System–Level Test Interface Considerations for High Speed Interconnect

Man Wah Chiang

Department of Electrical and Computer Engineering

McGill University, Montreal

February 2004

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Engineering

Your file Votre référence
ISBN: 0-612-98519-9
Our file Notre référence
ISBN: 0-612-98519-9

Abstract

Although modern system operating speed and performance largely benefit by the advanced technology, designing the system level test circuitry is still a great challenge. Traditional ad-hoc or informal design practice is not adequate, and an early well planned testing scheme is required to meet the concerns raised by the increasing size and complexity of the system. Moreover, many high speed communication issues including BER testing, jitter measurement and jitter injection also need to be considered due to the increasing need for newly developed high speed interconnect architectures such as PCI Express, Fibre Channel and InfiniBand. In this thesis, a disciplined test interface design methodology based on the layered model is presented in order to ease the test circuitry design by maximizing the flexibility and reusability. Instead of setting up a completely new test access channel, test circuitry is integrated into each functional layer of the system to reuse the existing communication mechanisms. InfiniBand and Sensor Networks are used to demonstrate the proposed design methodology and a new BER testing scheme is also discussed.

Résumé

Bien que la vitesse opérative et le fonctionnement du système moderne sont en grande mesure améliore par la technologie, concevoir un niveau système d'examen de circuit est tout de même un grand défi. La conception traditionnelle et familière n'est pas satisfaisante. Ainsi, un meilleur plan doit être élaboré pour rencontrer l'augmentation de grandeur et de complexité du système. De plus, il faut considérer les questions de haute vitesse comme l'examen BER, la mesure frousse et l'injection frousse à cause de leur architecture interconnective, par exemple PCI Express, Fibre Channel et InfiniBand. Dans cette thèse, une méthodologie pour tester, interface basé sur le modèle, est presenté pour faciliter le dessein de la conception de circuits visant à améliorer leur flexibilitéet et réutilisabilité. Au lieu d'installer un canal complètement nouveau, l'examen de circuit est integer à la couche fonctionnelle du système, ce qui permet de réutiliser les mécanismes existants de communication. InfiniBand et les réseaux de sonde sont employés pour démontrer la méthodologie de conception proposée. Finalement, un plan pour tester le nouveau BER est aussi discuté.

Acknowledgements

Lots of people have helped me throughout my thesis work and I would like to acknowledge them for their support and assistance.

Firstly, I would like to thank my supervisor, Professor Zeljko Zilic for his great support and guidance throughout my thesis work. His great academic knowledge and industrial experiences provide a good vision in the research topics. Without his patience and suggestions, it is impossible to complete this work. He also provided an excellent research environment including financial aid, up-to-date software which allowed me to focus on my research and under his great supervising, I really learn a lot of things that will definitely benefit in my future career development.

Secondly, I would like to thank to my family who supported me throughout my studies and had patience to endure my second scholastic stay. I really thanks for their support and understanding of my decision of quitting the job and continuing my master studies.

Lastly, I would like to thank my colleagues Yongquan Fan for his assistance in the BER testing area and all my friends in the Macs Lab who bring me the enjoyment

and happiness during my master studies in McGill.

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 Overview

With the increasing complexity of modern systems, testing becomes a fairly difficult task. According to the International Technology Roadmap For Semiconductors (ITRS) [1] produced in 1997, the manufacturing cost is projected to be gradually decreasing due to the improved manufacturing processes and technologies. However, the testing cost does not benefit from the advanced technology and is projected to remain constant as shown in Figure 1.1. With this trend, testing cost will be eventually equal to the manufacturing cost in 2010 and will become a major factor in the overall production cost. As a result, a well-planned and highly efficient system testing scheme is needed to lower the testing cost.

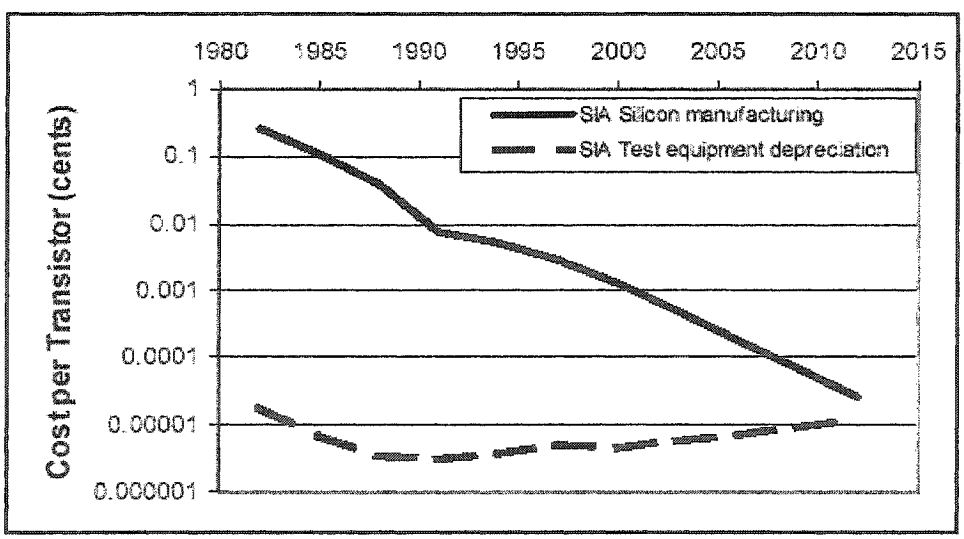

Figure 1.1 Silicon Manufacturing and Test Cost Trends Predicted in 2001 by ITRS [1]

Testing cost is always proportional to the complexity and size of the system. With the help of the rapidly advancing manufacturing processes, the transistor density of the chip is gradually increased. Designers can now put the whole System on a Chip (SoC). In addition, IP cores are usually used in the SoC design which makes the system-level testing more difficult. In order to ease the integrations of the system, designers usually include the Design For Test (DFT) features in their board-level and chip-level designs. Since most of these testing features are non-functional during normal operation, a system-level access channel is needed for test control. Test data also needs to be imported and exported to the target modules.

Notice that by decreasing the size of the transistors, we lower the power dissipation and increase the system operating speed. This trend influences the interconnect architecture development and many new high bandwidth, low latency interconnect architectures have been proposed. Traditional interconnect schemes use a tri-state buffer which shares a common bus among different modules. Typical such example is Peripheral Component Interconnect (PCI), which is one of the most popular interconnect architectures used in the computer systems. Due to its slow transmission speed, PCI becomes the bottleneck of the modern computer systems. Several upgraded versions of PCI, such as PCIX or PCI Express are developed to provide the higher transmission speed by adapting the serial packet based communication mechanisms. The main advantage of these schemes is their backward compatibility with the existing PCI components. It maximizes the reuse and decreases the development cost. On the other hand, many new interconnect architecture standards are also developed (such as InfiniBand and Fiber Channel) to overcome the speed problem. Among all these standards, InfiniBand seems to be the most promising architecture. It is largely supported and developed by the major computer industrial vendors such as Sun, IBM etc. Moreover, its high transmission speed and flexibility are achieved by its networked nature. Existing testing standards such as JTAG and MTM-Bus

become inadequate since they are mainly targeting the board level designs and are unsuitable for the coming complex networked systems.

Figure 1.2 shows a typical distributed measurement system. Depending on the applications domain, various interconnect architectures are used. Sensors are placed in the remote area to capture the environmental data. Low power RF transmissions are used to communicate with the control center to preserve energy. Control center is formed by several personal computers (PC) for data analysis and measurement control. Each of them interconnects through the Ethernet in order to form a Local Area Network (LAN). Ethernet is used because of its great flexibility and low setup cost. In order to provide a fast and reliable data storage, a file server is included in the system. File server is made up by interconnecting several storage devices including tapes and harddisks through Fibre Channel. The LAN is also connected to the Internet to allow user obtaining the measurement data through the wireless mobile devices such as PDA handheld or cellular phone.

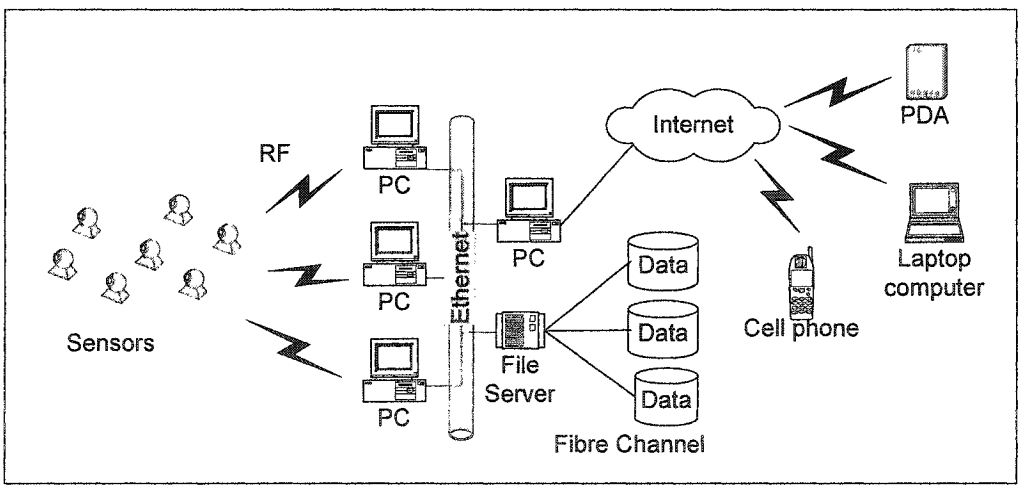

Figure 1.2 A Typical Measurement System

System-level testing is extremely difficult in such a system due to the existence of various interconnect architectures. Let us assume that a user is using the PDA handheld device and would like to test the sensors once the abnormal measured data is received. It is almost impossible to achieve this based on the existing

testing architectures. A new system-level testing scheme that can reuse the existing connection for test control is needed.

1.2 Motivations

As discussed above, the traditional interconnect architectures are starting to be replaced by the high speed point-to-point switching interconnect architectures. New system-level testing schemes need to be developed. As described in the 2001 ITRS, several major concerns are raised in the testing area as follows:

1. High frequency – As the new interconnect standards provide both the high bandwidth and high transmission speed, test equipment with low skewing and low latency is needed.
2. Port count – As the low-power gigabit IO becomes reality by the lower threshold voltage of CMOS, it enables massive integration into ASICs and SoCs. ASICs with more than 100 pairs of multi-gigabit transmitters and receivers are produced in 2002. Traditional rack model-based testing standards are not applicable due to the extensive wirings. New testing schemes should develop to address this issue.
3. Testing Cost – Both the high clock speed and large port count of the modern systems increase the testing cost due to the expensive test equipment and complicate test setup. An efficient and low cost testing scheme is needed.
4. Transparency – As the complexity of the system increases, various interconnect architectures may be used within the same system. System-level testing is only possible if the existing communication channels are reused for the testing purposes. All the testing circuitries should be transparent to the system during the normal operations.
5. DFT consideration – Currently, most of the high speed communication devices such as Serializers and Deserializers (SerDes) modules are equipped with the basic DFT features. They are based on the internal serial

and parallel data loopback mechanism. Additional DFT components such as build-in pseudo random bit sequence generator and error detector are needed to increase defect coverage.

6. Jitter - As the point-to-point switching mechanism is adapted in the new interconnect standards, some of the high speed communication concerns such as jitter measurement and jitter injection also need to be considered.
    a. Jitter Measurement – Existing jitter measurement instruments still fail to meet the noise floor, analog bandwidth and the test speed requirements of the high performance systems. On-chip DFT needs to be developed to address these issues and lower the testing cost.
    b. Jitter Injection – In order to accomplish the jitter measurement, well controlled jitter should be deliberately added to the data streams. Currently, there is no integrated ATE solution in this area due to the high speed requirements.

In this thesis, we addressed some of the above issues by introducing layered design methodology for the system level test interfaces. By doing so, a low cost, high flexibility and low latency test interfaces can be achieved. Moreover, a new FPGA-based BER testing scheme is introduced and used for jitter measurement.

1.3 Thesis Outline

In Chapter 2, the background of the system testing is described. Both new and old system interconnect architectures such as PCI, PCI-X, PCI Express, AMBA and InfiniBand will be introduced.

Chapter 3 is focused on applying the layered design methodology for designing the system-level test interface. Existing test architectures such as JTAG, MTM-Bus and P1500 are introduced. Their main advantages and disadvantages are also discussed. The basic concept of layered approach is presented and demonstrated by applying it to the InfiniBand architecture.

In Chapter 4, the proposed test interface is studied and its usage is expanded in other areas such as programming and instrumentations. In order to show the generalization of the new design approach, a distributed system such as sensor network is discussed. The possibility of applying the layered approach to design the system-level test interface to the sensor networks is also examined.

In Chapter 5, the communication issues that are raised by the new interconnect architecture are described. General BER measurement, jitter measurement and jitter injection are discussed. A new FPGA-based BER testing scheme is introduced and we validate its performance by performing a baseband transmission test. Based on this new BER testing scheme, a low cost and efficient jitter measurement test setup is proposed. The necessary modifications of the FPGA-based BERT [57] to facilitate the jitter measurement are discussed.

Finally, the conclusions and the future work are presented in Chapter 6.

Chapter 2 - Background

2.1 Overview

Before introducing layered approach for the system test interfaces, it is necessary to have an overview of the various system interconnect architectures. Figure 2.1 shows the conceptual block diagram of a computer system. It is made up of different functional modules that are interconnected by a common bus. In general, functional modules can be the Central Processor Units (CPU), Random Access memory (RAM) or I/O devices.

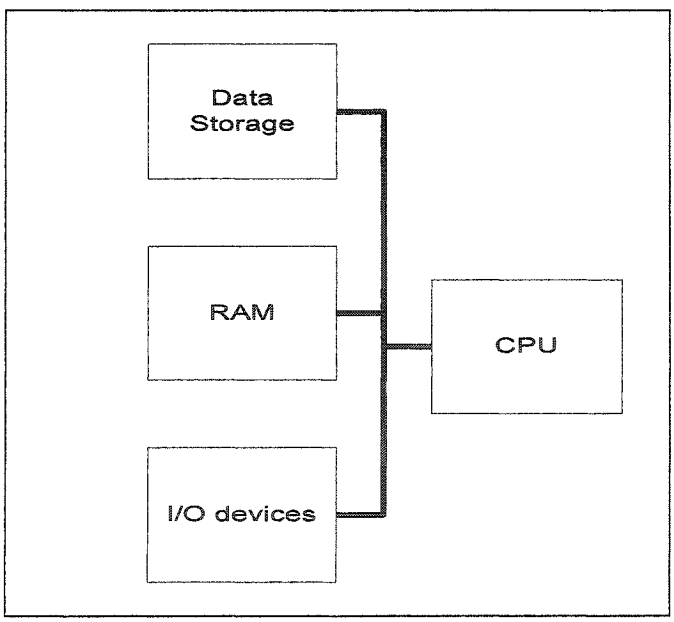

Figure 2.1 Block Diagram of a Typical Computer System

In the past, all these functional modules were the major limiting factor for the system performance. As now the CPUs have entered the gigaHertz era, the

bottleneck of the system performance has shifted to the interconnect. Industrial vendors have started optimizing the performance of the system by researching new interconnect architectures. Recently, many new interconnect architectures have been developed. Some are based on the existing widely accepted PCI architecture such as PCI-X and PCI-Express. On the other hand, some interconnects are based on the network architecture and use the point-to-point switching connection. By doing so, a direct and private communication channel can be established between the two connected devices and data can be transmitted at the maximum speed.

In this chapter, several popular system interconnect standards such as PCI, PCI-X, PCI Express, AMBA and InfiniBand will be introduced in Section 2.2, 2.3, 2.4 and 2.5 accordingly.

2.2 Peripheral Component Interconnect (PCI)

Since it was first time released by Intel in 1991, PCI [2] has evolved rapidly to replace the Industrial Standard Architecture (ISA) bus and become the most commonly used interconnect in the personal computer systems. Unlike the slow ISA bus, PCI is a synchronized parallel bus architecture which provides a maximum 33MHz transfer rate over a multiplexed 32-bit or 64-bit width data path. Figure 2.2 shows the block diagram of the PCI system. PCI bus provides a direct memory access to the connected devices and a bridge is used to connect the processor.

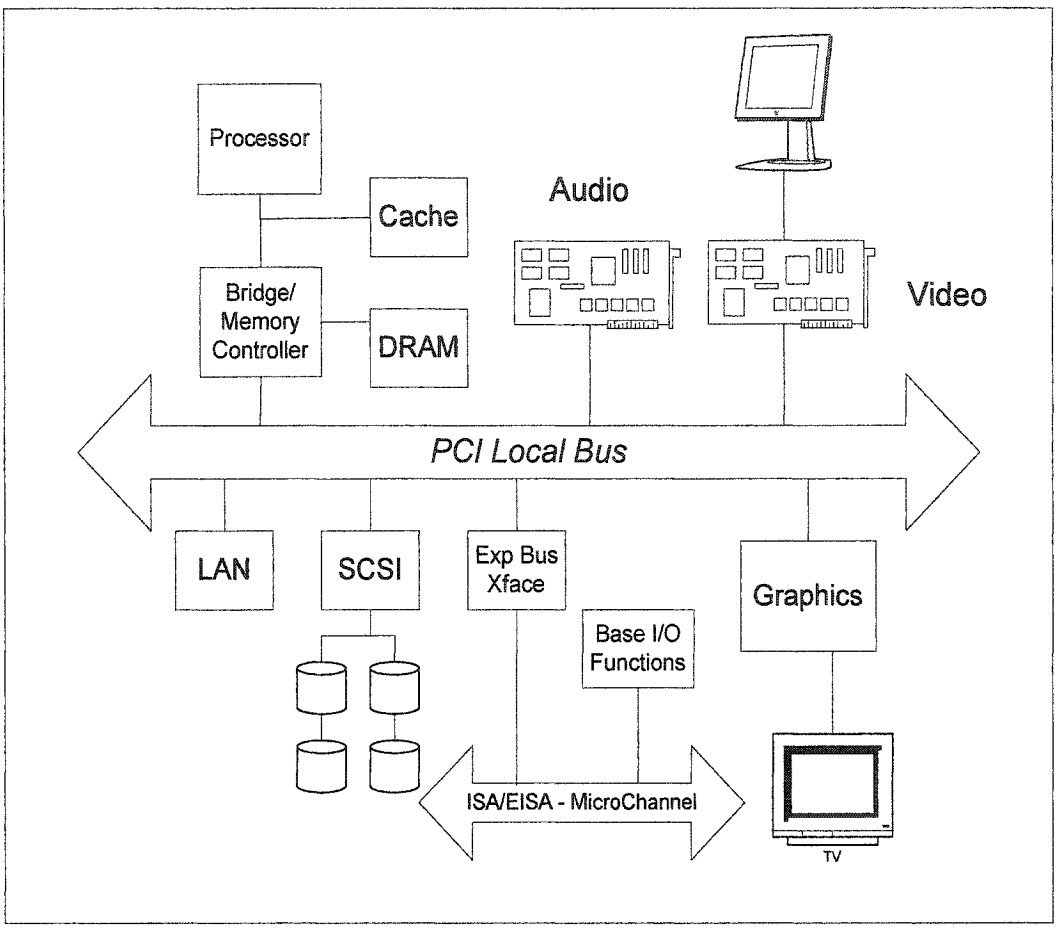

Figure 2.2 PCI System Block Diagram

Although PCI was first introduced in 1991, it did not gain the popularity until the arrival of the Windows 95 in 1995. This is mainly due to the fact that the new operating system supports the Plug and Play (PnP) feature that was standardized by Intel and implemented in PCI. With this feature, new hardware can be recognized and configured to work once it is inserted into the computer. In order to ensure that every PCI card is compatible with all PCI-enabled systems, PCI Special Interest Group (PCI-SIG) was formed to set, manage and develop the standard of the PCI bus.

In order to facilitate high computing power, several high speed versions of PCI standards were proposed. One of them is called PCI-X [3] and provides a maximum speed up to 4.3 gigabytes per second. Another new standard is PCI-Express [4] which uses the serial interconnect technology to offer a 250 megabyte

per second per lane in each direction. It provides a total bandwidth of 16 gigabytes per second for a 32-lane configuration.

2.3 AMBA

Advanced Microcontroller Bus Architecture (AMBA) [5] is an open standard, on-chip bus specification that details the modular interconnect in a System-on-Chip (SoC). As shown in Figure 2.3l, an AMBA based SoC system includes the high performance system bus – AHB and the low power peripheral bus – APB. AHB is responsible for connecting the processor, memory modules or any other devices that require high bandwidth usage. On the other hand, APB is responsible for all other devices which are less bandwidth demanding. APB's protocol is a simplified version of AHB and both can be communicated through a bridge.

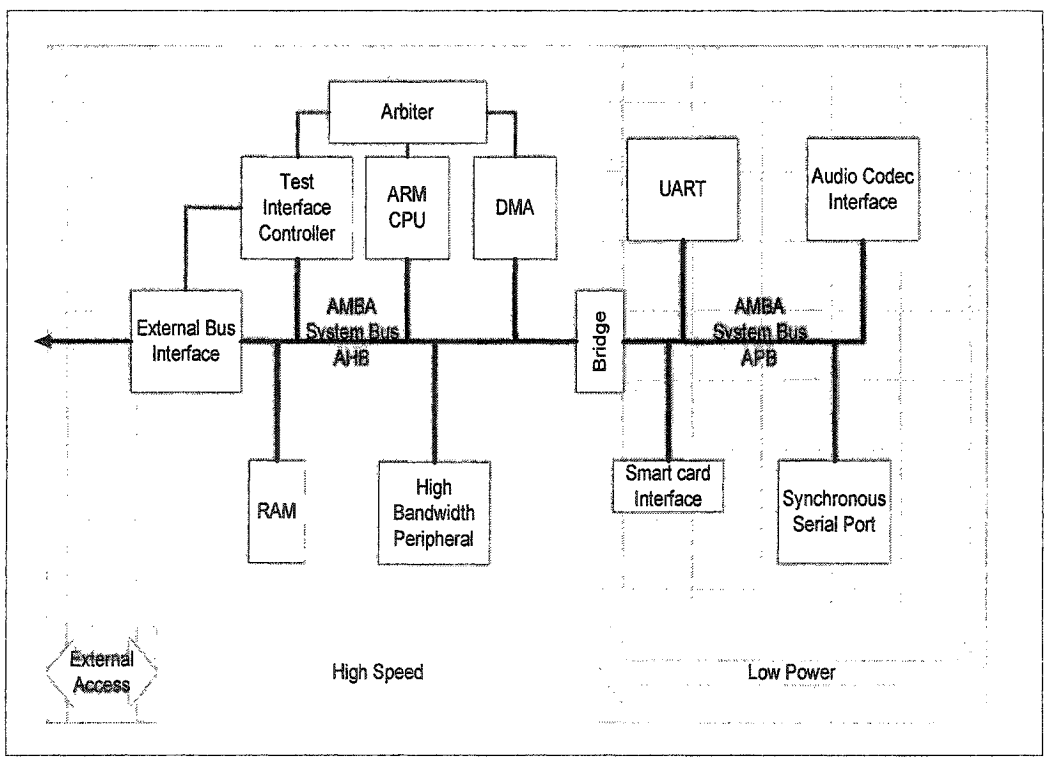

Figure 2.3 Basic SoC System Based on AMBA Bus

Unlike the traditional tri-state buffer bus architecture, AMBA uses the multiplexer

to gain the control of the bus. The multiplexers eliminate the timing issues and perform a one-to-one transaction as shown in Figure 2.4. During each transaction, one master module requests the access to the bus by sending a request command to the Arbiter. Arbiter controls the access of the bus based on the priority of the bus master module. Once the bus access is granted, an isolated transmission path is formed between the master and target modules and data can be transferred to the target module. Split transactions and burst transfers are also supported in the AHB to provide high transfer rate. The main difference between AHB and APB is that multiple bus masters are supported in AHB. Bridge is the only master module presented in APB and no arbiter is needed. Moreover, some advanced features like pipelined operations are supported in AHB, but are not included in APB.

