	0.9988
Header Efficiency =	8160 Bytes / 8192 Bytes = 0.9960
8B/10B Efficiency =	8 bits / 10 bits = 0.8
Packet Retransmit Efficiency =	$\frac{4.1\times 10^9}{4.1\times 10^9+1} = 1$
Streaming Overall Efficiency =	$0.9988 * 0.8 * 1 = 0.7990$
Framing Overall Efficiency =	$0.9983 * 0.9960 * 0.8 * 1 = 0.7954$
Streaming Effective Bandwidth =	$3.125 \times 0.7990 = 2.497$ Gbps
Framing Effective Bandwidth =	$3.125 \times 0.7954 = 2.486$ Gbps

Chapter 5

Conclusion

Serial ATA cables are a feasible cable to use in a high performance computing cluster.

The design of a HPC cluster based on Xilinx FPGA nodes and Serial ATA cables between the nodes was tested. A custom networking board was developed for the Reconfigurable Computing Cluster (RCC) which integrated a Serial ATA network with the 64 ML-410 nodes in the cluster. This board had to follow strict rules of high speed digital design to get a workable solution.

The Serial ATA cables showed very good bit error ratio — no errors were seen provided that the network was set up properly. Cables were tested up to 10 meters in length, with 5 meter cables being the maximum length of cable which will support the full bit rate of 3.125 Gbps. The link layer protocol Aurora was used to measure the latency and bandwidth. The latency from Aurora to Aurora is approximately 300 ns for both streaming and framing based networks. The overall effective uni-directional bandwidth is approximately 2.5 Gbps.

5.1 Future Work

5.1.1 Revision 3

A third revision of the networking board is needed. This board will include a USB interface which will allow for JTAG programming as well as serial port UART readout from each node in the cluster. The previous plan was to have no JTAG programming ability for a node in the cluster, and to use an off-the-shelf RS-232 to USB converter to access the serial port output of the FPGA node. The third revision board will incorporate this functionality and add JTAG programming. The $6 additional cost of the dual purpose UART/JTAG chip to the networking board is much less than the $20 for the RS232-USB converter.

The third revision will also reflect yet another change in chassis for the RCC. The decision was made to use the Xilinx MLX10 [23] chassis, which is a 19" rack-mount chassis specially designed for either the ML-310 or ML-410 FPGA board. The networking board must physically change in size to fit where the MLX10 chassis has designated for "Personality Modules" — meaning anything that connects to the Z-Dok+ connector. The standard "Personality Module" is about 4 inches longer than the second revision of the networking board. However, it was shown for the second revision that the some signals were already too long on board.

5.1.2 Network Router

A key idea of the reconfigurable computing cluster which we need to test is to integrate the network router onto the FPGA. This network router needs to be developed. The primary constraints are that it must have 8 ports to connect to all 8 transceivers on chip, and that it consume few resources on the FPGA. The great

possibility of the RCC cluster will be completely eliminated if the network router consumes the entire FPGA — leaving no room for FPGA based acceleration cores.

References

[1] "Ml410 virtex-4 fx embedded development platform," Xilinx, url: http://www.xilinx.com/ml410.

[2] P.-Y. Droz, "Physical design and implementation of BEE2: A high end reconfigurable computer," MS Project, University of California at Berkeley, Department of Electrical Engineering and Computer Sciences, December 2005.

[3] B. Noseworthy, "Gigabit ethernet 1000base-t focus : Physical coding sublayer," 1998. [Online]. Available: http://www.iol.unh.edu/services/testing/ge/training/1000BASE-T/pcs.pdf

[4] *CAT5, CAT5E, CAT6 Cable Characterization with RocketIO MGTs*, Xilinx.

[5] J. Liu, B. Chandrasekaran, W. Yu, J. Wu, D. Buntinas, S. Kini, D. K. Panda, and P. Wyckoff, "Microbenchmark performance comparison of high-speed cluster interconnects," *Micro, IEEE*, vol. 24, no. 1, pp. 42–51, 2004.

[6] *Infiniband Cable Characterization with RocketIO MGTs*, Xilinx.

[7] A. SPECTRA-STRIP, *2 pair 26 AWG 100 ohm internal SATA*, url: http://www.spectra-strip.amphenol.com/150-2699-947RevJ.pdf.

[8] ——, *2 pair 26 AWG 100 ohm 1X External SAS / SATA*, url: http://www.spectra-strip.amphenol.com/160-2699-994Rev2.pdf.

[9] ——, *2 pair 26 AWG 100 ohm IB, SFP*, url: http://www.spectra-strip.amphenol.com/160-2699-997Rev3.pdf.

[10] *Serial ATA Cable Characterization with RocketIO MGTs*, Xilinx.

[11] R. Sass, W. V. Kritikos, A. G. Schmidt, S. Beeravolu, P. Beeraka, K. Datta, D. Andrews, R. S. Miller, and J. Daniel Stanzione, "Reconfigurable computing cluster (rcc) project: Investigating the feasibility of fpga-based petascale computing," in **to appear in** *FCCM '07: Proceedings of the 13th Annual IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM'07)*. Washington, DC, USA: IEEE Computer Society, 2007, p. ???

[12] "Rocketio transceiver user guide," Xilinx, url: http://direct.xilinx.com/bvdocs/userguides/ug024.pdf.

[13] *Virtex-4 RocketIO Multi-Gigabit Transceiver User Guide*, Xilinx, url: http://direct.xilinx.com/bvdocs/userguides/ug076.pdf.

[14] "Ml310 embedded development platform," Xilinx, url: http://www.xilinx.com/ml310.

[15] "Serial ata," url: http://www.serialata.org.

[16] "Aurora link-layer protocol," Xilinx, url: http://www.xilinx.com/aurora.

[17] *Virtex-II Pro / Virtex-II Pro X Complete Data Sheet*, Xilinx, url: http://direct.xilinx.com/bvdocs/publications/ds083.pdf.

[18] H. Johnson and M. Graham, *High-Speed Digital Design: A Handbook of Black Magic*, 1st ed. Prentice Hall PTR.

[19] Agilent, "Agilent advanced design system (ads)," 2007. [Online]. Available: http://eesof.tm.agilent.com/products/ads_main.html

[20] T. Electronics, "Overview for z-dok," 2007. [Online]. Available: http://zdok.tycoelectronics.com

[21] *RocketIO BERT Reference Design User Guide*, Xilinx, url: http://direct.xilinx.com/bvdocs/userguides/ug064.pdf.

[22] *XAPP661 - RocketIO Transceiver Bit-Error Rate Tester*, Xilinx, url: http://direct.xilinx.com/bvdocs/appnotes/xapp661.pdf.

[23] *MLX10 Rack-Mount User Guide*, Xilinx, url: http://www.xilinx.com/bvdocs/userguides/ug095.pdf.