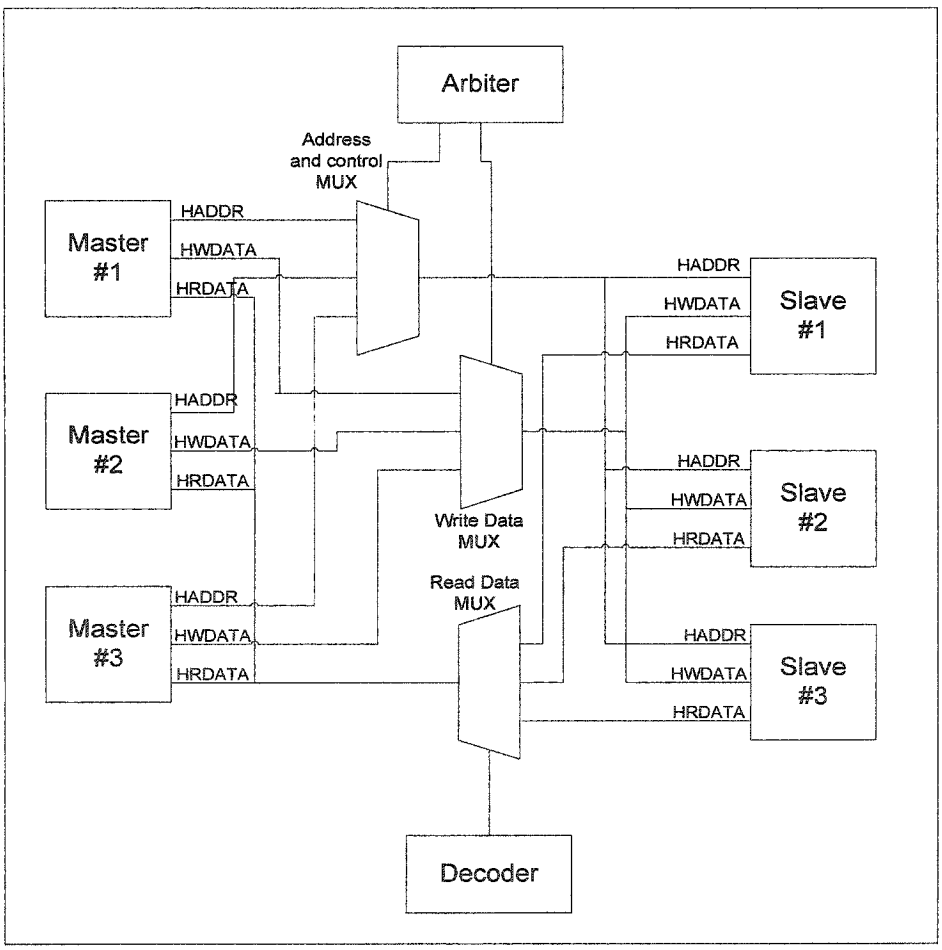

Figure 2.4 Block Diagram of the AHB

2.4 InfiniBand

InfiniBand Architecture (IBA) [6] is a new interconnect architecture. It is jointly developed by major computer industrial vendors aiming to replace the PCI bus architecture. IBA defines a System Area Network (SAN) that connects the nodes. These nodes can be independent processors, I/O units or routers to another network through switches, as shown in Figure 2.5. Each node contains one or multiple Channel Adapters (CAs) that connect to the fabric.

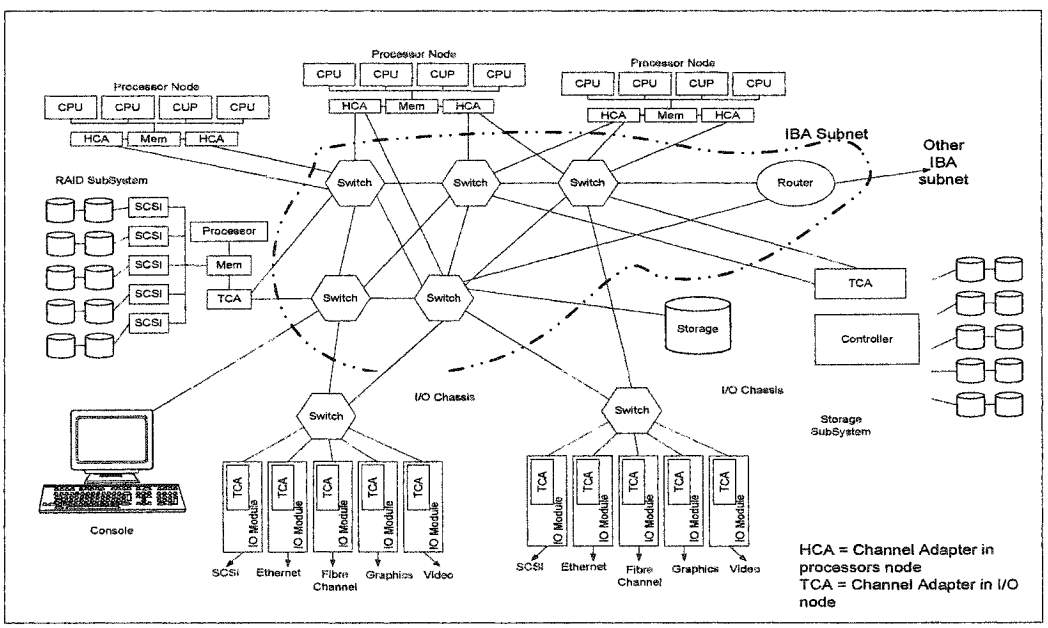

Figure 2.5 Block Diagram of the InfiniBand Network

Queuing mechanism is used in IBA to initialize the communication between nodes. Consumers can queue up a set of hardware-executed instructions by creating a Work Queue Pair (QP), one for each send and receive operation, as shown in Figure 2.6. During normal operation, consumers submit a Work Request that causes the creation of an instruction called a Work Queue Element (WQE) to be placed on the work queue. Once the channel adapter executes and completes a WQE, a Completion Queue Element (CQE) is placed on the completion queue.

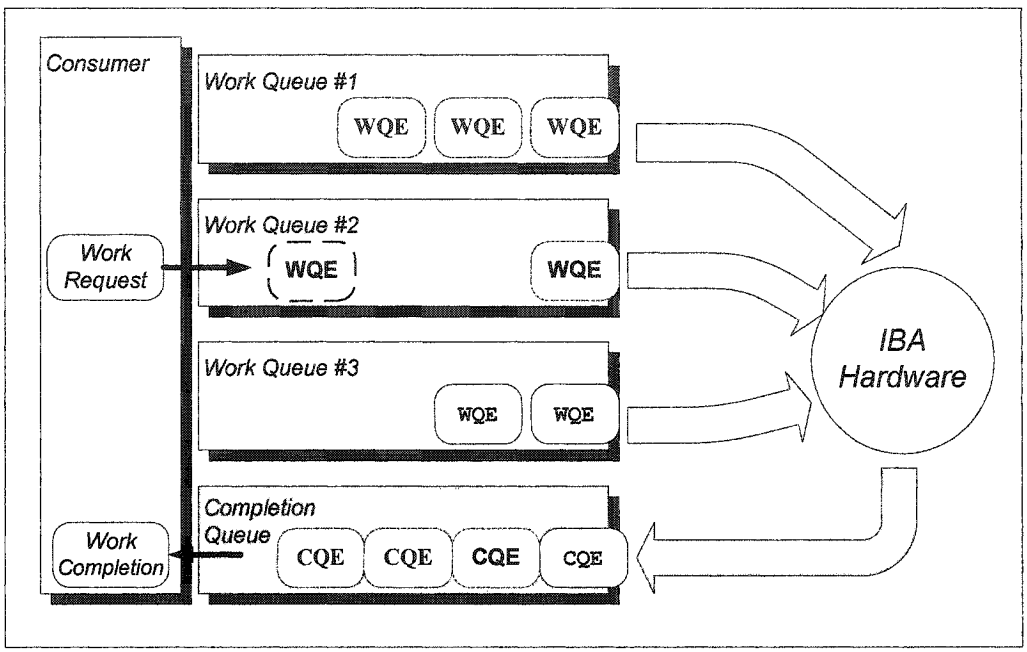

Figure 2.6 Consumer Queuing Model

Consumer posts the WQE to the QP and channel adapter interprets each WQE to perform the operation. InfiniBand supports three kinds of send queue operations, SEND, Remote Memory Access (RDMA) and Memory Binding.

- For a SEND operation, WQE specifies a block of memory in the consumer's memory space to send to the destination. In the destination, an already queued WQE will decide where to place the received data.
- For a RDMA operation, WQE also specifies the target memory address of the destination and the destination WQE is not involved. Three kinds of RDMA operations, RDMA-WRITE, RDMA-READ and ATOMIC are supported by IBA:
  - In the RDMA-WRITE operation, data is transferred from the consumer's memory to the destination consumer's memory address.
  - In the RDMA-READ operation, data is transferred from the target consumer's memory address to the local consumer memory.
  - ATOMIC operation allows performing a read operation to a remote 64-bit memory location. The target node returns the read value and updates the remote memory contents.

- For a Memory Binding operation, consumer defines the portion of the registered memory that is shared with the target nodes and allows the remote nodes to perform the RDMA operations. This information will pass through the network as the form of memory key which is called R_KEYS.

There is only one receive queue operation in IBA, called RECEIVE WQE. It is used to specify the memory location that the received data should be placed in.

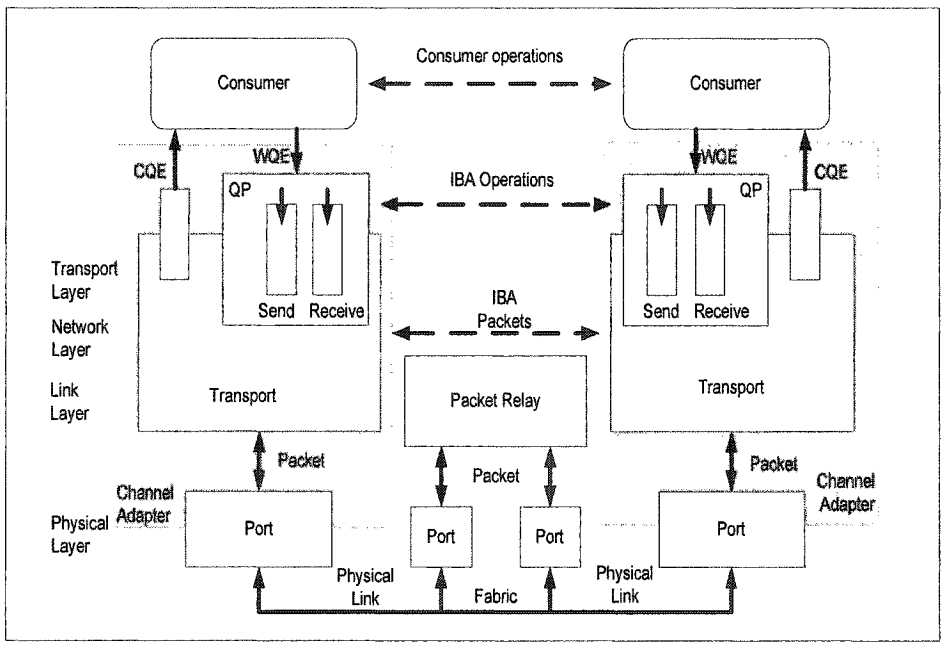

Figure 2.7 IBA Communication Stack

Figure 2.7 shows the IBA communication protocol stack. IBA *Verbs* are specified to describe the service interface between the channel adapter and the operating system. Instead of specifying an API, IBA *Verbs* describe the parameters necessary to configure, manage and operate the channel adapter. It permits the OS vendors to define appropriate APIs that take advantage of the architecture. This network system approach provides a high bandwidth interconnect that can be easily upgraded or expanded by adding the nodes to the network, rather than rebuilding the whole system.

Chapter 3 - Layered Approach to Designing System Test Interfaces

3.1 Overview

The difficulty of designing the system level test circuitry continues to rise with the increasing size and complexity of the system. An early well-planned test interface becomes essential to the needs of future complex systems on a chip (SoCs) and even distributed systems such as sensor networks [7], and home automation, including HomePNA [8]. In such systems, incorporating testing subsystems involves considerable design complexity.

In this chapter we propose a layered approach to the test interface design. Due to its flexibility and reusability, a wide class of systems can benefit from this approach. In Section 3.2, the evolution to layered test interfaces is presented by studying both advantages and disadvantages of several existing test architectures, including Boundary Scan Test techniques (JTAG), MTM-Bus, P1500, PCI test architecture and AMBA test architecture. The layered test methodology is introduced in Section 3.3. In Section 3.4, we illustrate the proposed method in detail by constructing the system test circuits for InfiniBand Architecture. We validate the proposed method by comparing it with the JTAG in Section 3.5 and conclude this chapter in Section 3.6.

3.2 Previous Test Architecture

Several standards for system test architecture have been developed. The most widely known is the IEEE 1149.5. In this section, some of the existing test architectures are presented and their advantages and disadvantages are examined.

3.2.1 IEEE 1149.5

IEEE Boundary Scan Test (JTAG) standard [9] is one of the early, well-defined system testing standards. It is defined primarily for testing interconnect between integrated circuits once they have been placed onto a printed circuit board. In this technique, all the modules under test are serially connected to form a daisy chain through the register cells, as shown in Figure 3.1.

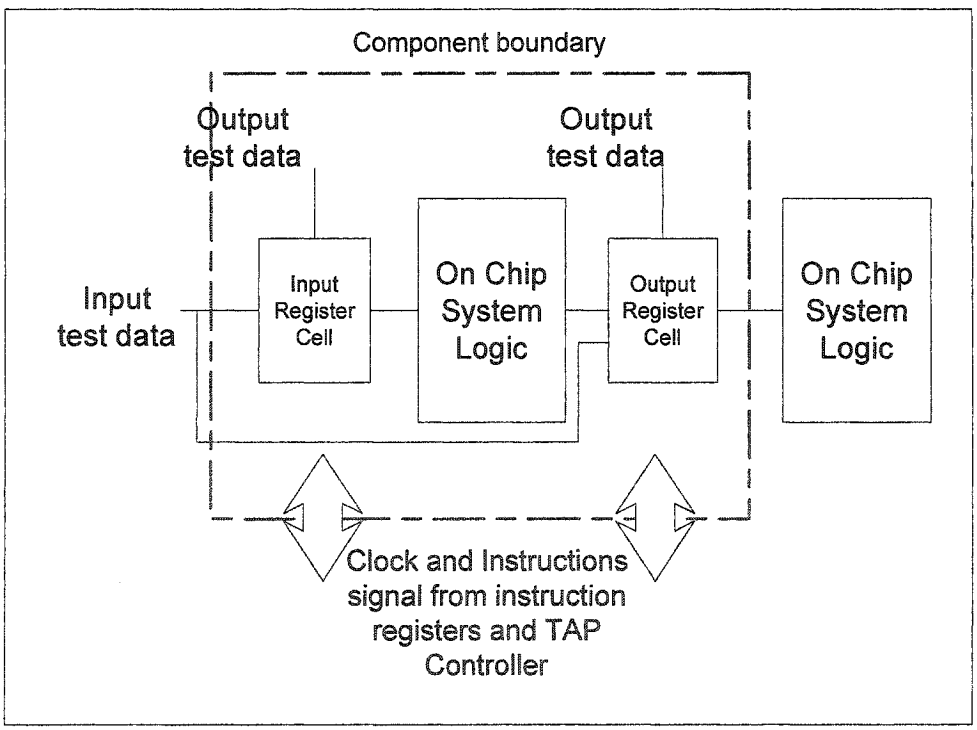

Figure 3.1 Block Diagram of a Typical Boundary Scan Test Circuitry

The register cells, which are controlled by Test Access Port (TAP) controller,

isolate the target module from the rest of the system and provide the external interface to input the test vectors and receive the test results. By controlling the following four pins: Test Clock (TCK), Test Mode Select (TMS), Test Data In (TDI), and Test Data Out (TDO), data is allowed to propagate through the device when the registered cells are inactive. During test mode, test data is shifted in or out of each module serially through the TDI and TDO of the register cell. Finally, Instruction Registers (IR) is working in conjunction with the TAP controller to perform the requested type of test.

The main advantages of JTAG are wide industry support and a good test and diagnostics at board level. It has also been used for many applications beyond the intended test application such as in-circuit programming and instrumentation. However, there are several major pitfalls while applying JTAG to the large-scale system level testing [10].

First, the test speed decreases significantly as the system size increases because of its serial data bus architecture. Second, the flexibility of the system is limited by the daisy chain configuration. Any system changes will affect the test circuitry configuration and create the hardware and software changes. Third, to avoid limitations of a single scan chain, the test setup cost becomes high due to large amount of interconnect. As demonstrated in [10], a specially designed cabinet is needed to overcome the daisy chain configuration of a highly integrated switching system. The cost of the test setup is proportional to the size and the complexity of the system. Finally, the speed of the testing is two orders of magnitude smaller than its normal operation [10].

3.2.2 PCI

In the PCI bus standard, the test circuitry was not included in the original specification. In a fairly ad-hoc manner, the JTAG interface presence on the bus has been instead added in the later specification. By employing this widely

accepted standard technique, the system test circuitry development becomes easy and increases the product compatibility. As we have discussed before, the main bottlenecks are the test speed and flexibility, which limit the capacity of the PCI system. At the moment, these disadvantages did not appear to be critical as PCI standard defined a single bus system with a limited number of components.

3.2.3 MTM-Bus

In order to overcome the system level constrains of the IEEE 1149.1, IEEE standard 1149.5: Test and Maintenance Bus (MTM-Bus) Protocol [11] was developed. MTM-Bus is a solution to the integration of logic boards with different board-level test buses into a testable and maintainable subsystem. Due to the fact that design-for-test (DFT) features are included in the modern component design, a system-level access channel is necessary to enable users to initialize and control these features in the final product.

As shown in Figure 3.2, MTM-Bus consists of two main components: MTM-Bus Master Module and MTM-Bus Slave Module. MTM-Bus Master Module is responsible for controlling and initializing the testing process through communicating with the MTM-Bus Slave modules by the MTM-Bus. Each MTM-Bus Master can control up to 255 slave modules. MTM-Bus Slave module acts as a transceiver and coordinates various board-level testing architectures, as shown in Figure 3.2. A standard MTM- Bus is used to communicate the Master module with the Slave module. It consists of five signals which are Master Data (MMD), Slave Data (MSD), Pause Request (MPR), Control (MCTL) and Clock (MCLK) as shown in Figure 3.3.

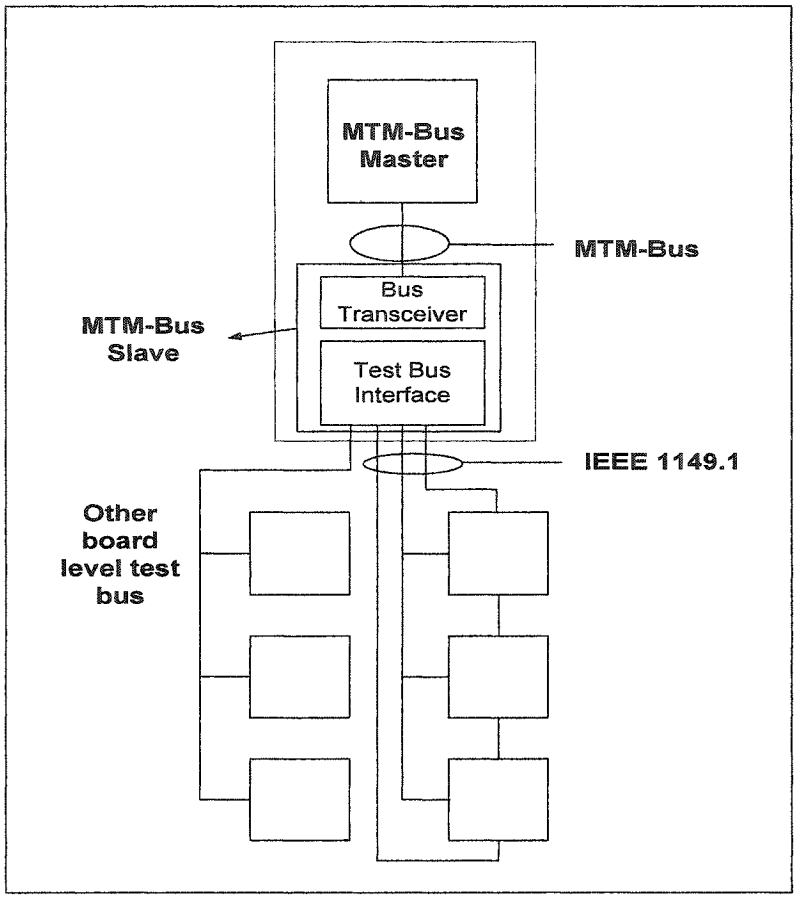

Figure 3.2 Block Diagram of the MTM-Bus Architecture

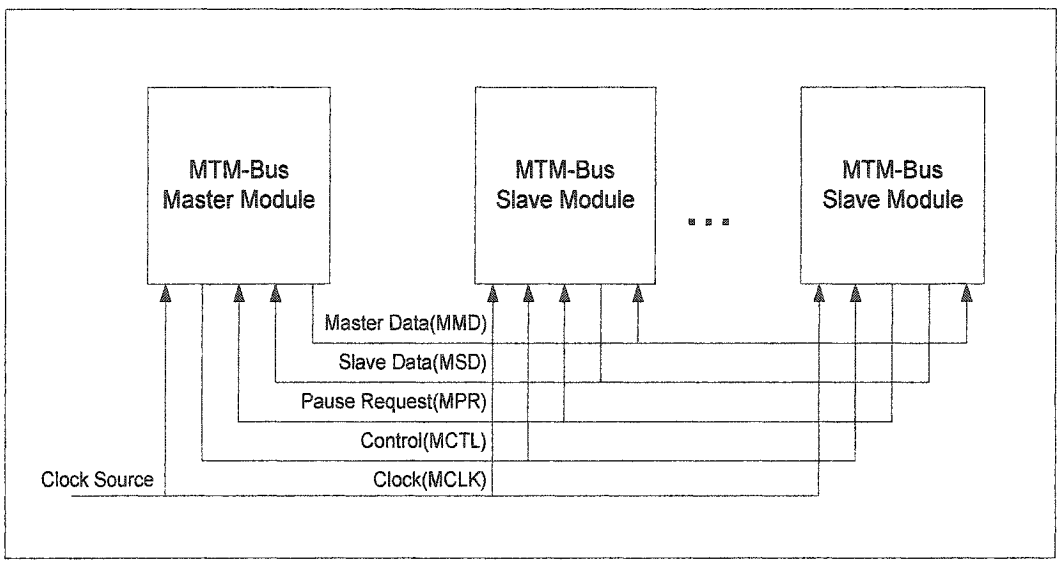

Figure 3.3 Block Diagram of the MTM-Bus Connections

By applying the MTM-Bus to different systems such as "centralized design" system or "distributed design" system, it simplifies the system integration of different board-level test architectures. In the centralized system, users can reduce the development cost by pushing all the local board-level test control functions into the MTM-Bus Slave module. This also reduces the cost of the test interface in each module. While applying it to a distributed system, the MTM-Bus interface allows each module to be fully interchangeable as long as the self-test routines and test patterns are self-contained within the module.

However this standard is not widely adapted by the industry and no commercial products are available yet. As demonstrated in [12], authors intended to use this protocol in testing VMA-based large backplane bus-based systems, with many parallel data lines on each bus. Due to the lack of commercial MTM-Bus supported product, authors are forced to use other commercial Boundary Scan Interface Devices to provide the connection link between one or several scan chains and the common test bus.

3.2.4 P1500

P1500 [13] is a new standard which targets the core testing. A typical SoC system may include several IP cores such as CPU, DSP, Memory and ADC/DAC, provided by different industrial vendors. Two major design problems are as follows. First, to decrease both the system development time and production cost, designers need to optimize the usage of IP cores and eliminate any probable compatibility issues. Secondly, system-level testing becomes more difficult due to the increased system complexity. Test Standard for Embedded Core (P1500) is developed by IEEE in order to eliminate these testing issues in the SoC systems. As described in Section 3.2.1, Boundary Scan method achieved success in the board-level testing and the scan-in mechanism is borrowed to the embedded core testing by P1500.

Figure 3.4 shows the block diagram of a SoC system equipped with P1500. P1500 provides a pseudo path that allows test data to scan in or out to different cores within the system. It defines the testing interface between the embedded core and the system chip, setup of the access and isolation mechanisms. As a result, it facilitates the test reuse and allows the P1500 compatible core to be tested in any SoC that supports P1500 without any hardware modifications.

In P1500, each core is wrapped by the standard P1500 Core test Wrapper. Test data can be imported from TAM-Source and transferred to the embedded cores through TAM-In in either serial or parallel manner. Once the test is completed, test result can be exported from TAM-Out and sent to TAM-Sink for analysis. P1500 can be divided into three parts, Core Test Wrapper, Chip Level Controller and User-Defined Access Mechanism.

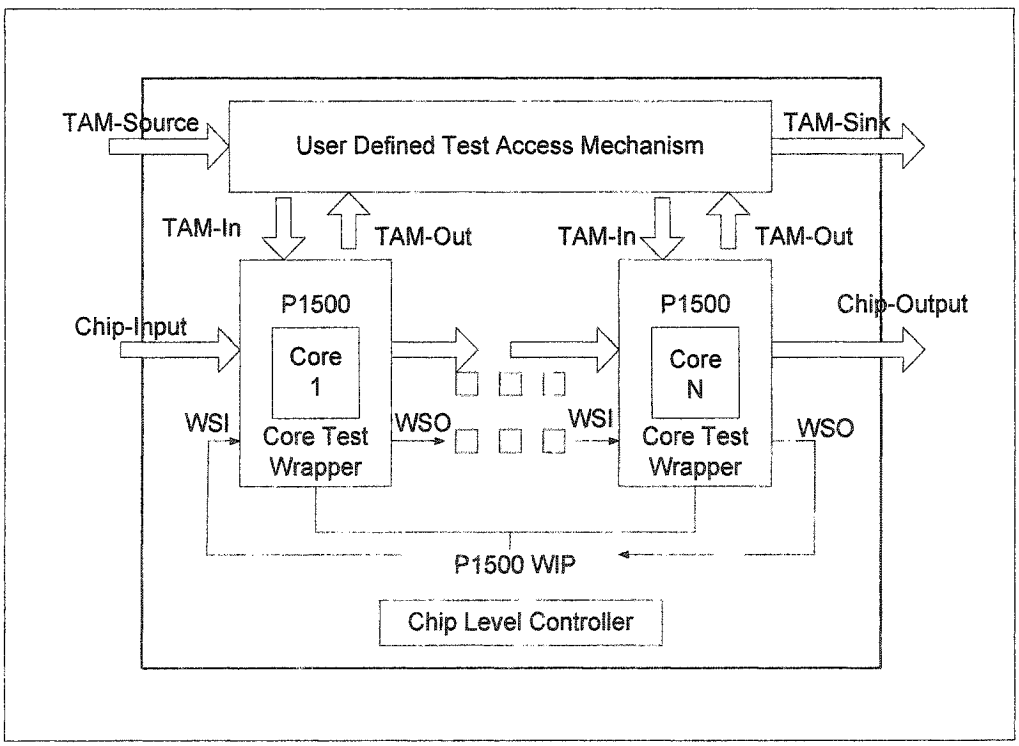

Figure 3.4 Block Diagram of P1500

- Core Test Wrapper - By applying the standard Core Test Wrapper to wrap all the input and output pins of the core, necessary test data and control signals can be easily transmitted to the cores.
- Chip Level Controller - Instead of specifying the standard Chip Level Controller, P1500 provides freedom to specify it based on the designers' system requirements. It generates the control signals for the TAM and Wrapper Instruction Register inside the standard P1500 Core Test Wrapper in order to complete the testing process.
- User-Defined Test Access Mechanism (TAM) – TAM is a user defined mechanism which transfers the test data to the target core through the TAM-In and exports the test result through TAM-Out.

P1500 specifies the scan-in mechanism. Test data are scanned into the target core through the Wrapper Serial Input (WSI) port and scanned out to the next core through the Wrapper Serial Output (WSO) port as shown in Figure 3.5.

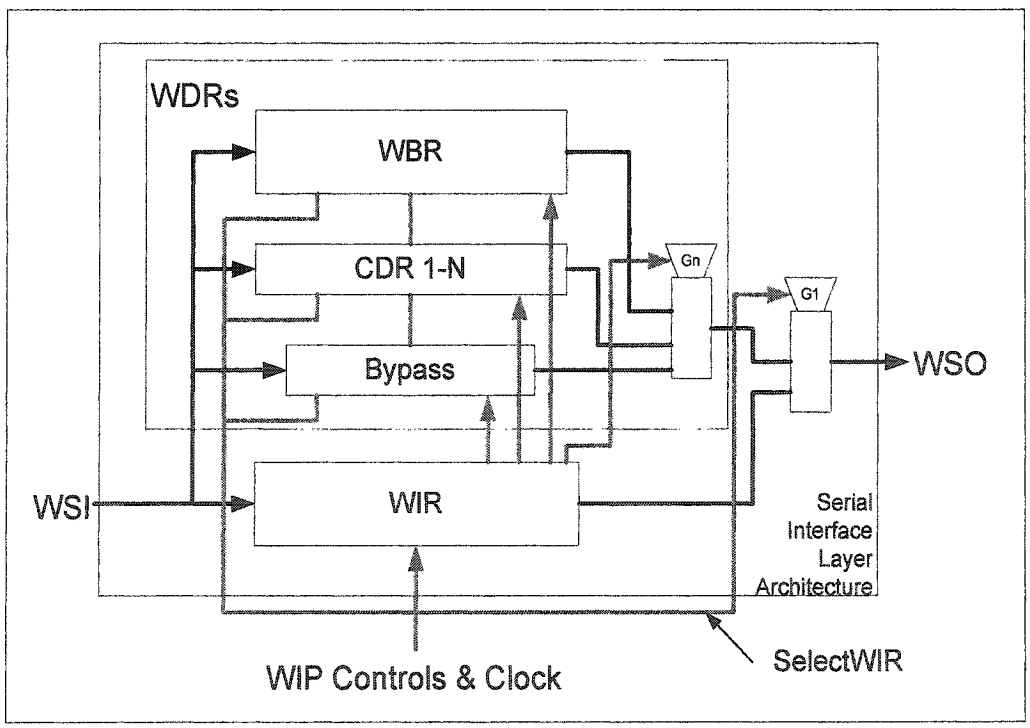

Figure 3.5 P1500 Serial Interface Layer (SIL) Architecture

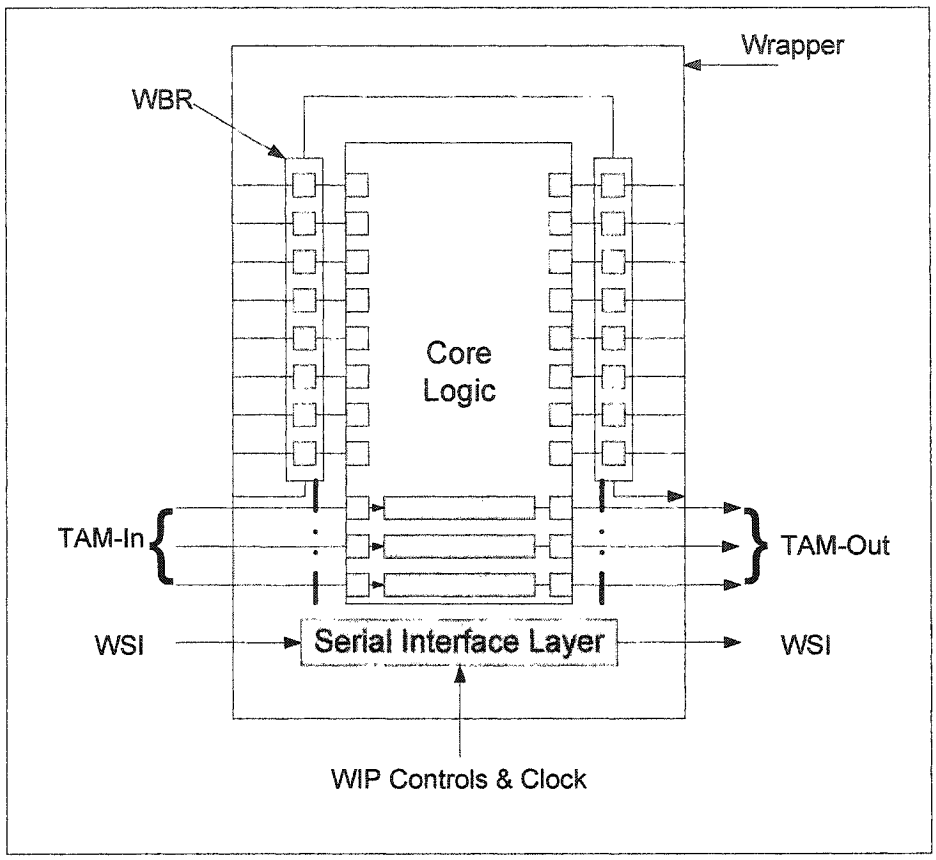

Figure 3.6 Core Test Wrapper

During the testing process, Wrapper Interface Port (WIP) Control signals are sent to the Wrapper Instruction Register (WIR). It controls and updates the content of the Wrapper Boundary Register (WBR), Core Data Register (CDR) and Bypass Register. WBR is made up of the Wrapper Boundary Cell (WBC) which is applied to each I/O pin of the core. WBC provides the same function as the register cell in the IEEE 1149.1 Boundary Scan. It allows the test data to scan into or scan out of the target core. Bypass Register is a 1-bit register that allows the test data to bypass the core through the WSO.

Since the serial test data transmission provided by the SIL architecture may not meet the system requirements, P1500 also supports the parallel data transmission as shown in Figure 3.6. On top of the original SIL architecture (WSI, WSO), parallel Test Access Mechanism (TAM-In, TAM-Out) is added to the Core Test

Wrapper. This parallel interface allows multiple I/O pins can get access to the test data at the same time to decrease the testing time. Unfortunately, the implementation of this parallel TAM is not specified in P1500 and users need to define it based on their own systems. Recently, many studies were done in this area. Several parallel TAMs have already been proposed, including CSA-BUS [39] and TestRail [40].

Although P1500 provides an easy way for chip designers to test the target cores in the SoCs by scan of the test data, topological limitation is the main disadvantage due to its daisy chain configuration. Moreover, a system-level access path that allows users to initialize and control the P1500 is needed once the SoC is integrated into the system.

3.2.5 AMBA Test Interface

The challenges of designing the SoC test interface were addressed when ARM proposed a test methodology for its AMBA, instead of adapting the standard boundary scan (IEEE standard) technique. In AMBA bus architecture, testing is performed by a centralized scheme. As shown in Figure 3.7, a specialized Test Interface Controller (TIC) module is used for the modular testing and the external interface for test control and data inputs. TIC uses a minimal three wire handshake mechanism, consisting of Test Bus Request Input (TREQA), Test Control Input (TREQB) and Test Acknowledge (TACK) to control the test operation and the External Bus Interface (EBI) that is used to provide a 32-bit, high speed, parallel vector interface.

As shown in Figure 3.8, the TREQA and TREQB signals are pipelined in testing mode. They are used to encode the mode of operation (as shown in Table 3.1.) and indicate the type of vectors that will be applied in the next cycle. The bus access is granted by the arbiter to ensure that only a single module initializes one data transfer at a time. When TREQA is inserted during the normal system operation,

TIC requests the arbiter to grant the bus access to the External Bus Interface (EBI) and the system enters into the test mode. By inserting the TACK, the TIC gives the external indication of the test bus has been granted. The TIC initializes the special burst data transfer by broadcasting the start address and indicating the bus slave for a sequential transaction through the TRANS data line. The write data is sampled and driven on the AMBA bus (HDATA) in the next cycle. If the transfer is not completed, the TACK signal is driven to low, and the external test vectors must be applied for another cycle. AMBA's test methodology demonstrates an early well-planned test approach that is essential for an efficient and flexible test circuitry.

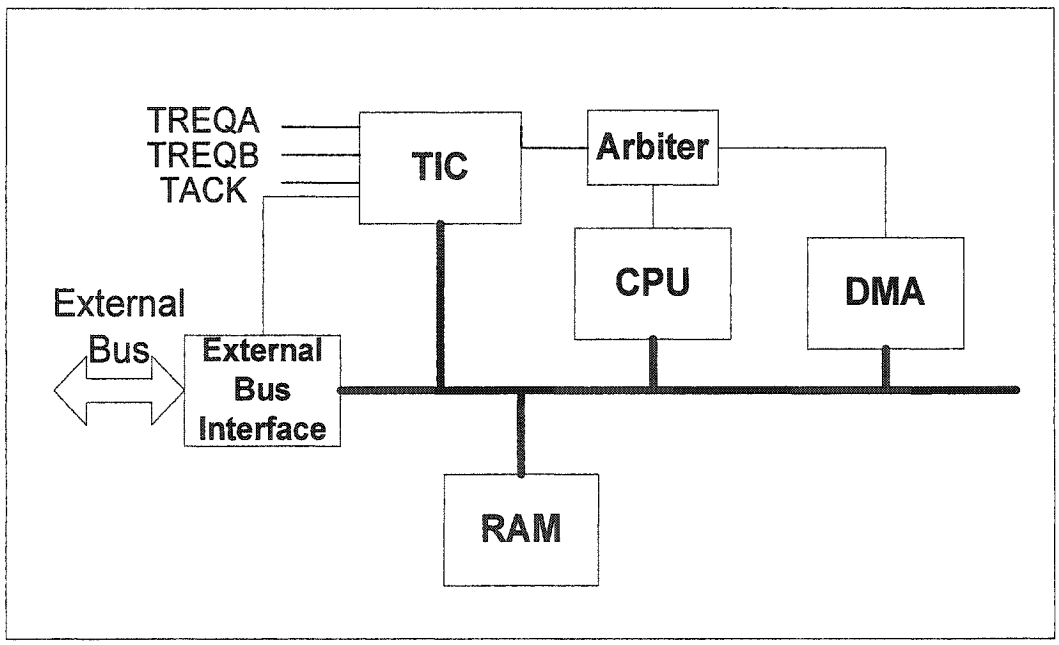

Figure 3.7 Block Diagram of the Typical AMBA-Based SoC

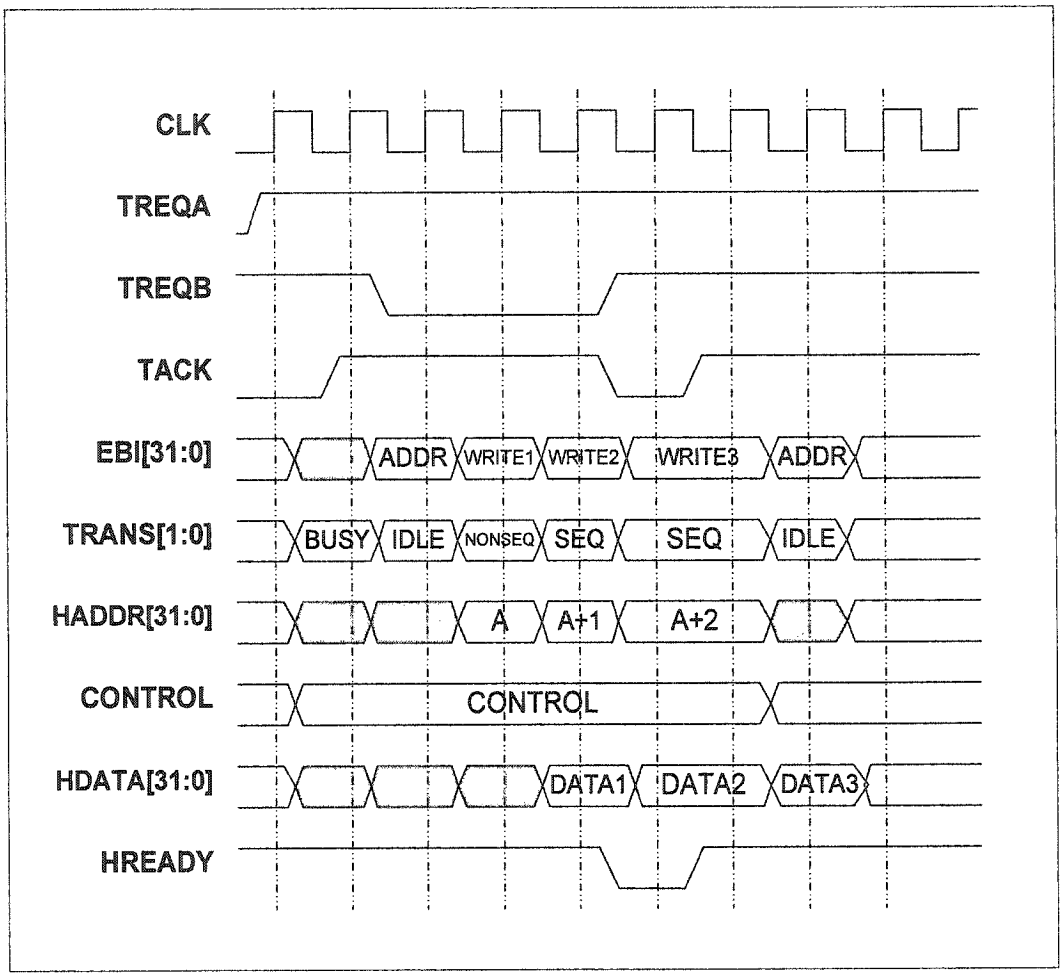

Figure 3.8 Sequence of Events of the Write Operation

TREQA	TREQB	TACK	Description
-	-	0	Current access incomplete
1	1	1	Address vector, Control vector or Turnaround vector
1	0	1	Write vector
0	1	1	Read vector
0	0	1	Exit test mode

Table 3.1 Test Control Signals during Test Mode

The speed of AMBA bus system test is improved relative to the JTAG standard by providing a high-speed parallel test interface. Further, topology limitations that are given by the fixed order of JTAG interconnections are avoided in AMBA. The centralized TAC controller can access any unit under test directly, through the common shared bus. For these reasons, system self-test of complex systems-on-chip is better addressed by AMBA standard than by employing the JTAG standard.

3.3 Layered Methodology

Both JTAG and AMBA approaches are based on one–to-one direct bus connection which leads to a high development cost as the size and the complexity of the system increase. Because of that, authors in [16] proposed a Test Access Mechanism (TAM) architecture for SoCs based on a packet switching communication network. Other studies conducted in applying the layered approach to SoC designs include [17].

We believe that applying the layered approach to design the test circuitry will be more efficient for large and complex systems. Instead of directly producing the detailed specifications of the test circuitry for a given system, which is inflexible, one is focused on specifying the requirements of each layer that maximizes the reuse and minimizes the development cost. The increased flexibility also allows this approach to be applied to a wide class of systems such as sensor network, HomePNA or SoC designs.

Each system can be partitioned into Open System Interconnect (OSI) layers, which are the physical layer, data link layer, network layer, transport layer, session layer, application layer and service layer, according to their functionality. The number of layers used is determined by the size and the complexity of the system. For example, a system may be partitioned into seven layers in HomePNA system while only three layers are needed in a simple SoC design.

The layered structure gives the framework for our test circuitry design and designers should specify the requirements of the test circuitry based on its system functionality. Once we have decided the basic structure of the test interface, each system layer is examined in detail for necessary hardware and software changes.

In this chapter, we describe the layered approach to test circuitry design within an example of InfiniBand Architecture (IBA) test architecture proposal. The suggested methodologies will target developing a test interface to control and initialize all the tests of the system. It also cooperates with the standards or with vendor developed local board level testing circuitry for in-circuit testing. Notice that it is not limited to IBA but can also be applied to any general system.

3.3.1 Case Study: InfiniBand

As described in Chapter 2, InfiniBand Architecture (IBA) is a new interconnect architecture aiming to replace the PCI bus architecture. It defines a System Area Network (SAN) that connects the nodes that can be independent processors nodes, I/O units or routers to another network through switches as shown in Figure 3.9. Each node contains one or multiple Channel Adapters that connect to the network and forms the IBA Subnet. Each IBA subnet can interconnect together through router.

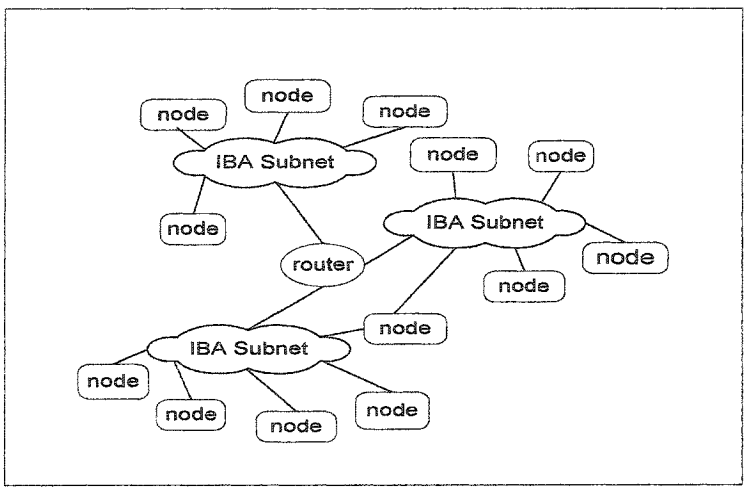

Figure 3.9 Typical IBA Network

3.3.1.1 IBA Test Architecture Considerations

It is clear that boundary scan test circuitry is not applicable for this complex network system because of its low speed and inflexibility. Traditional ad-hoc and informal design approaches are also unable to provide an efficient test circuitry and only layered approach design methodology can fit this system.

Due to the networked nature of IBA systems, it can be defined to the layered test structure as shown in Figure 3.10.

1. Physical – Physical layer is the lowest level. It specifies the transmit medium. IBA architecture defines it in terms of voltage, data encoding and packet framing. They provide the service for:
    - Establishing the physical link
    - Informing link layer of the status of the physical link
    - Monitoring the status of the physical link
    - Transmitting and receiving the control and data bytes from the link layer to the physical link when available

2. Data Link – This layer ensures the error-free transmission of the data. In IBA architecture, it is defined as the state machine and provides the error detection service by checking the packet by the length, operation code, Variant CRC (VCRC), Invariant CRC (ICRC) and virtual lane. It is also responsible for addressing, buffering and the flow control.

3. Network – The network layer provides a topology-independent view of end-to-end communication to the upper level protocol. In IBA architecture, the network layer is responsible for routing the packets to the sub-net. IBA supports a two-layer topological division. The lower layer refers to a

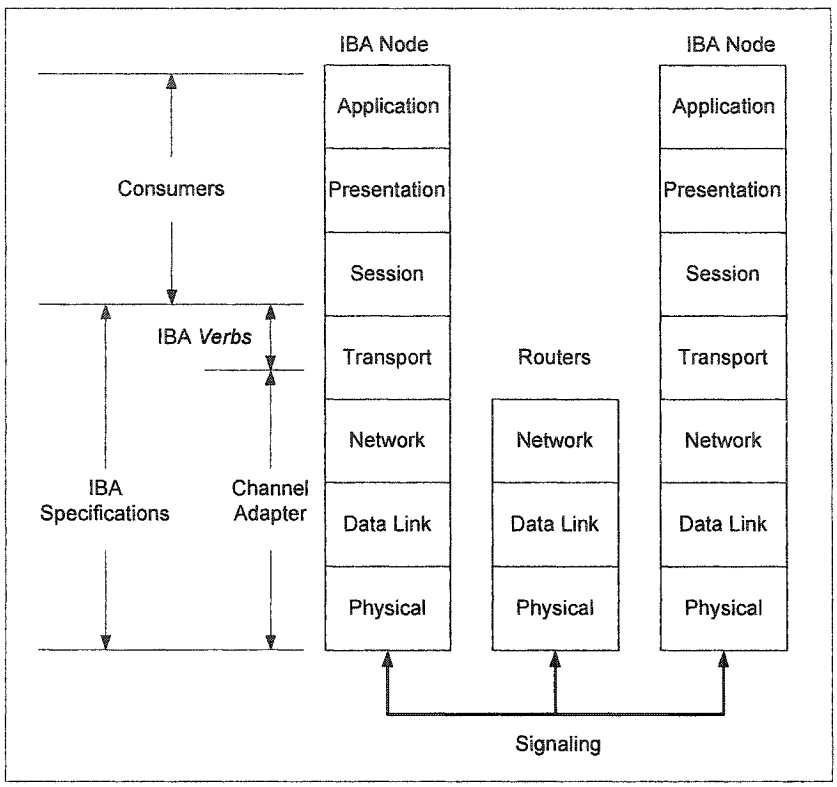

Figure 3.10 IBA Layer Model

sub-net in which packets are forwarded by the switches. The higher layer refers to the interconnection of the sub-nets where packets are forward by the routers.

4. Transport – The transport layer establishes and maintains the end-to-end connection. It is responsible to deliver the packets to the proper Queue Pair (QP) and provide the operation instructions. If the message data is larger than the maximum payload size of the packet, the transport layer partitions it into multiple packets for transmission and reassembles it in the receiver's memory. Five different transport service types are supported in IBA:
    - Reliable Connection
    - Reliable Datagram
    - Unreliable Datagram
    - Unreliable Connection

- Raw IPv6 Datagram & Raw Ethertype datagram

IBA uses the software interface that is called software transport verbs to provide the abstract definition of the IBA hardware.

5. Session – Session layer synchronizes messages between peers and ties all different data streams into a single application.
6. Presentation – Presentation Layer handles the format of the data exchange between peers.
7. Application – Application layer hides the details of the lower layers and focuses on the abstraction of the overall system.

The top three layers are beyond the scope of the IBA specification and are omitted since they depend on the software applications which eventually are based on the software transport verbs of the IBA.

Once the system is partitioned into several layers, designer should specify the requirements of its design. In IBA, test circuitry should obey the following requirements:

- Flexibility – The high upgradeability of the IBA architecture should be preserved and no modification of the test circuitry system should be needed as the system is expanded or collapsed by adding or removing the function nodes.
- Reusability – The reuse of the existing design can dramatically cut the development cost in the complex network system.
- Test Speed – With the provided high bandwidth interconnect system, it is preferable to have a high speed testing circuitry.

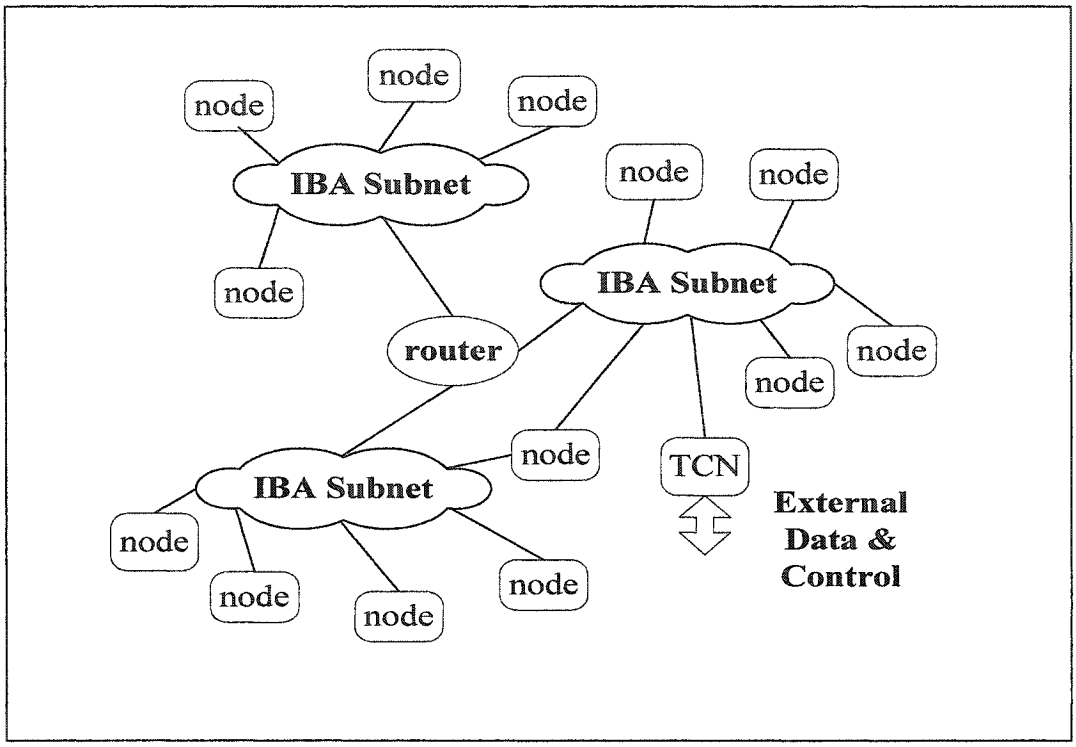

Figure 3.11 IBA Network with Proposed Test Circuitry Design

We decided to use a specialized functional node, Test Control Node (TCN) to initialize tests and to provide the external interface for data input and test control. As Figure 3.11 shows, the TCN is connected to the IBA network through the standard IBA switches and routers. One can initialize and control the testing of the target node through the external interface provided by the TCN. In testing mode, TCN and target nodes should be isolated and become transparent to the rest of the system. If a failure node is detected, based on the failure nature, user can isolate the end node through the provided standard IBA subnet management or replace the failure node.

Due to this transparency requirement, there should be no changes in the physical layer, the data link layer and the network layer. By doing this, TCN can reuse the standard design of these three layers and no modification is needed for both switches and routers since the standard IBA communication protocol will be reused. For the transport layer, specific operation code should be added to handle starting the test operation, sending test vectors and receiving test result. Within

the five different transport service types, reliable connection should be used to minimize the data loss through the network.

At this stage, the framework of the IBA system test circuitry is formed and designers can detail each layer of the TCN based on the given topology of IBA. Consider the difference between two IBA systems is the transmission media, where one uses cables while the other uses fiber optics for interconnect. Instead of designing two TCNs for each system, the layered design approach allows reusing the TCN design for both systems as long as the physical layer is modified appropriately. This provides a high flexibility and decreases the designing cost through the reuse.

3.3.1.2 The TCN Design

As shown in Figure 3.12, TCN is divided into three main components, Channel adapter, Memory and Test Controller.

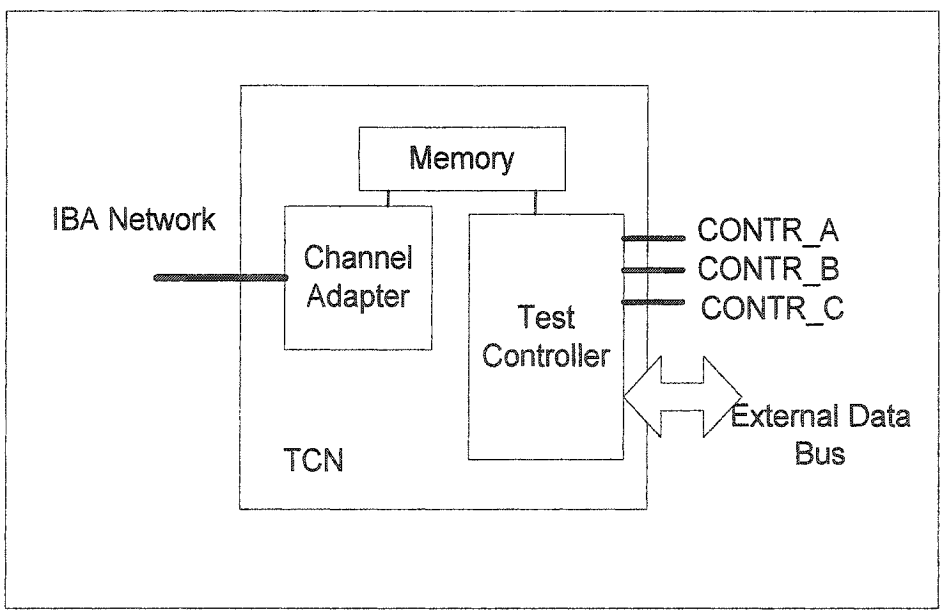

Figure 3.12 Block Diagram of the Test Control Node (TCN)

3.3.1.2.1 Channel Adapter

Channel Adapter is a standard IBA component that provides the communication channel between the TCN and IBA network. It provides the service of the physical layer, data link layer and network layer, such as performs the error checking to the incoming packets, by introducing the appropriate connection interface to the IBA.

As shown in Figure 3.13, the standard IBA channel adapter works as the Direct Memory Administrator (DMA) that receives the data and commands from the local memory and interfaces with the IBA network through the ports. Each port has its own set of transmit and receive buffers that are channeled through the Virtual Lane (VL). Memory Translation & Protection (MTP) mechanism is used to translate virtual addresses to physical addresses and validate access rights. When communication is established, both ports from the master and slave nodes are connected and isolated from the rest of the network.

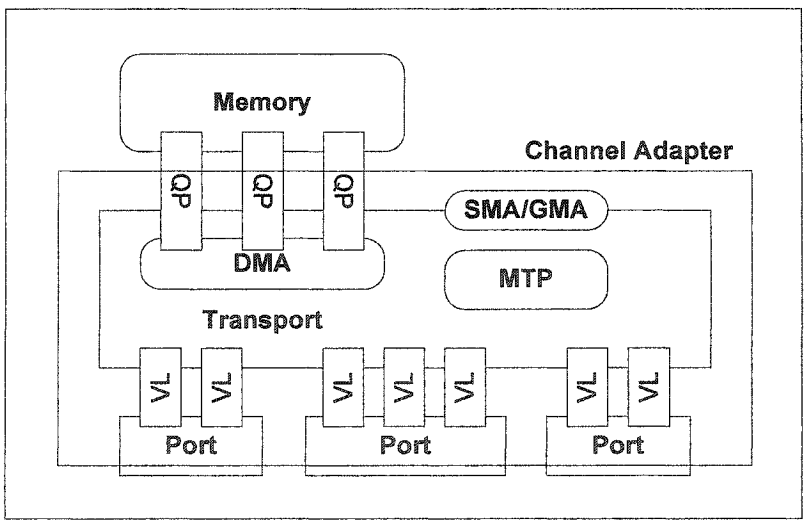

Figure 3.13 Block Diagram of the Standard IBA Channel Adapter

3.3.1.2.2 Test Controller

Test controller is a finite state machine as shown in Figure 3.14. It controls and

monitors testing based on the control signals and test vectors that are received from the external interface. On one hand, it analyzes the input packets and exports the test result to the external data bus. On the other hand, it receives all the input commands from the user interface and controls the test processes. Table 3.2 shows the control signals of the TCN during test mode.

3.3.1.2.3 Memory

Based on the IBA specification, memory buffers all the data and instructions that are sent between Test Controller and Channel Adapter. In IBA architecture, all the instructions are sent as send/receive queue pairs which are stored in the memory. The size of the memory will be determined by the number of the target nodes and the number of ports in the channel adapter.

Control Pin			STATE	Description
A	B	C		
0	0	0	IDLE	TCN is in the IDLE state and no test operation is performed
0	0	1	START	Test operation starts and transits to the ADDRESS state in the next clock cycle
0	1	1	WRITE	Write test data
1	0	1	READ	Read test result

Table 3.2 TCN Control Signals during Test Mode

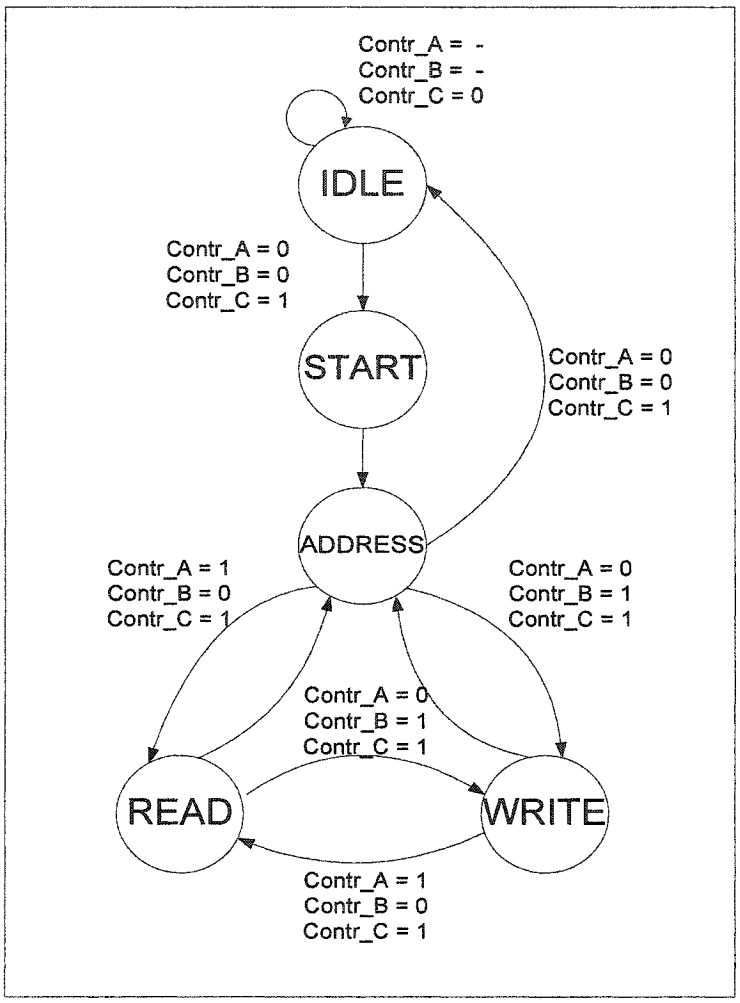

Figure 3.14 State-Machine of Test Controller

3.3.1.2.4 Single Node Testing Sequences

Three input signal pins, Control Pin A (CONTR_A), Control Pin B (CONTR_B) and Control Pin C (CONTR_C) which are driven by the external tester are used to encode the mode of operation. As shown in Figure 3.15, by inserting the CONTR_C, Controller places a Test Start Work Queue Entry on the work queue. In the next cycle, target address is imported to the Controller through the External Data Bus and buffered in the memory. Channel Adapter detects the test start request, validates the target address, formats the test request instruction packet and sends it the target node. Once the target node received the packet, the built-in-self test activates and the functional test starts. By controlling the CONTR_A and

CONTR_B signal pins, data can be written to or read from the target node following the same transaction flow. Since IBA architecture specifies various management infrastructures within the standard channel adapters, a reliable communication service is guaranteed.

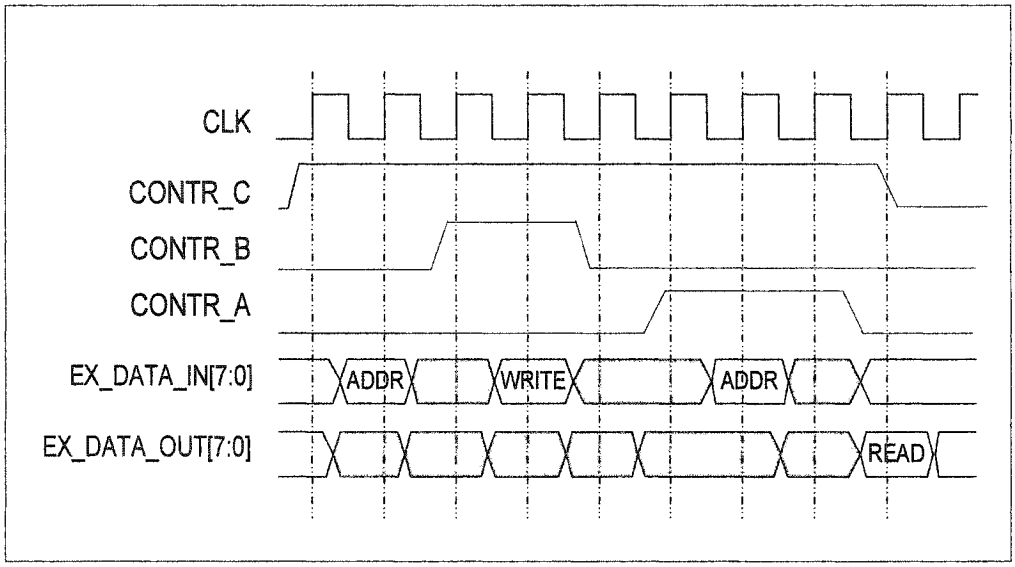

Figure 3.15 Timing Diagram of the Write and Read Test Operation

3.3.1.2.5 Multiple Node Testing

Although a single TCN is sufficient to handle the single node testing operation within the system, multiple concurrent node testing may require in some system. Fortunately, designers can add multiple TCNs to form the TCN network while no hardware modification is necessary. Each TCN has the ability to test every single end node within the system concurrently. During testing, each target node is isolated from the rest of the system. All the tests run locally, depending only on the externally provided test data. The Subnet Management Agent (SMA) and General Management Agent (GMA), located on the Channel Adapter of the TCN, handle the test scheduling and isolate the target nodes within the system. The testing sequences of each TCN and target node pair will be the same as a single node testing while multiple pairs can be performed in parallel. Notice that ideally if all the test processes are controlled locally and no external test vectors are

needed, the test time of multiple node testing is the same as the single node testing case. However, test time will increase dramatically in multiple node testing scenarios as external control and test vectors are needed.

3.4 Experimental Results

To validate the proposed approach based on the layered system design methodology, we designed and simulated the proposed IBA test circuits in VHDL and ported it to the FPGAs.

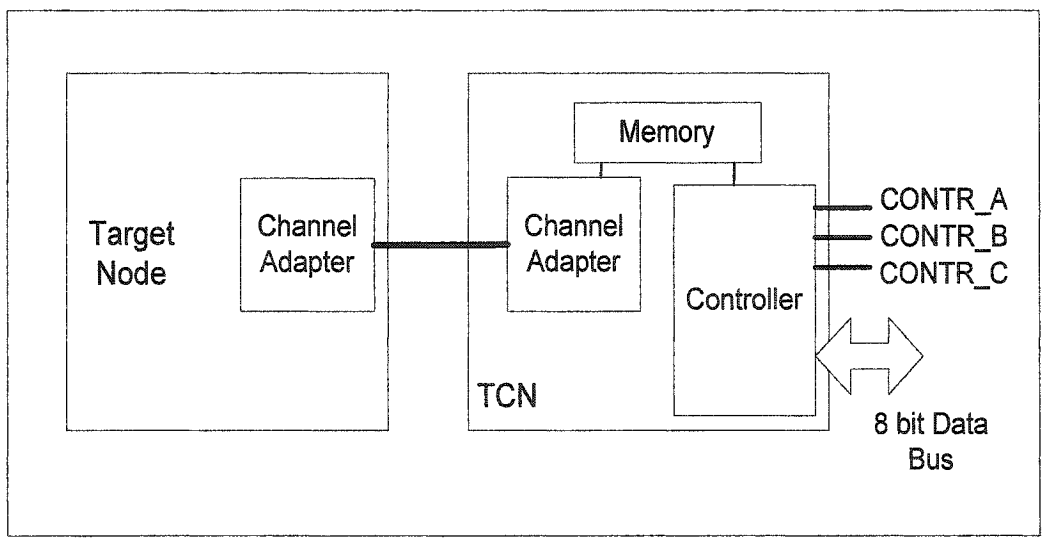

Figure 3.16 Block Diagram of Our TCN Design with Single Target Node

Figure 3.16 shows the block diagram of our TCN design with the target node. To simplify the design, we used an eight bit external data bus with 2K memory in our TCN design. Both the channel adapters of the TCN and the target node are designed to handle the packet formation. As discussed above, the controller is a state machine that controls the testing process, inputs the test vectors and exports the test result. The design is ported to two different Altera Corp. programmable logic device architectures, Flex 10KE device (EPF10K30ETC144-1) with 1728 logic elements and MAX 9000 (EPM9320LC84-15) device with 320 macrocells. The area and speed results are shown in the Table 3.3.

	Flex 10KE	MAX9000
% of Logic Cells Utilized	30	78
Max Speed	90MHz	90.9MHZ

Table 3.3 Performance Results of TCN with Different Programmable Logic Architectures

The size of our TCN design is mainly dictated by the Channel Adapters and the choice of the Memory core. In our design, the gate count of the TCN is around 10K and the size of the Controller is only 20% of the whole design. The size of the TCN may increase by expanding the memory size and the functionality of the channel adapter; however, it will keep constant for various system sizes. Notice that the maximum test speed is only 90MHz. This is due to the choice of FPGA devices used. The speed can substantially increase by using FPGA with better speed grade or using in ASICs.

3.4.1 IBA Test Interface Based on MTM-Bus Architecture

For comparison, the IBA test interface model of a single end node based on MTM-Bus is simulated and ported to Altera's Flex 10KE (EPF10K30ETC144-1) and MAX9000 (EPM9320LC84-15) devices and the results are shown in Table 2.

Figure 3.17 shows the details of applying the MTM bus protocol to the IBA. MTM-bus Master module controls and monitors the entire test system through the dedicated MTM-Bus by communicating with the MTM-Bus Slave interfaces of each end node. Notice that by using eight bit address size, a maximum of 255 modules (end nodes) can be controlled by one MTM-Bus Master Module.

	Flex 10KE	MAX9000
% of Logic Cells Utilized	9	13
Max Speed	81MHz	29MHZ

Table 3.4 Performance Results of IBA Test interface Based on MTM-Bus with a Single End Node

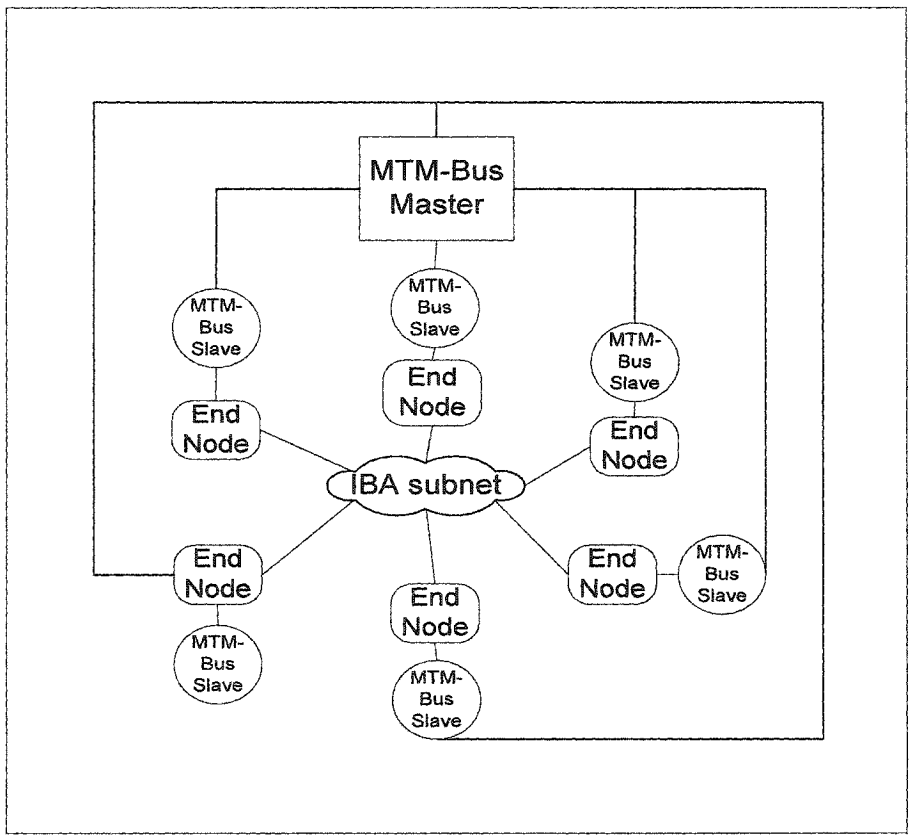

Figure 3.17 Block Diagram of the IBA test Interface Based on the MTM-Bus Protocol

3.4.2 Comparison

Reusability is the main disadvantages of MTM-Bus when compared with our layered approach methodology. As shown in Figure 3.17, the size of the test interface based on MTM-Bus will grow proportionally to the number of end nodes.

In order to visualize this problem, the logic cells utilizations of both test interfaces are estimated and plotted against the IBA network with different number of end nodes. As shown in Figure 3.18, TCN enables a low gate count design that is not affected by the system size, while additional hardware (MTM-Bus Slave Interface) and wires will be needed to serially connect each node if MTM-Bus is applied. This degrades the expandability and flexibility of the system. As a result, the cost of our test scheme only depends on the physical layer of the system. Although in some systems, they may consist costly components such as wireless transceivers or optical front-ends, our approach will still cost less when compared to the MTM-Bus since the test cost is kept constant and not related to the system size.

Due to the fact that the physical layer, data link layer and network layer are unchanged in our test circuitry, this approach maximizes the reuse of the existing hardware. Moreover, multiple nodes can also be tested concurrently by adding TCNs to form the TCN network, which is not achievable in the MTM-Bus architecture.

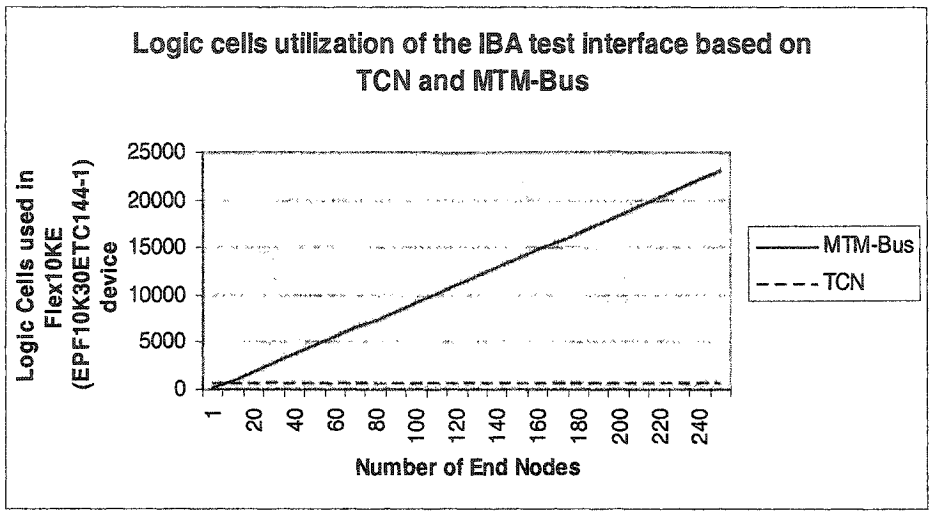

Figure 3.18 Logic Cells Utilizations of the TCN and MTM-Bus Interface with Different System Size

3.5 Conclusions

The existing boundary-scan test circuitry fails to meet the challenges posed by the complex networked systems. By comparing the PCI bus architecture and AMBA, one can conclude that ad-hoc or informal design approach ends up with inefficient and slow testing circuitry. In this Chapter, we have presented a disciplined layered design methodology for the complex system test circuitry and demonstrated its benefits on the InfiniBand architecture. This approach is not limited to the IBA network system, but is equally applicable to the wider class of systems including Systems on Chip (SoC). The proposed designed methodology is also presented in the VLSI Test Symposium 2002 [36].

Chapter 4 - Extended Usage of the Test Interface Based on Layered Design Approach

4.1 Overview

As mentioned in Chapter 3, the main concept of layered approach is to partition the system into different layers based on their functionality. Instead of building the test interface directly from scratch, each functional layer is modified so as to provide the testing access. By doing so, the developed test interface is embedded in the system, which can maximize the reuse and flexibility. Because of this reuse, the developed test interface can also be viewed as the access channel to the various components within the system. This provides the possibility of using the developed test interface for other operations such as online programming, debugging and software upgrade. In terms of flexibility, the highly reused system-level test interface also ensures the compatibility to the distributed systems such as sensor networks and eases the product development.

In this chapter, we explore the benefits of our approach in several systems. In Section 4.2, the extended usage such as online programming, debugging and software upgrade, of our proposed test interface in the general system is discussed. Our IBA TCN is revisited in Section 4.3 and examined the necessary modifications in order to provide the system access features. In Section 4.4, we apply our design methodology to the distributed systems and demonstrate it in the sensor networks. Finally, this chapter is concluded in Section 4.5.

4.2 Extended Usage of the Test Interface

Rapidly developed technology not only forces industry to push their products to the market sooner but also shortens the product life. As a result, industry vendors are always concerned about decreasing the length of the development phase and adding the upgradeability features to their designs in order to lengthen the product life. In order to support this upgradeability feature, the system must provide the access interface for the designer to communicate with the target devices. As our proposed test interface is based on the layered design approach, only small modifications are needed to provide the access channel for the various components within the system. In order to show the simplicity of adding the system access features in the test interface based on layered design methodologies, our proposed IBA test interface is used for demonstration.

4.2.1 IEEE 1532

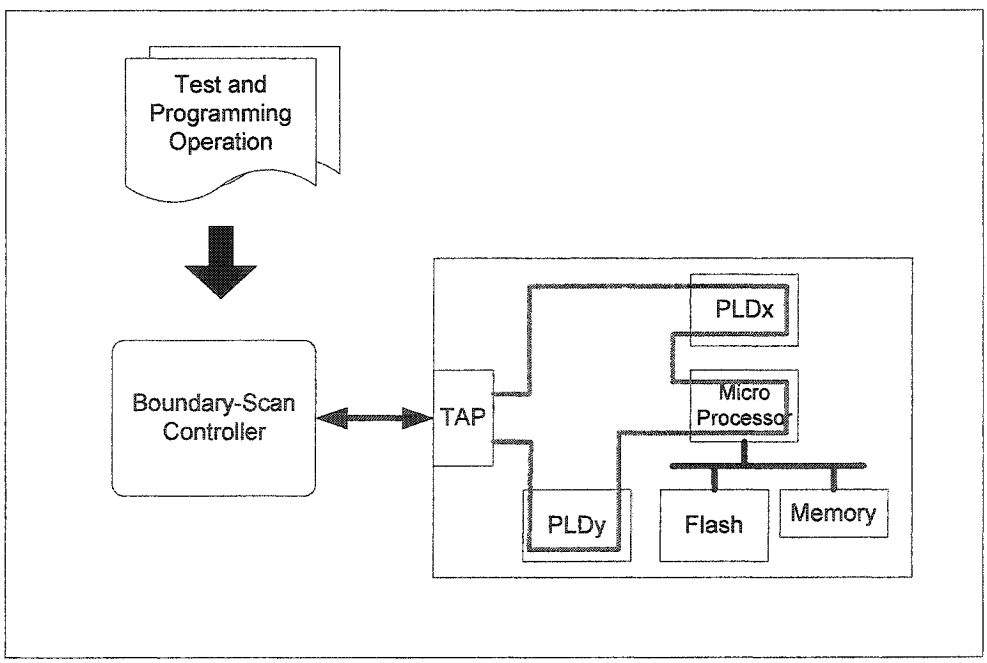

Figure 4.1 Block Diagram of IEEE 1532

In JTAG, the programming features are accomplished by establishing a new set of programming instruction and associated data registers. IEEE standard 1532 [52] defines the additional instructions, the data file format and Boundary-Scan Description Language (BSDL) extensions that are used for the programming algorithms. As shown in Chapter 3, section 3.2, device that is compliant with IEEE standard 1149.1 has a 'test' mode and a 'system' mode. During test mode, the device is isolated from the system and test data is scanned into or out of the device through the register cells. IEEE standard 1532 (as shown in Figure 4.1) extends the JTAG functions by introducing four additional system modal states, programmed, Operational, In-System Configurations (ISC) Accessed and ISC Completed, that handles the in-system configurations. Three control signals, ISC_Done, ISC_Completed and ISC_Disable_Completing are used to control the mode transition. Unprogrammed devices can be isolated from the system by setting it into the Unprogrammed mode. By activating the ISC_Enable signal, the target device enters the ISC_Accessed mode and performs the programming operation. Once the programming operation is done, it transits to the ISC_Completed mode by deactivating the ISC_Enable and activating the ISC_Disable_Completing signals. Target device can resume its normal system operations by entering the Operational mode once the ISC_Done signal is set.

4.2.2 Modifications of the IBA Test Interface

The main tasks in the programming mechanisms include writing the programming data to the target devices and reading the programming results. Based on the IBA layer model that we defined in Chapter 3, the lowest three layers, the physical, data link and the network layers are only responsible for the IBA communication mechanisms. They do not need any modification and can be reused. The only part that needs to be modified in order to handle the software upgrading and programming mechanisms is the transport layer.

Control Pin				STATE	Description
A	B	C	D		
0	0	0	0	IDLE	TCN is idle
0	0	1	0	Test START	Test operation starts and wait for the target address in the next clock cycle
0	1	1	0	Test WRITE	Write test data to the target device in the next clock cycle
1	0	1	0	Test READ	Read test result to the target device in the next clock cycle
0	0	1	1	Program START	Programming operation starts and wait for the target address in the next clock cycle
0	1	1	1	Program WRITE	Write programming data to the target device in the next clock cycle
1	0	1	1	Program READ	Read programming result to the target device in the next clock cycle
-	-	0	0	Exit Program Mode	Exit the program mode and TCN is idle in the next clock cycle

Table 4.1 Control Signals during Test Mode and Program Mode

For our proposed TCN, Channel Adapter (CA) and memory should not be changed while Controller needs to be slightly adjusted to fulfill this requirement. Four additional states, Program START, Program ADDRESS, Program WRITE and Program READ are added in the Controller to handle the programming functions as shown in Figure 4.2.

In order to enter and control these programming states, a special control signal pin, CONTR_D, is added for the programming function. The control signals for both

the test mode and program mode are shown in Table 4.1.

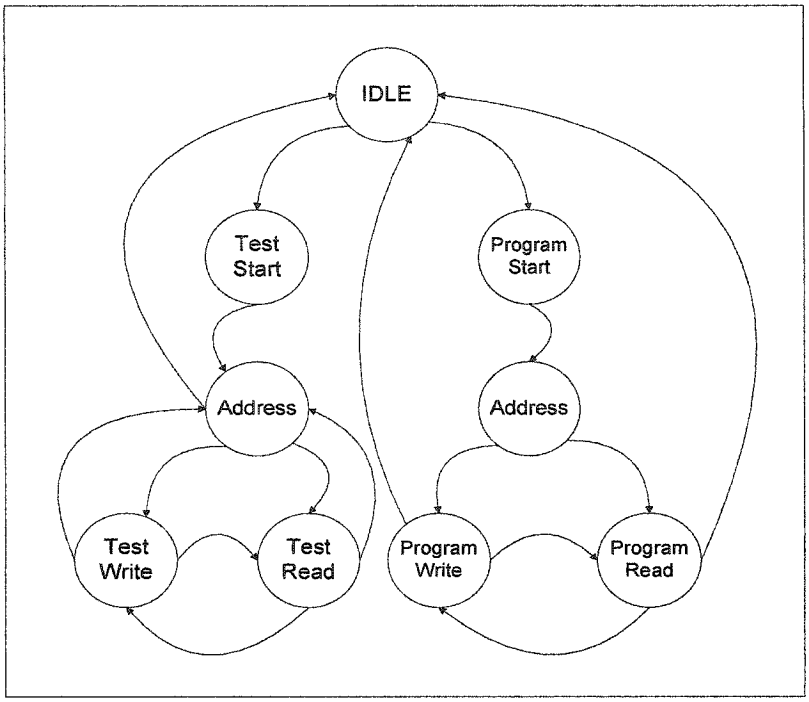

Figure 4.2 State Machine of the Modified TCN's Controller

4.2.3 Experimental Results

The design is ported to two different Altera Corp. programmable logic device architectures, Flex 10KE device (EPF10K30ETC144-1) with 1728 logic elements, -1 speed grade and MAX 9000 (EPM9320LC84-15) device with 320 macrocells, -15 speed grade. The area and speed results are shown in the Table 4.2.

	Flex 10KE	MAX9000
% of Logic Cells Utilized	40	86
Max Speed	90MHz	90.9MHZ

Table 4.2 Performance Results of modified TCN with Different Programmable Logic Architectures

By using the Flex10KE device, the gate count of our modified TCN is around 12K. Notice that there is a 20% increased in size when comparing with the original TCN (which the size is 10K) as shown in Table 3.3. This overhead is caused by the additional programming function in the TCN's controller. Since the state machine of the program mode is similar to the one in the test mode, the maximum operating speed is remained the same.

From this example, our proposed test interface shows the flexibility by adding the programming functions while requiring minimal modifications to the system. All the changes are made locally in the TCN node. Since the IBA lowest three layers are designed based on the standard specifications, the modified TCN is backward compatible with all the standard IBA devices.

4.3 Application to the Distributed Measurement System

Traditionally, centralized design approach is used in the control and measurement systems. In these systems, a central controller is used to directly control the peripheral devices and perform the data management and coordination operations. Communication between the peripherals and the central controller has been usually established through the serial protocol such as RS-232 in the past. Recently, various network technologies such as LonTalk™, USB and DeviceNet™ are being used to replace the old serial protocol. With the contributions of the IEEE1451.1 [18] and IEEE1451.2 [19] standards, more and more measurement systems are shifted from the centralized design to the distributed system design.

4.3.1 The Distributed System

Figure 4.3 shows a typical distributed system which consists of a collection of

computers, file servers, printer servers and other devices. In the distributed system, they are all referred to as nodes. Nodes are interconnected by the networking interconnect architecture such as Ethernet. As shown in Figure 4.3, data and printer can be shared within the system through the distributed system. Additional freedom and flexibility are achieved by dividing the applications among the connected nodes. The overall system performance and the cost-effectiveness are enhanced since multiple nodes are working on the same process. Due to its highly flexible and robust characteristics, it fits the needs for the measurement and control systems. Many studies [20] [21] are done on effectively applying the distributed design philosophy to these systems.

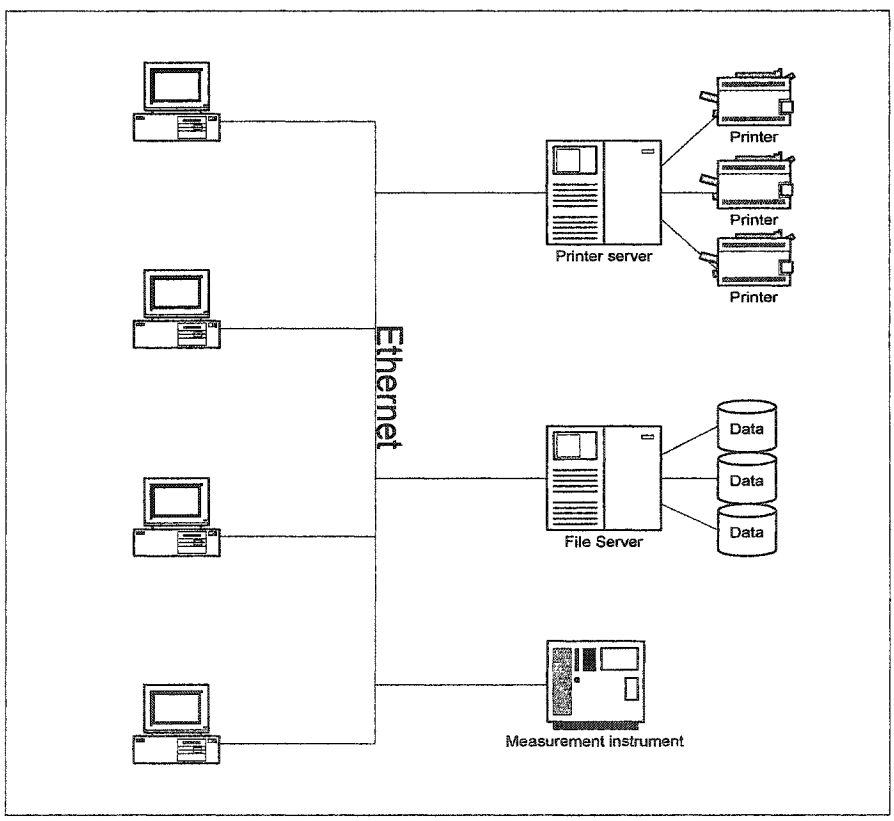

Figure 4.3 Typical Distributed System

In the centrally controlled system, many tasks such as events ordering and state management are exercised locally in the central controller. In the distributed system, these tasks need to be completed through the communication between the

nodes. Unlike with the central controller in the centralized system, applications are partitioned into small portions and are completed by each node within the system. Each attached instrument or device needs to have independent application capabilities so that they are capable of implementing a small portion of the system application. In addition, coordinating the time between individual nodes is essential for the parallel execution. Real time control, data management and robust communication also need to be carefully designed. One of the commonly used distributed measurement system is the sensor networks.

4.3.2 Sensor Networks

A general sensor network is made up of three main components: Sensors nodes, Task Manager Node (user) and Interconnect backbone, as shown in Figure 4.4. Task Manager Node provides the visual display and the user interface to the system. It allows user controlling the sensor nodes through different kinds of wireless transmission media such as RF radio, Bluetooth or infrared. Different sensors are used to collect the target environmental data that is transferred to the user through the Internet for data storage, display and analysis. Besides collecting the data, sensors may need to perform some computation on the measured data and direct communication between each sensor is required. Because of its flexibility and low setup cost, they can be used for various home, health or military applications.

Sensor networks should meet all the real-time measurement requirements. On the other hand, they should provide a robust system. General requirements for the sensor networks are the following.

1. Scalability – Due to the fact that a large number of the sensor nodes are used for measurement, network with 10,000 or even 100,000 sensor nodes are common in the sensor networks. Although the sensors are stationary in most cases, mobile sensors may also be used in the military or

environmental applications. The scalability of the system becomes a major concern.

2. Low Power Consumption - Most of the time, sensor networks are used in the extreme environments that exclude human beings for the data measurement and collection. The servicing of the nodes is not possible which makes the lifetime of the sensor nodes heavily reliant on their battery life. As a result, minimization of the power consumption becomes critical to achieve a robust system.

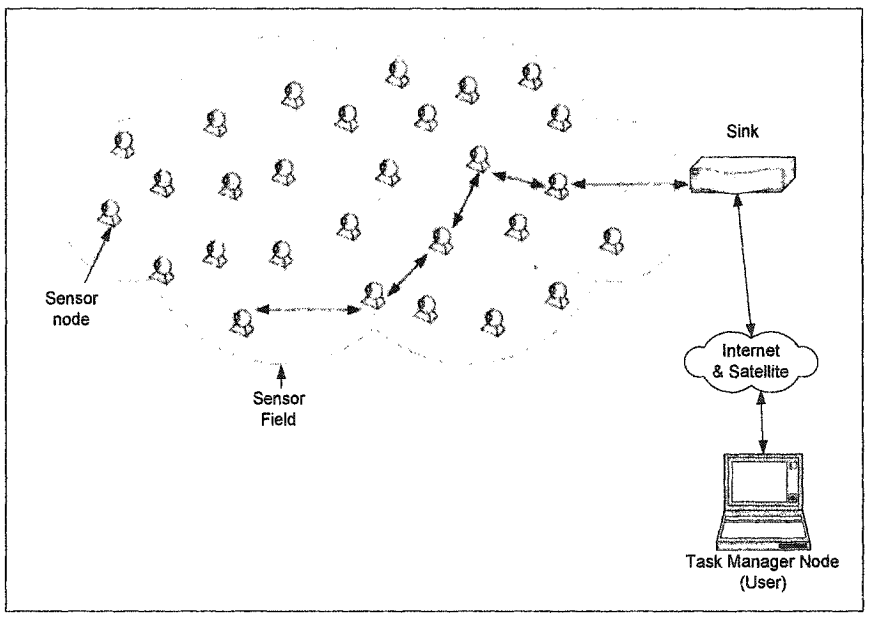

Figure 4.4 A General Sensor Network

3. Self-Organization Ability – As mentioned before, sensor networks are large in size and work in the hostile environment. It causes individual sensors to fail or join the network randomly. Manual setup can not be performed and the self-organization ability is essential.
4. Querying Ability – Due to the network size, user may want to collect the data or information in a particular region or from certain nodes. The amount of the request data may be too large to be transmitted through the network. Certain nodes may need to collect all the data and create a

summary. A query function is needed in order to bring the users to these nodes and obtain the desired measurement data.

4.3.3 System-Level Testing Considerations for Sensor Networks

Since the sensor networks are based on the distributed design approach, each node should be equipped with Design-For-Test (DFT) features. It should have enough processing power to handle testing and maintenance functions. Once the sensor node is connected to the network, an alternate path is needed in the system-level to access these DFT features.

Although MTM-Bus provides the accessibility of the board-level DFT features by establishing a test bus, it fails to meet the general requirements of the sensor networks. As discussed in Chapter 2, MTM-Bus requires each node equipped by a slave module interface controlled by a MTM-Bus master module. This is obviously not suitable for the sensor networks since a MTM-Bus master module can only handle up to maximum 255 slave modules. There are usually ten thousands of sensor nodes within the sensor networks. Setting it up with MTM-Bus would require extensive wiring and coordination of multiple MTM-Bus master modules which complicates the whole testing processes. Moreover, the MTM-Bus protocol is not optimized to lower the power consumption and makes it impossible for adapting the MTM-Bus architecture in the general sensor networks.

In order to fulfill all the requirements for the sensor networks, one of the best solutions is to reuse its only communication channel and data transfer mechanism to provide the system-level test access path. This can only be achieved by applying the layered design methodology. If we partition a typical sensor network based on its functionality, the layered model in Figure 4.5 is obtained. It contains five functional layers.

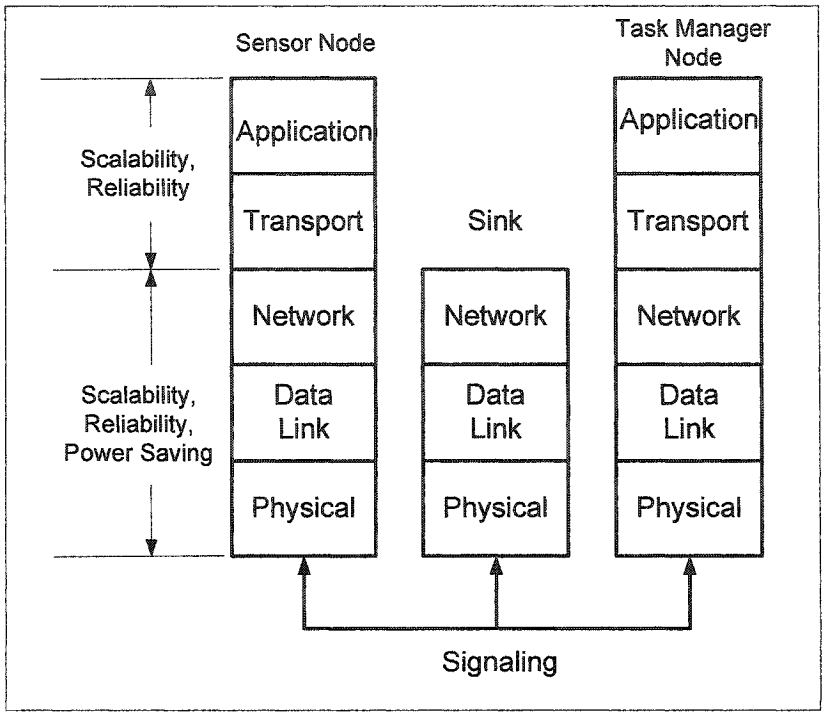

Figure 4.5 General Sensor Networks Layer Model

- Physical Layer – Based on the measurement purposes, transmission media can be any wireless connections such as radio and infrared for mobility or stationary measurements. Low power and robust connections are the main requirement.
- Data Link Layer – In wireless connection, a Media Access Control (MAC) mechanism is required to handle the distribution of the network resources. Due to the low power consumption and self-organization ability requirements, conventional MAC protocol cannot be adapted and many new sensor networks MAC protocols [44] [45] [46] are proposed.
- Network Layer – An efficient routing technique in sensor networks is essential to preserve energy. Because of the hostile operating environment, random failures of sensor nodes are common, which complicates the routing problem. Dedicated sensor network routing techniques such as SPIN [47] and LEACH [48] are proposed to address these issues.

In order to address the low power requirements of the wireless connections, several standards including P802.15.4 [58] and Zigbee [59] are under development to specify these three layers.

- Transport Layer – Instead of global addressing, attribute based naming is used in the sensor networks. Hence, TCP/IP cannot be used. Since sensor nodes have limited memory and processing power, purely UDP-type protocol may be a solution. New transport layer protocols are still investigated for allowing sensor network to interact with other networks.
- Application Layer –It consists of three main protocols which are Sensor Management Protocol (SMP), Task Assignment and Data Advertisement protocol (TADAP) and Sensor Query and Data Dissemination Protocol (SQDDP).
    - SMP allows administrators to interact with the sensor nodes including location finding, data aggregation, power on or off, network configuration and time synchronization by connecting the hardware and software of the lower layers with sensor network management applications.
    - TADAP provides the user software with efficient interface which allows users send their interest to sensor node and the sensor nodes can also advertise their available data to the users.
    - SQDDP provides the user interface to handle the queuing functions.

Since the lowest three layers, the physical, network and data link layers are used to control the network communication mechanism, they can be reused in the system-level test interface. Based on the applications of the sensor networks, the transport and application layers need to be modified accordingly to provide the functions of initializing the DFT features on the remote nodes. Once all these requirements are set, designers can setup the system-level test interface for their sensor networks.

Figure 4.6 shows the conceptual block diagram of applying the layered design methodology to the test interface for the sensor networks. As discussed above, the system interconnect architecture is unchanged and reused. The transport and the application layers of both the receiver (sensors node) and the transmitter (Test Management Node) need to be modified in order to initialize and control the testing process. Task Management Node should include the Test Controller User Interface that allows users initializing and controlling the DFT features of the sensor nodes. External test data may or may not be needed based on the local DFT features that the sensor nodes support. Data can be imported through the Task Management Node. The Sensor Test Interface provides the service of the modified transport and application layers of the senor node which handles and responds to the test control commands from the Task Management Node. By integrating the test interface into the system, the general sensor networks requirements including the scalability and the low power consumption are fulfilled in this scheme.

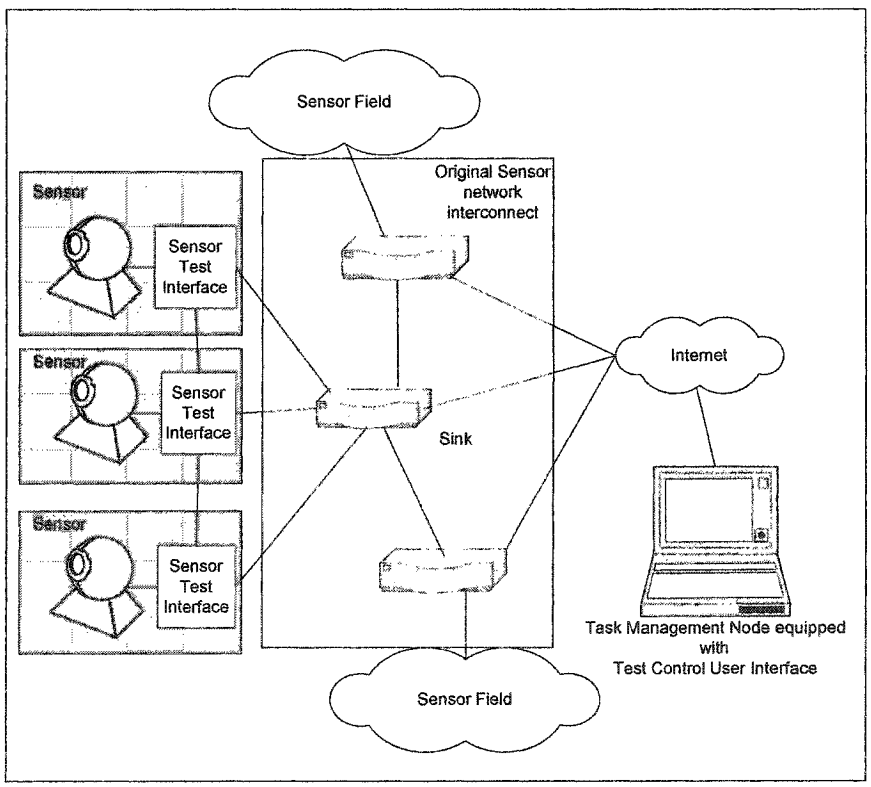

Figure 4.6 Proposed Sensor Networks Test Interface

4.3.4 Case Studies: Sensor Node based on MSP430 Architecture

Figure 4.7 shows the block diagram of the architecture of a wireless sensor node. Measured environmental data are captured by the sensor. It is transmitted to the microcontroller for computation through the analog to digital converter (ADC). The microcontroller is also connected to the sensor networks by the RF transceiver. Once the request is received, the captured data is transmitted to the network through the transceiver.

One of the typical microcontrollers is the Texas Instrument MSP430 [56]. In the MSP430 architecture, a 16-bit RISC CPU and peripherals are combined by using a von-Neumann common memory address bus (MAB) and memory data bus (MDB), as shown in Figure 4.8.

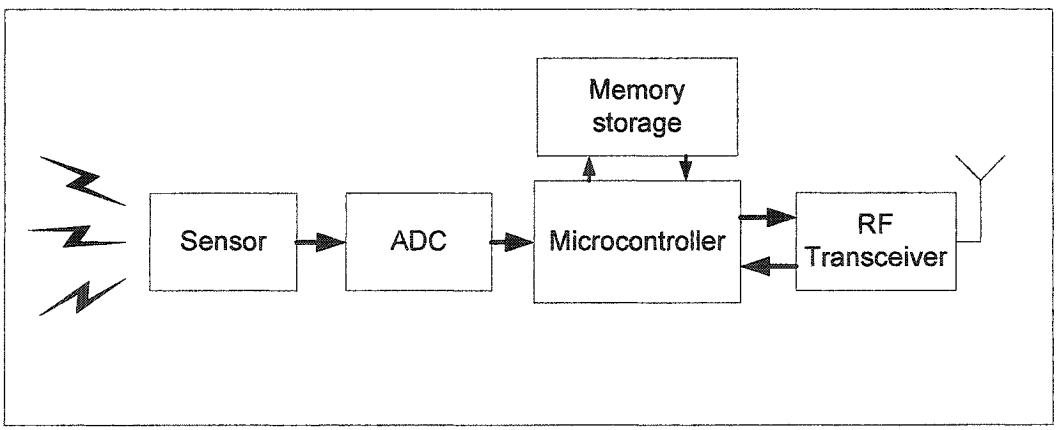

Figure 4.7 Block Diagram of Sensor Node

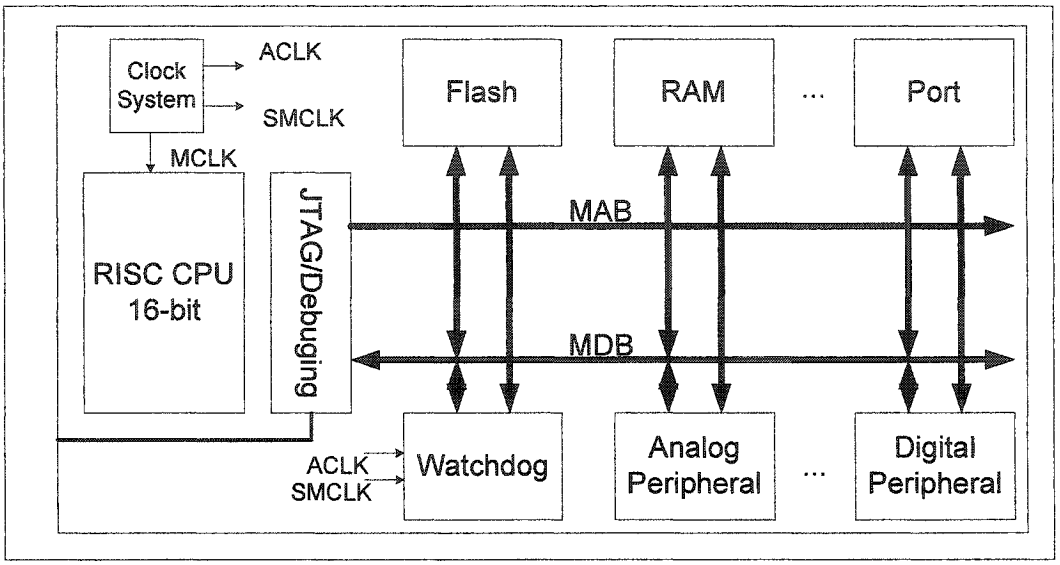

Figure 4.8 MSP430 von-Neumann Architecture

The MSP430 preserves the energy by turning off the CPU and the peripherals in different operation modes. A 12-bits ADC is included for measurements. JTAG is also supported for testing and in-system development.

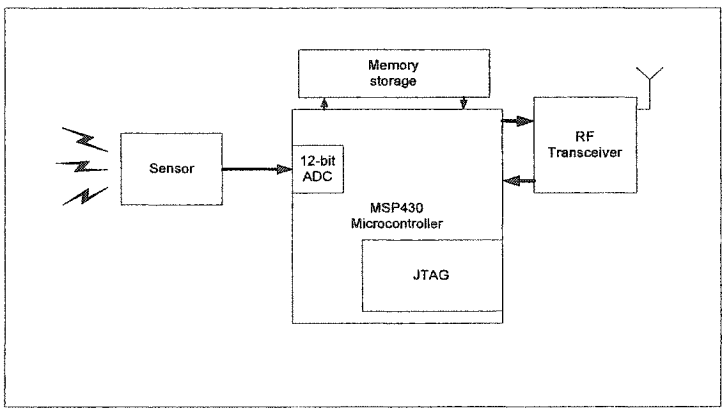

Figure 4.9 Sensor Node Based on MSP430

Figure 4.9 shows a sensor node design based on the MSP430 architecture. Based on the layered model that is presented in Section 4.3.3, the RF transceiver handles the services of the physical, data link, network and transportation layers. The test interface can be implemented in several different ways.

4.3.4.1 Test Interface Module

Since MSP430 includes the JTAG module, it is logical to use it for the system-level testing. As discussed in Chapter 2, the JTAG module is controlled by the TMS pin and test data is scanned into the microcontrollers through the TDI pin. During the testing process, CPU should be stopped while the memory location is accessed by the JTAG module. In order to achieve this, a Test Interface module is added to control the JTAG module.

4.3.4.1.1 JTAG Control by CPLD

As the JTAG module is a state machine that allows the test data to serially shift in and out of the target devices, using programmable logic devices to control them becomes a possible solution.

Figure 4.10 shows the block diagram of the sensor node equipped with the combinational programmable logic device (CPLD). CPLD is used to implement Test Interface Module which is a state machine that controls the boundary scan process.

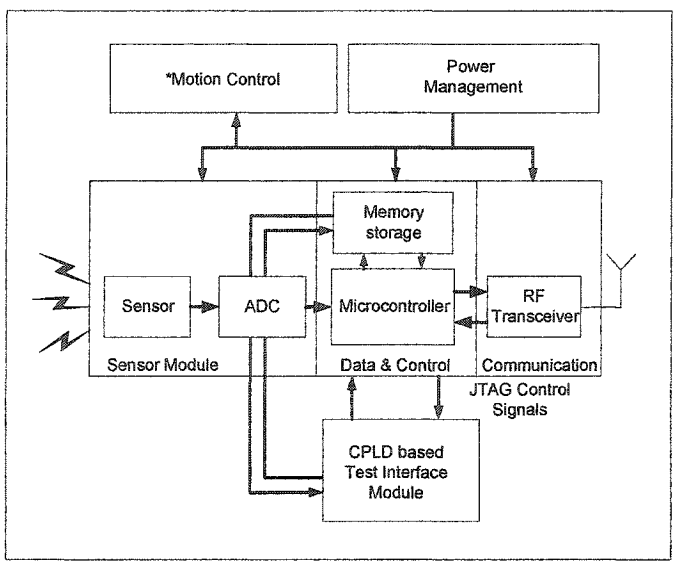

Figure 4.10 CPLD Based Test interface Module

Once the microcontroller received the *test start* command, it enters the self-test mode and waits for the import of the test vectors. Notice that the testing process won't start unless complete test vectors are received. This ensures that the testing process won't be interrupted by the data loss due to the poor communication environment.

Once all the test data is received and buffered in the local memory, microcontroller activates the testing process by sending a *start* signal to the Test Interface Module. From that point, Test Interface Module starts the boundary scanning test by controlling the TMS, TACK pins and the processor of the microcontroller is haltered. Test vectors are read from the local memory storage and shift in the JTAG module through the TDI pin. Test data which has been shifted out from TDO pin will store in the memory and can be used for analyze locally by the microcontroller after the test is completed.

Since the Test Interface Module is a simple state machine, the design is straightforward. However the cost of the CPLD is the main disadvantage in this scheme. Consider the cost of the microcontroller is only less than 1 US dollar, the cost of the CPLD is higher that the cost of the whole sensor node which makes this approach is not preferable.

4.3.4.1.2 JTAG Control by Microcontroller

Since the sensor network dedicated microcontrollers are cheap, second microcontroller is used instead of the CPLD to handle the JTAG control as shown in Figure 4.11.

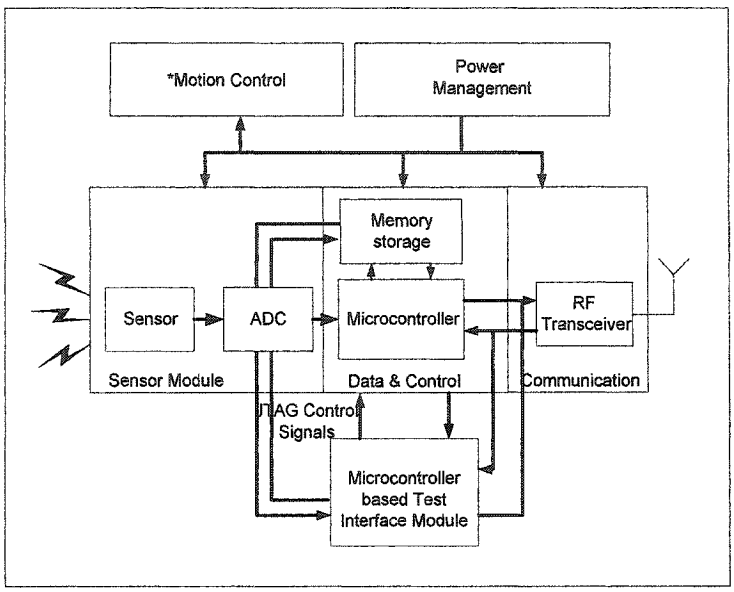

Figure 4.11 Microcontroller Based Test Interface Module

Unlike the CPLD approach, both microcontrollers can communicate with the wireless sensor network through the RF transceiver. During normal operation, only the Data Controlled microcontroller is connected with the transceiver. Test Controlled microcontroller (Test Interface Module) stays in the idle mode and waits for the test command. Once it is received, the Data Controlled microcontroller will activate the test process by sending the Test Start signal to the Test Interface Module. Similar to the CPLD approach, data controlled processor under tested will be deactivated and externally controlled by the Test Interface module through the JTAG module. The main difference here is that test vectors don't need to be buffered prior the test process. Moreover test process can be interrupt by the consumer since Test Interface Module gains the control of the transceiver during the testing process. This provides the real time control of the sensor node even during testing.

Since all microcontrollers support the software programming through the JTAG port, the dual processors architecture also increases the flexibility and enables the in-circuit programming or software upgrade. As the cost of microcontroller is much lower than CPLD, the hardware cost of this scheme is acceptable.

In this testing scheme, the whole testing process is controlled by the test software. Designers can change the testing routines easily without any hardware modifications. Additional applications such as online programming can also be enhanced by modifying the software program. Notice that other microcontroller can also be used as the Test Interface Module. MSP430 is chosen in this case since it is easy to program and the price is low.

However this testing scheme is based on two assumptions. First, since the network connections are reused for test control, a reliable and error-free communication channel is required. Second, in order to control and monitor the testing processes, a well established Transport Layer protocol should be designed to ensure the reliable data delivery. Unfortunately, current sensor networks failed to meet these two requirements. Due to the hostile operating environment, wireless communications in the sensor networks are unreliable. One solution is to increase the signal level of the transmitting data. However, this is not achievable due to the low power consumption requirement. Low power RF transmissions are commonly used in the sensor networks. Fortunately, since the vast majority of sensor networks applications such as temperature measurement, can tolerate the occasional loss of the measurement readings, reliable data delivery is not required. For the testing purposes, the reliable data delivery is required for transmitting the test control and monitoring signals.

To complicate the situations, there is little work on the design of a reliable transport layer for the sensor networks. Pump Slowly, Fetch Quickly (PSFQ) [53] is the only reliable transport layer protocol that is being proposed for the sensor networks currently. Instead of the traditional end-to-end data recovery mechanisms, PSFQ uses the hop-to-hop error recovery scheme. In the sensor networks, data is exchanged by multi-hop forwarding techniques and errors accumulate exponentially over multi-hops. PSFQ allows the intermediate nodes taking the responsibility for error detection and recovery. As a result, the errors accumulated can be eliminated. Report operation is also supported in this scheme

to provide the data delivery status information.

Although PSFQ seems promising in providing a reliable data delivery mechanism, it is still in the early development stages. Moreover, it is designed based on the assumption that data errors are due to the poor quality of the wireless link instead of the traffic congestions. In order to make the proposed sensor networks test interface feasible, further research on the reliable transport layer protocol for the sensor networks is needed.

4.4 Conclusions

In this chapter, we examined the generalizations of the proposed layered approach designed test interface by extending its functionality and applying it to the distributed systems. Since the proposed test interface provides a simple test access channel to various modules within the system, it can also be used for programming and software upgrading purposes. We demonstrated it by modifying the proposed InfiniBand test interface to provide the programming feature.

The flexibility of the layered design approach allows the designed test interface application to different systems. This generalization property is shown in the sensor networks, which have the many system design requirements including scalability and lower energy consumption. Standard MTM-Bus architecture cannot be used and layered design methodology can solve these problems.

Chapter 5 - Systems Performance and Measurements

5.1 Overview

One of the big challenges in today system design is the data integrity which is affected by the noise and interference. Modern complex systems are not only large in size but are also equipped with the high speed interconnect. As a result, small delay and interference that were negligible in the past will create lot of problems. Communication problems such as the Electromagnetic Interference (EMI) and noise interference, transmissions delay and jitter are also raised by the next generation point-to-point switching based interconnect standard such as PCI-Express and InfiniBand.

In this chapter, we first introduce the performance indicators such as BER, SNR and jitter in the communication systems and show their importance in the system design in Section 5.2. The current jitter measurement techniques and their advantages and disadvantages are studied in Section 5.3. In Section 5.4, we examine a new FPGA-based BER testing scheme and validate it by comparing with the existing industrial BER tester. Based on the new FPGA-based BER testing scheme, a new jitter measurement scheme for the high speed interconnect architecture is proposed in Section 5.5. This chapter is concluded in Section 5.6.

5.2 Communication System Performance Indicators

5.2.1 Signal-to-Noise Ratio (SNR)

Due to the physical limitation of the transmission media, the amount of data that can be transmitted at a time is limited. Based on the basic Shannon Law (Equation 1), the channel capacity (C) is controlled by the SNR for a given bandwidth (B). Because of this relationship, communication system designers always try to maximize the SNR by applying different transmission techniques in order to obtain the maximum channel capacity.

$$C = B \log_2(1 + SNR) \qquad \text{Equation 1}$$

Theoretically, digital data is sent as the waveform from the sender to the receiver through different kinds of transmission media, such as fibre-optics and copper wire. In the baseband transmission, different data encoding schemes such as *Unipolar nonreturn-to-zero* (NRZ), *Unipolar return-to-zero* (RZ) and *Manchester* are used to encode both the clock and the data signal. By carefully implementing the encoding and decoding schemes, data can be transmitted with a high SNR. As technology advances, the baseband transmission no longer fulfills the demand and different kinds of modulation schemes such as QPSK are used to shift the transmitting signal to different frequency. Lately, spread spectrum transmission scheme such as CDMA are widely used because they allow multiple users accessing the same channel by modulating the data with a specific pseudo random sequence.

5.2.2 Bite Error Rate (BER)

Bit Error Rate (BER) is defined as the probability bit error. It is the ratio of the number of the received error bits to the total number of the received bits. The

lower the BER, the less error occurs, which implies a better communication system. More importantly, BER is closely related to the Signal-To-Noise Ratio (SNR). Figure 5.1 shows the BER and SNR relationships in the baseband transmission of different data encoding schemes. Since the performance of the communication is affected by various factors, such as the type of transmission waveform, the transmission power, the noise of the channel, the time jitter and the data encoding and decoding methods, an efficient BER test is necessary in the production phase to ensure the quality of the design.

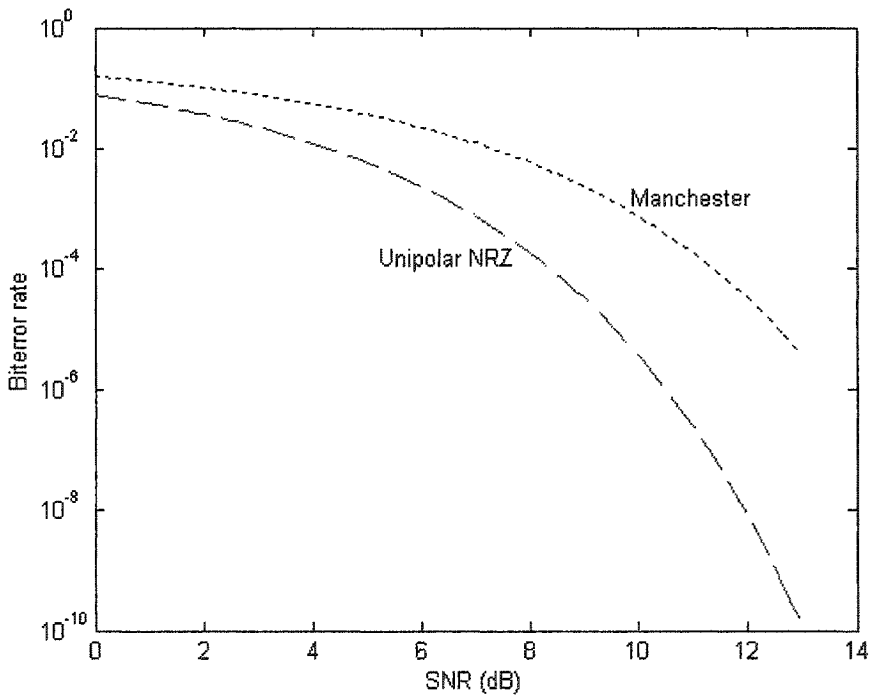

Figure 5.1 BER vs SNR for Baseband Transmission of Different Encoding Schemes

5.2.3 Jitter

Jitter is defined as

> *The deviation from the ideal timing of an event. The reference event is the differential zero crossing for electrical signals and the nominal receiver threshold power level for optical systems.* [54]

Jitter is used to describe the timing errors within the system. Due to the cascade

setup of the multiple clock recovery and synthesis schemes in the modern system, jitter is accumulated through the different repeaters and transponders. As a result, transceivers are needed to tolerate the jitter in order to compensate these problems.

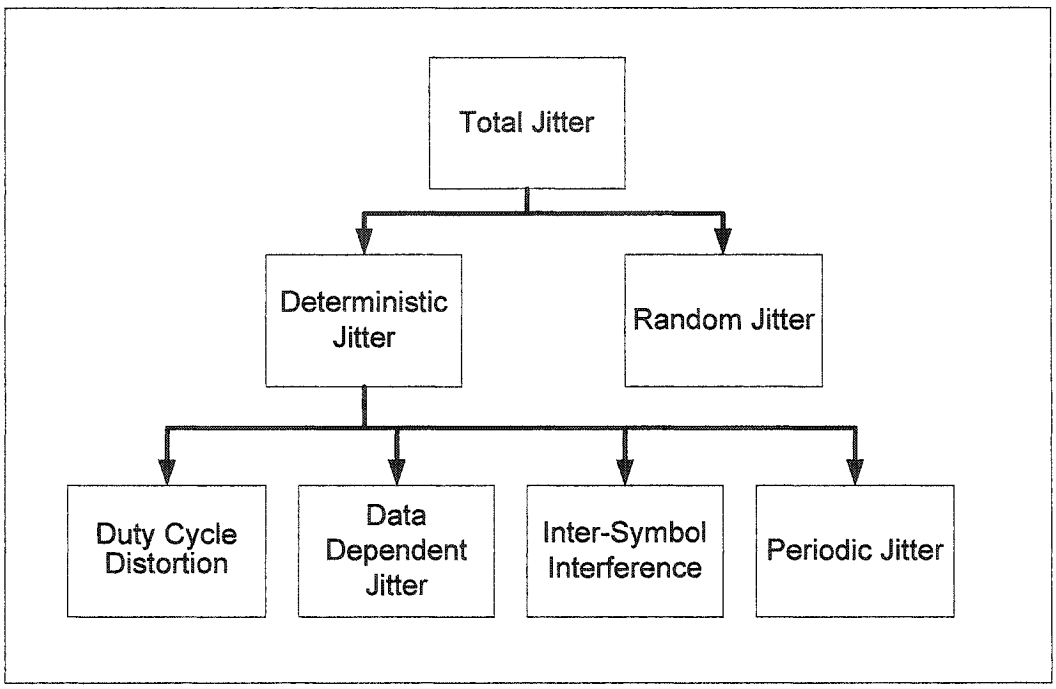

Figure 5.2 Jitter Classification Diagram

Jitter [43] consists of different components such as deterministic jitter (DJ), random jitter (RJ), sinusoidal/periodic jitter (PJ) and total jitter (TJ). Their relationship is shown in Figure 5.2.

- Total Jitter (TJ) – It is the summation of both the deterministic and the random jitter through the equation: $TJ = DJ + n * RJ$ while n is the number of the standard deviations corresponding to the required BER. In general, based on this equation, TJ is overestimated since the maximum RJ will not always coincide with the maximum DJ.
- Deterministic jitter (DJ) – It is always bounded in amplitude and has specific causes. It is typically caused by the crosstalk, EMI, simultaneous switching outputs, transmitted data patterns and other interference signals. Based on the interference sources, there are four different kinds of DJ.

- Duty Cycle Distortion (DCD) – It is defined as the difference of the mean pulse width of '1' when compared with the mean pulse width of '0' in a alternating bit sequences (01010..). This generates the period variations in the bit streams and is only affecting the clock signals.
- Periodic Jitter (PJ) – This is mainly caused by the crosstalk of the EMI sources such as power supply noise. Both clock signals and data signals are affected by the PJ.
- Inter-Symbol Interference (ISI) – Based on the different bit stream patterns, there is a time difference to reach the receiver's threshold value. Consider two different bit streams 1,0,1,0... and 0,0,0,1,1,1,1,..., it is obvious that the time to reach the receiver's threshold is shortened in the alternating patterns. Since the run length of four lows produces higher negative amplitude, it requires more time to change the bit value. This time difference creates the ISI and affects the data signals.
- Data Dependent Jitter (DPJ) – Similar to the ISI, different transmission bit patterns will also create the jitter. Based on the different bit patterns, the amplitude of the '1' in a long stream of '1's will be higher than the '1' in an alternating bit pattern. The voltage difference causes DPJ and affects the data signals.

- Random Jitter (RD) – Random jitter is unbounded and can be described by the Gaussian distribution. It is caused by the Gaussian electrical noise within the system. Timing errors are created when the noise is affecting the slew rate of the signals. Due to its random nature, it affects both the clock and data signals as well as the long term reliability of the system.

In order to ensure the data transmission quality, every communication standard has different specifications of each jitter component as shown in table 5.1. All these jitter components can be inferred by the measurement of the BER within a data eye.

Standard	TXDJ	TXTJ	RXDJ	RXRJ	RXTJ
Fiber Channel	0.38 UI	0.65 UI	0.38 UI	0.22 UI	0.7 UI
InfiniBand	0.17 UI	0.35 UI	0.41 UI	-	0.65 UI
XAUI	0.37 UI	0.35 UI	0.37 UI	-	0.65 UI

Table 5.1 Jitter Specifications of Different Communication Systems

5.2.4 Eye Diagram

As shown in Figure 5.3, eye diagram is the result of cascading many cycles of logic high and logic low transitions together. Ideally, every cycle of the data transition should have the same logic level (either positive or negative), rising time and falling time. In reality, due to the interference such as noise and jitter, the logic levels and the transition time are not the same in every data cycle and can create jitter.

The probability density function (PDF) of the jitter can be plotted by taking the histograms of transitions at the threshold level, as shown in Figure 5.3. Notice that the left and right histograms have a fairly equal shape with the same height. Based on the assumption that the general bit streams have equal amount of 1s and 0s, the same amount of the high to low and low to high transitions is present.

The probability of error transitions due to jitter is affected by the sampling point. Sampling in the middle of the data decreases the probability of errors since signals are settled down and there is less variations of signal level in that region. On the other hand, if the data is sampled close to the two threshold regions, it increases the probability of errors, as shown in the PDF curve in Figure 5.4.

The amount of the jitter is represented by the sum of the area from the sampling point to tail of the left histogram (+ ∞) and the area from the tail of the right histogram (- ∞) to the sampling point. As seen in Figure 5.4, the receiver should

ideally sample at the center of the data eye which causes the lowest possibility of the transition errors due to jitter.

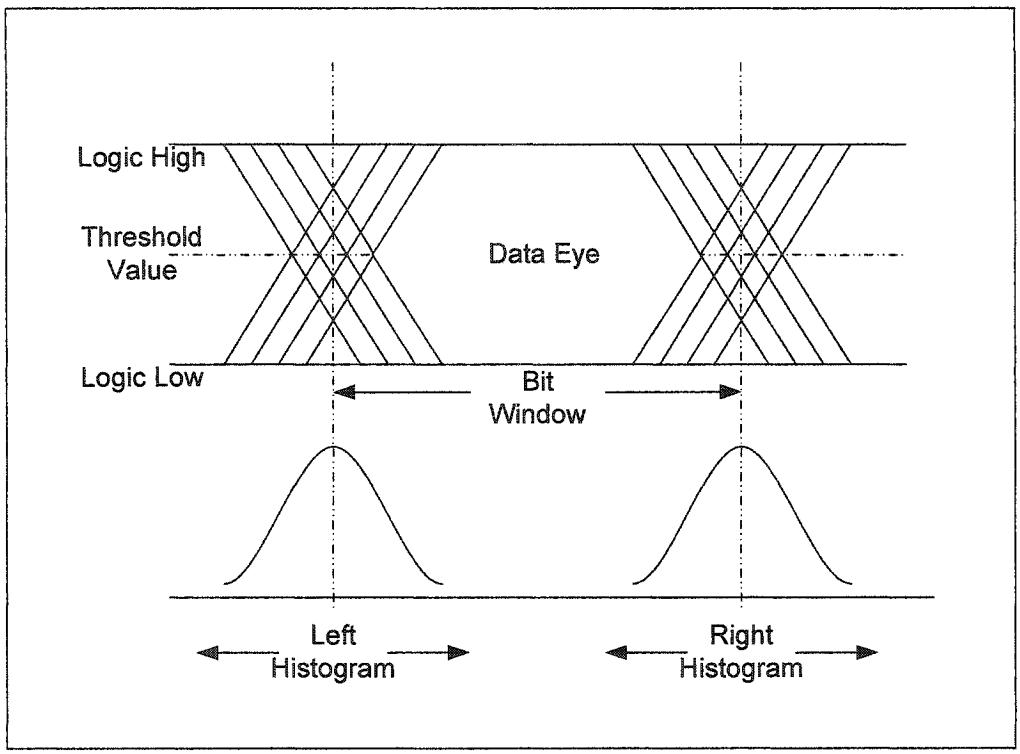

Figure 5.3 Eye Diagram

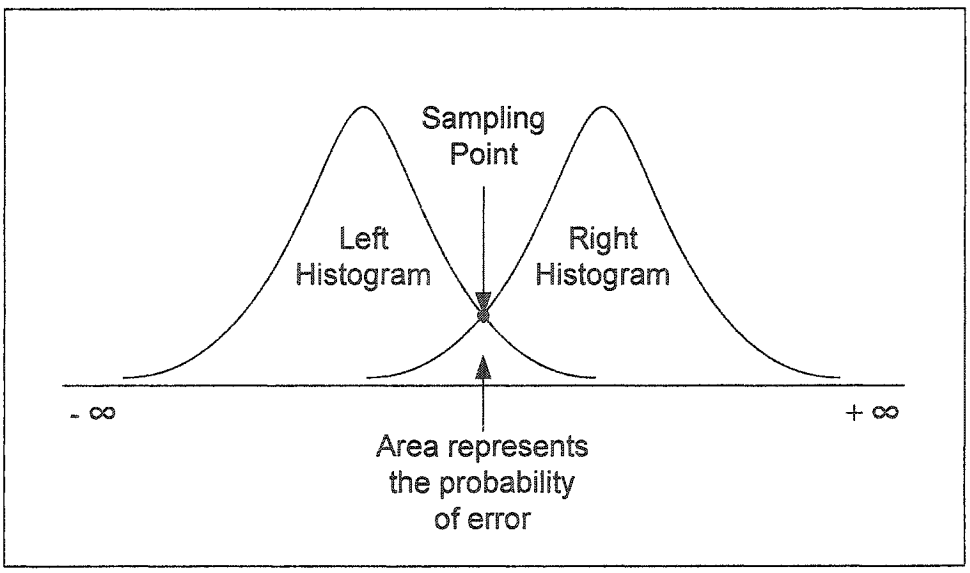

Figure 5.4 PDF of Jitter Error

5.3 Jitter Tolerance Test Methodologies

As seen in the previous section, jitter and BER are closely related. Before we discuss the general jitter tolerance test methodologies, it is important to first have an overview of the BER testing.

5.3.1 Bit Error Rate Tester (BERT)

Bit Error Rate Tester (BERT) is used to measure the BER. The BERT block diagram is shown in Figure 5.5. BERT contains two main components: Pattern Generator and Error Detector.

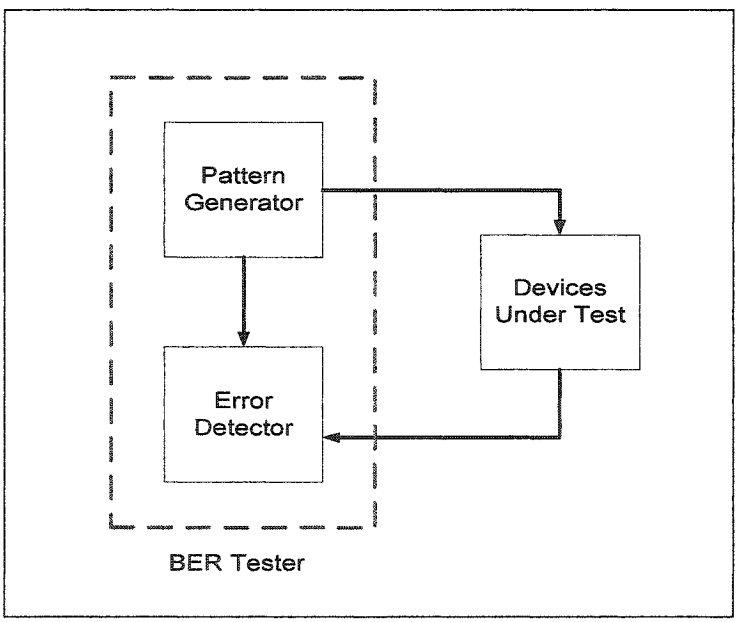

Figure 5.5 Conceptual BERT Block Diagram

Pattern Generator is providing the stimulus to the Device Under Test (DUT). Bit patterns generated are sent to the Error Detector for comparison. Error Detector is responsible for bit error detection by comparing the output data from the DUT with the expected data.

BER bathtub curve is usually used for jitter measurement and BER can be

determined by multiplying the probability of the error due to jitter and the probability of transitions occurring. Normally the probability of transitions is the average transition density (Γ). It is around 50% in the typical data stream. By applying this concept, the BER becomes:

$$BER_{left}(\tau_{sample}, W, \sigma) = \Gamma_{transition} * \int_{\infty}^{\tau_{sample}} JT(\tau, W, \sigma)\delta\tau$$

$$BER_{right}(\tau_{sample}, W, \sigma) = \Gamma_{transition} * \int_{\infty}^{\tau_{sample}} TJ(\tau - UI, W, \sigma)\delta\tau$$

where JT and TJ is PDF of the jitter error for the high-to-low and low-to-high transitions and τ is the sampling position. The total BER due to jitter is the sum of these two functions:

$$BER_{total}(\tau_{sample}, W, \sigma) = BER_{left}(\tau_{sample}, W, \sigma) + BER_{right}(\tau_{sample}, W, \sigma)$$

By measuring the sample point, τ_{sample}, which is swept between the two eye crossings, the BER bathtub curve is obtained.

5.3.2 Jitter Source Injection

The receiver's ability to recover error-free data from the jittery inputs is important in high speed communication systems. It can be obtained by measuring the BER of the DUT with a well controlled jittery signal. Figure 5.6 and 5.7 show the block diagrams of the jitter tolerance test setup based on two different standards [22] [23] and three kinds of jitter sources, Deterministic Jitter (DJ), Random/Gaussian Jitter (RJ) and Sinusoidal/Periodic Jitter (PJ) are provided and added to the bit streams.

The source of the DJ is mainly due to bandwidth limitation of the transmission

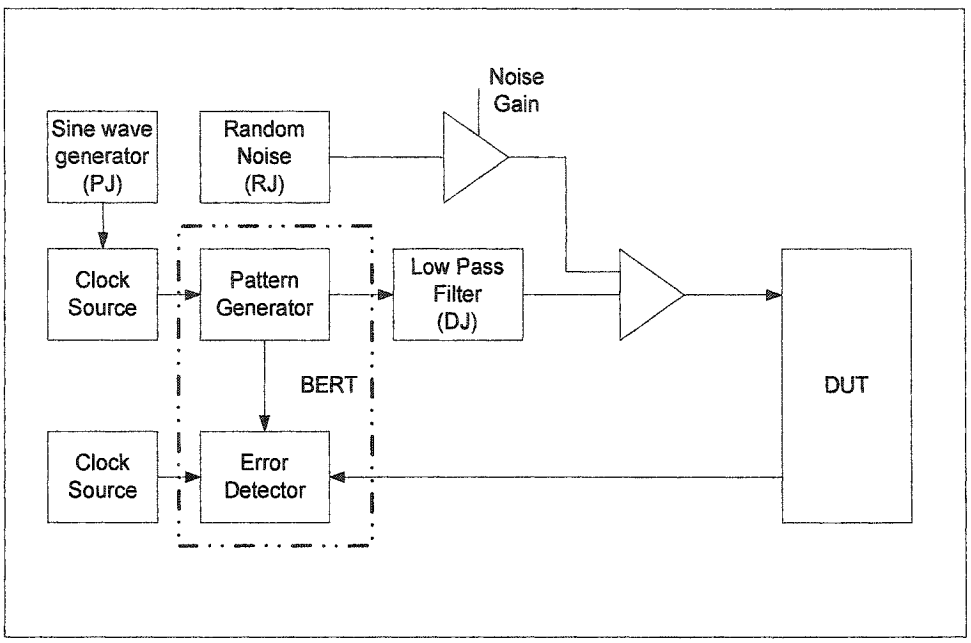

Figure 5.6 Jitter Tolerance Test Setup for Fibre Channel

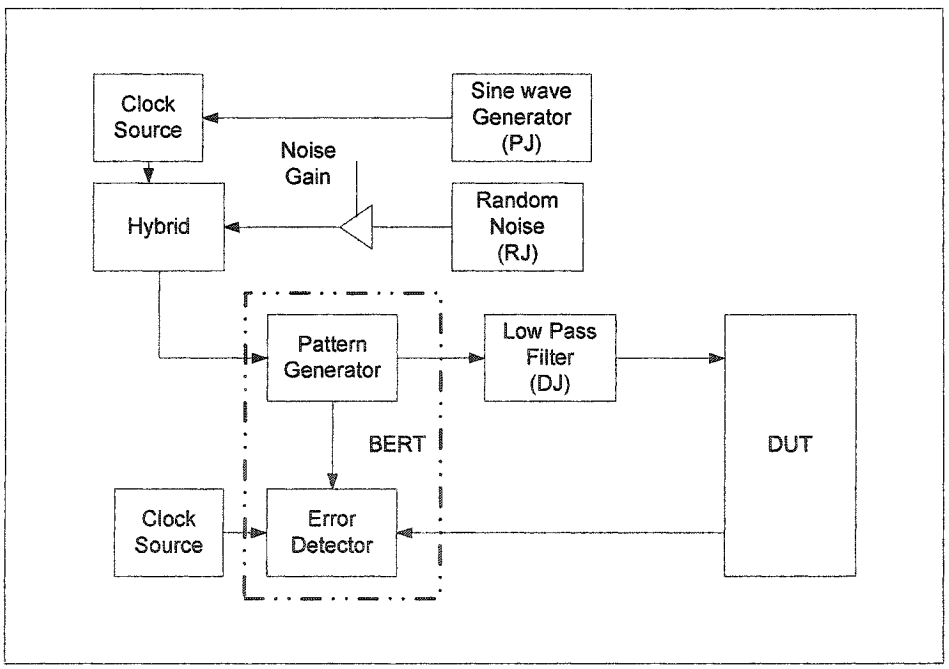

Figure 5.7 Jitter Tolerance Test Setup for XAUI

media and is emulated by using a long cable or lowpass filter. They are tunable by altering the cable length or filter characteristics. Due to the differences of the

standard requirements, different RJ injection methods are shown in Figure 5.6 and Figure 5.7. In Fibre Channel, RJ is directly injected into the data stream, while it is injected into the clock signal path of the pattern generator in the XAUI standard. Notice that a clean clock signal is needed for the Error Detector because of the presence of the PJ injection. Further, two separate clock sources are required.

5.3.3 Jitter Test Methodologies

Three commonly used jitter measurement methods [51] will be discussed in this sub-section.

5.3.3.1 Direct TJ Measurement

As each interconnect standard specifies the TJ at a specified BER level, it can be directly measured with the previously discussed jitter tolerance test setup. The main advantage of this method is its absolute resolution and accuracy. However, it suffers from a long testing time. Considering the XAUI standard, TJ is specified at 10^{-12} BER level which means the data stream with minimum 10^{12} bits should be used. Testing such long bit streams is time consuming and inefficient.

5.3.3.2 Oscilloscope Measurement

Modern high speed sampling scopes can collect and present the output jitter data in terms of eye diagram and timing histogram. They provide the feature of comparing the captured eye diagram with a specific "eye mask". Eye mask is the transmitter eye diagram that provides the characteristics of the transmitter pulse shape. It includes the minimum width of the eye opening, signal level, rise time, fall time, voltage overshoot and undershoot. However, using this to measure the TJ is not accurate since the sampling rate is too low to capture all the RJ components.

5.3.3.3 BERT Scan

BER bathtub curve is obtained by measuring the sample point, t_{sample}, which is swept between the two eye crossings. By using 2 measurement points, individual jitter components can be estimated by the following sets of equations [51].

Random Jitter:

$$RJ_{RMS} = 0.5 \left| \frac{t_1 - t_0}{Q_1 - Q_0} \right| \qquad \text{Equation 2}$$

Deterministic Jitter:

$$DJ = UI - t_0 - (2\, Q_0 RJ_{RMS}) \qquad \text{Equation 3}$$

Total Jitter:

$$TJ = DJ + 13.73 * RJ_{RMS} \qquad \text{Equation 4}$$

This estimation method consists of the following steps:

A. Measure the eye opening at two different BERs from the bathtub curve which is BER_0 at t_0 and BER_1 at t_1.

B. Based on the Quality (Q) factor and BER relation in Equation 5, obtain the Q values of these two BERs.

$$BER = \frac{1}{2} erfc\left(\frac{Q}{\sqrt{2}}\right) \approx \frac{1}{\sqrt{2\pi}Q} exp\left(-\frac{Q^2}{2}\right) \qquad \text{Equation 5}$$

C. Substitute all the BERs and Q values in equation 2, 3, 4 and calculate each jitter component.

The main advantage of this method is that the quantity of RJ and DJ can be obtained with only 2 measurement points and without long testing times. However, it is based on the assumption that two tail sections in the bathtub curve are due to the RJ which are Gaussian and symmetrical. As shown in [24], there is a 2-3 ps rms error in RJ measurements. Based on the total jitter equation, this 2-3 ps error is amplified by 14 and creates a TJ error that is larger than 20-30 ps.

5.4 New Proposal for BER Testing

As seen above, all current measurement methods have suffered either from the long testing time (Direct method) or the low accuracy (BERT scan). Moreover no industrial jitter source product is currently available and most of the tests are based on the home-made equipment. In addition, the available BER testing instruments are expensive and do not include any noise generator or injection functions. Setting them up with the home-made jitter generators are difficult and time consuming. This problem will be more severe as the jitter measurement becomes the standard testing requirement when the new high speed point-to-point switching interconnect standards are widely adapted in the industry. Low cost, efficient and easy to setup jitter measurement methodologies are needed.

5.4.1 Existing Industrial Solutions

One of the popular industrial BERTs is the Agilent 81200 Data Generator/Analyzer [38], shown in Figure 5.8. It is made up of two main components that are the personal computer and the BERT modules. The personal computer acts as the Error detector. Special software is used to control the testing process, collect and analyze the received bit patterns. BER Test rack mounts different Agilent BERT modules with specified functions. Agilent BERT uses the modular design and there are two main kinds of modules: clock generator and the I/O modules. Clock modules control the speed of the transmitted bit patterns and enable other external control mechanism such as external clock sources, external input or output triggering. I/O modules provide the input and output interfaces to the Device Under Test (DUT).

One of the main advantages of this modular approach is its flexibility and scalability. The system can be upgraded through adding a faster clock or I/O modules. By using the personal computer for error detection, it cuts down the system cost. Software programs are supplied to analyze the captured data. Users

can also design their own test patterns. However, the setup cost for a complex system is still very high and is proportional to the number of the I/O pins of the DUT. Due to the limitation of the test rack size, DUT with large number of I/O pins may need two or more test racks connected together to perform the test. This increases both the setup cost and complexity. Moreover, no noise or jitter generators are included in these systems and setting them up with the external jitter source will be difficult and time consuming.

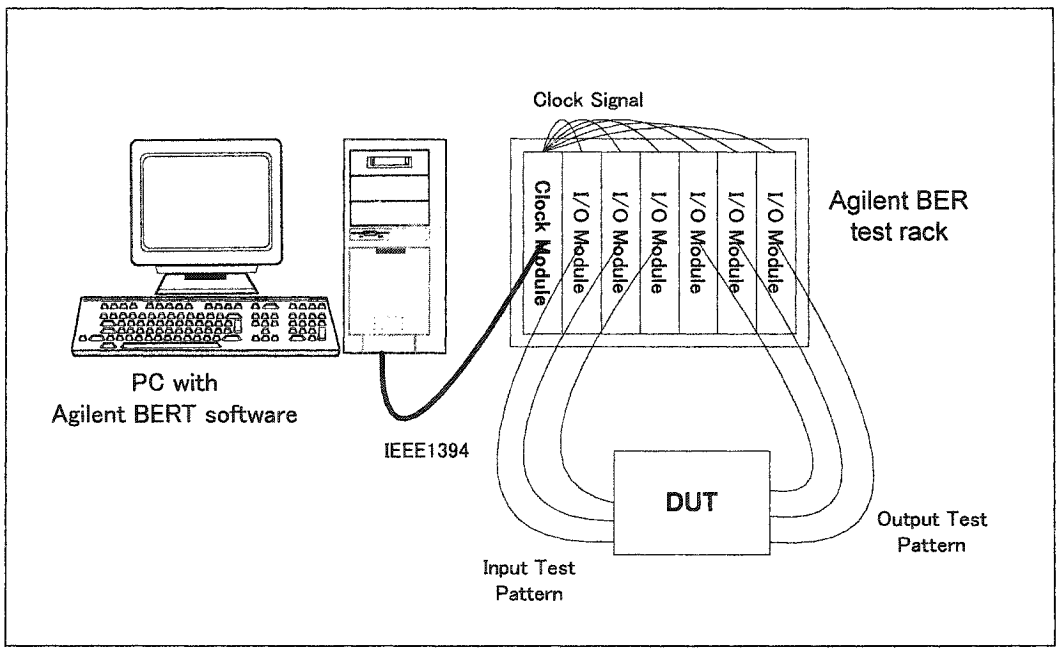

Figure 5.8 Agilent BERT

5.4.2 New FPGA-based BER Testing Scheme

Currently, lots of studies are undertaken in the jitter measurement area [24], [26], [27], [28], [29], [30]. Most of them are concentrated on how to generate and control different kinds of jitters. However, none of them have considered proposing a better BER and jitter test setup that can facilitate a fast and low cost testing process. Because of that we created a test setup for Yongquan Fan's work on FPGA-based BER testing methodology [31] to address these issues.

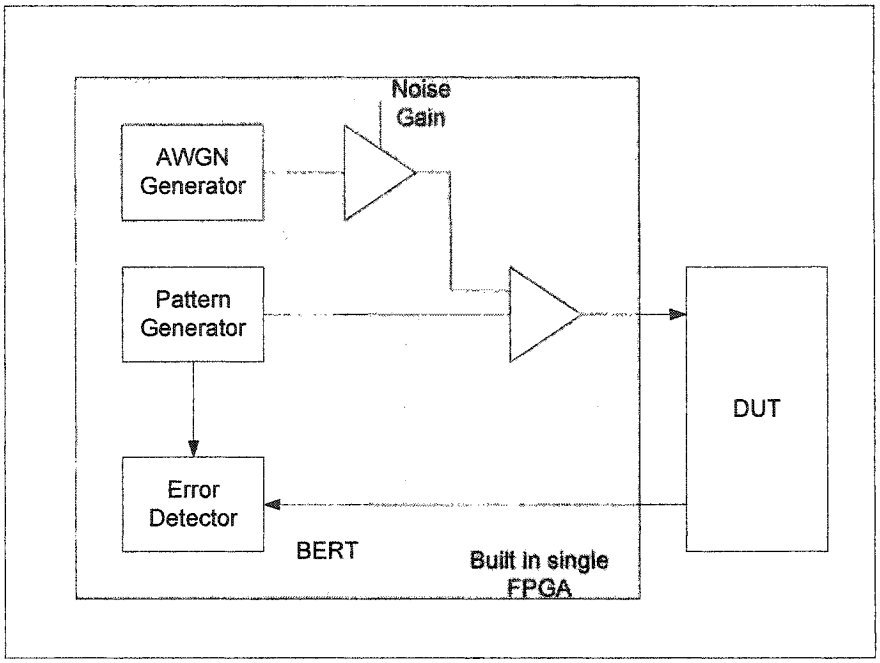

Figure 5.9 Yongquan Fan's proposed BERT testing scheme

Figure 5.9 shows the block diagram of Yongquan Fan's proposed BER testing scheme with the additive Gaussian noise generator. There are three main components in this scheme: Pattern Generator, AWGA Generator and Error Detector. Pattern generator provides the stimulus to the DUT. Error detector is responsible for output comparison. The additional noise generator is responsible for noise injection which is normally not included in the traditional BERT. It emulates the channel characteristics by adding the controlled Gaussian noise to the data streams. For better precision, the data stream is expanded to the same word width as the noise generated by the AWGN generator. This provides the flexibility and accessibility for the designer to measure and analyze their products in the extreme conditions while decreasing the test setup cost. Although up to this point, jitter source is not included in the proposed this testing scheme, adding various jitter sources is feasible.

5.4.3 Performance

In order to understand the performance of the proposed BER testing scheme, I

participated in the performance tests for the FPGA-based BERT. The main goal is to validate the functionality of the FPGA-based BERT with the additive Gaussian noise generator.

5.4.3.1 FPGA-based BERT Design

The proposed FPGA-based BERT is designed as shown in Figure 5.10. Our BERT consists of three main components: Pattern Generator, Testing Synchronization and Error Detector

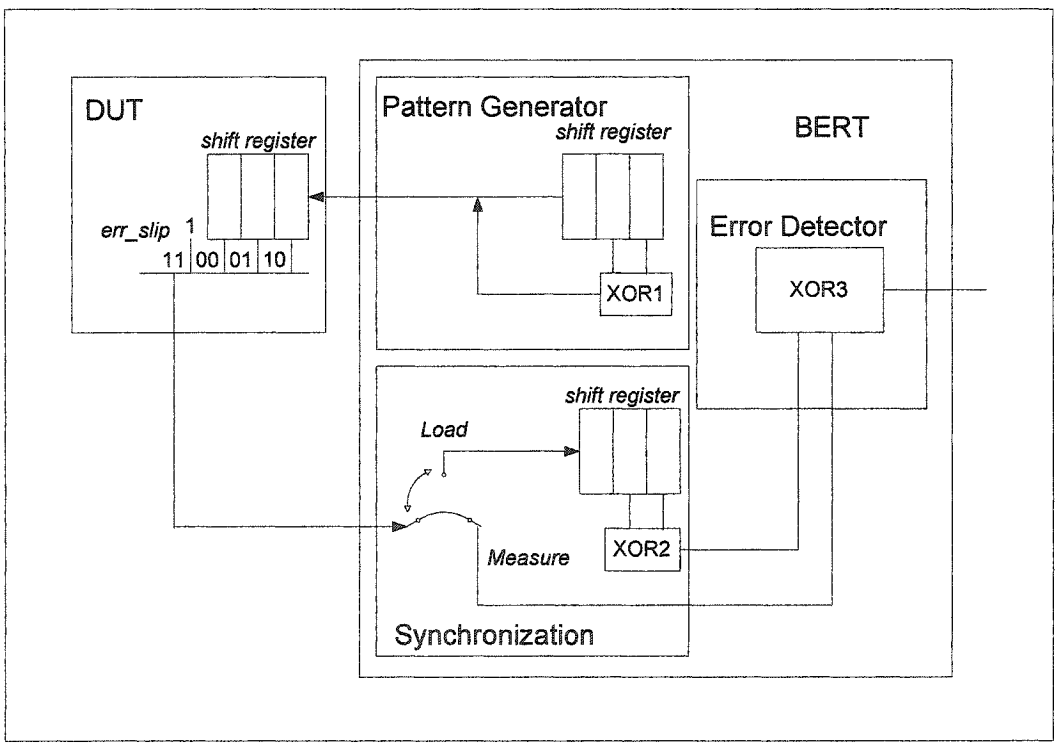

Figure 5.10 FPGA-based BERT Design

- Pattern generator – In our BERT, Linear Feedback Shift Register (LFSR) is used to generate the pseudo random bit sequence (PRBS) as shown in Figure 5.11. PRBS is used to simulate the transmitted signals and repeats after certain number of bits. The sequence length is determined by the stage number of the shift register, *n*. The

maximum period of the sequence is $2^n - 1$. By using the bigger number of n, PBRS is closer to the real transmitted data. However, a long synchronization time is needed for a big number of n. In our design, n is chosen to be 3 and the PRBS would repeat every 7 clock cycles.

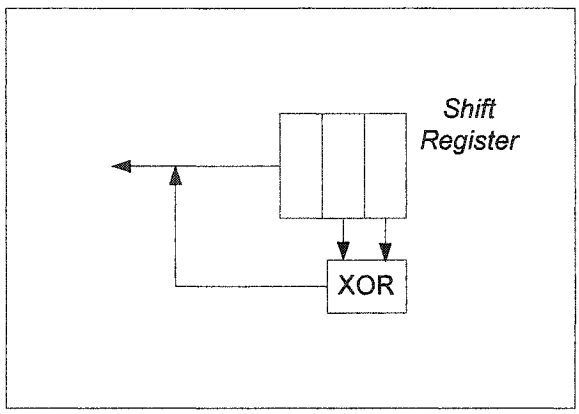

Figure 5.11 Pattern Generator

- Testing Synchronization – DUT needs some time to process the input test sequence and export to the BERT. In order to compensate this time delay, testing synchronization circuit is used to synchronize the DUT output with the expected data. Figure 5.12 shows the Test Synchronization circuit. It is made up by the shift registers with the XOR gate. The stage number of the shift register should be the equal to the stage number n in the Pattern Generator. Before the start of the BER measurement, the transmitted PBRSs are loaded into the shift registers by turning the switch to the *load* state. The stored PBRSs are used as the reference patterns. During BER testing, the switch is turned to the *measure* state and each bit of the *DUTout* sequence is compared with the output of *XOR2*.

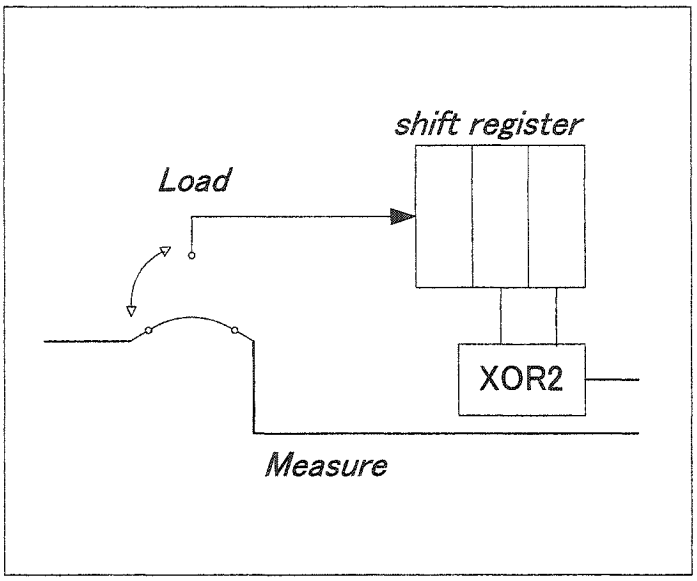

Figure 5.12 Test Synchronization Circuit

Based on our LFSR circuitry (as shown in Figure 5.10), the k^{th} bit of the sequence is generated by the XOR operations of the $(k\text{-}2)^{th}$ and $(k\text{-}3)^{th}$ bits. In our BERT, if the *DUTin* sequence is denoted by $x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_1, x_2, x_3$....., we have

$$x_4 = x_1 \oplus x_2$$

$$x_5 = x_2 \oplus x_3$$

$$x_6 = x_3 \oplus x_4$$

$$x_7 = x_4 \oplus x_5$$

$$x_1 = x_5 \oplus x_6$$

$$x_2 = x_6 \oplus x_7$$

$$x_3 = x_7 \oplus x_1$$

Due to this relationship, *XOR2* can regenerate the test sequences by the XOR operation of the first two bits of the stored PBRS. The regenerated test patterns are synchronized with the *DUTout* signal and are used for comparison.

Notice that in this scheme, the transmitted test patterns need to be error-free during the *load* state. Since the loading operation is only

last for a couple of clock cycles, it is reasonable to assume that the transmitted test pattern is not corrupted under such short period of time.

- Error Detector – A XOR gate, *XOR3*, is used to compare the reference patterns with the *DUTout* signal. The output of *XOR3* is monitored every clock cycle. It remains zero if the transmission is error-free and sets to one when a bit error is occurred.

To verify the functionality of our BERT design, an adjustable shift register is used to simulate the DUT. As shown in Table 5.2, *err_slip* signal controls the delay of the *DUTout*.

err_slip	Effect
00	Delay *DUTout* by 3 cycles
01	Delay *DUTout* by 2 cycles
10	Delay *DUTout* by 1 cycles
11	*DUTout* always 1

Table 5.2 Control Signals for DUT

The simulation waveform is shown in Figure 5.13. In our simulation, the *DUTout* is delayed by 3 clock cycles while the *err_slip* signal is set to "00". Since there is no bit error occurred, the output of *XOR3* should remain zero. As shown in Figure 5.13, it takes 6 clock cycles to complete the synchronization process. The glitches are due to the delays of the XOR gates and the shift registers. They can be eliminated by adding a register before the output of each signal.

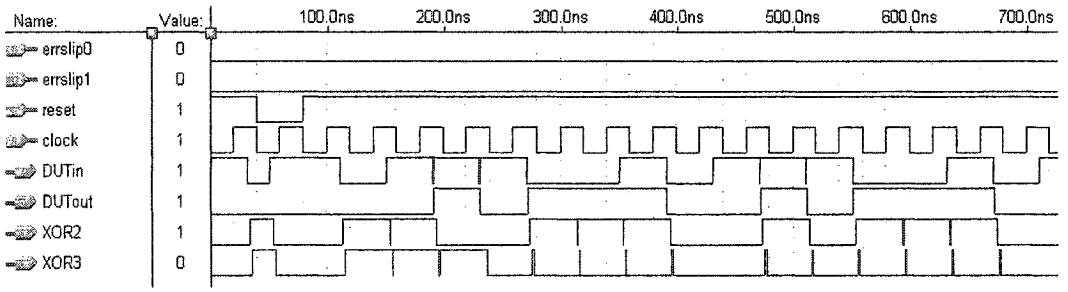

Figure 5.13 Simulation Waveform of the BERT Design

5.4.3.2 Baseband Transmission Testing

In order to validate the performance of our FPGA-based BERT, we perform a baseband transmission testing for our BERT with the adaptive Gaussian noise (AWGN) generator, as shown in Figure 5.14.

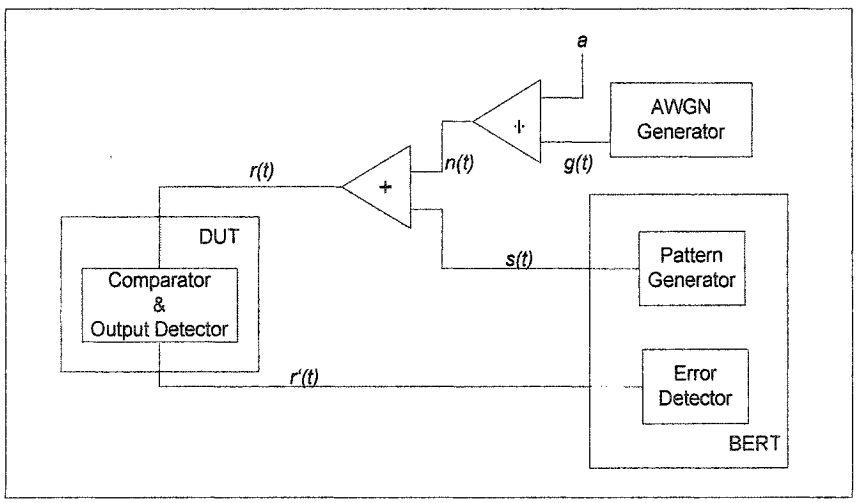

Figure 5.14 BER Testing Setup

In the BER testing setup, our BERT design which is presented in previous section is reused. PBRSs are generated by the BERT as the input test pattern *s(t)*. This test pattern is corrupted by the noise signal *n(t)* which is derived from output of the AWGN generator, *g(t)*. By dividing *g(t)* by the scaling factor, *a*, the amount of the random Gaussian noise that is injected into the data stream can be controlled. Yongquan Fan's AWGN generator which is presented in the [31] was used in our

test setup. It provides both the advantages of high accuracy and high speed. Detail AWGN generator core design can be referred to [31] and [34].

In our testing setup, DUT is made up by a comparator and the output detection circuitry. The noise corrupted test sequence, $r(t)$ is imported into the DUT and compared with the threshold value (T). If the value of $r(t)$ is larger than T, the received bit sequence, $r'(t)$ is set to '1'; otherwise, $r'(t)$ is set to '0'. The received bit sequence $r'(t)$ is imported back into our BERT and compared with the delayed input test pattern $s(t)$ bit by bit. To simplify the design, the threshold value, T, is set to 0.5. Finally, the measured BER is obtained by the ratio of the counted bit errors and the total number of transmitted bits.

In digital baseband transmission systems, there are various serial-bit signaling formats called *line codes*. There are two major categories which are *return-to-zero* (RZ) and *nonreturn-to-zero* (NRZ). With NRZ coding, the binary 1 is represented by a high level (+A volts) and the binary 0 is represented by a zero level. This is also true with RZ coding except that the waveform returns to zero-volt level for a portion of the bit interval. In our testing setup, NRZ coding scheme is used. Theoretically, the error performance of the NRZ is

$$BER = Q(\sqrt{\frac{E}{N_o}})$$

where E is the signal energy, N_o is the noise energy and $\frac{E}{N_o}$ is the SNR.

With NRZ coding, the energy of the signal, E, is 0.5. It is based on the assumption that both data '1' and data '0' have equal occurring opportunity. The noise energy, N_o, is controlled by the variance of the AWGN generator, δ^2.

$$N_o = 2\,\delta^2$$

In our testing setup, an AWGN generator with zero mean and unity variance is used. As shown in Figure 5.14, the output of the generator is divided by the scaling factor, a. This changes the variance of the generator and the new variance

becomes $1/a^2$. As a result, the SNR can be determined by the scaling factor, a by the following equation.

$$SNR = \frac{a^2}{4}$$

The whole setup is ported to the Mercury EP1M120 devices [32] and the measurement result is shown as Table 5.3.

Scaling Factor a	2	3	4	5	6	7	8
Transmitted Bits	12448	12448	12448	20000	1000000	1000000	1000000
SNR (dB)	0	3.53	6.02	7.96	9.54	10.88	12.04
Measured BER	1.62e-1	6.58e-2	2.32e-2	5.65e-3	1.20e-3	1.76e-4	1.62e-5

Table 5.3 BER measurement result of FPGA-based BERT

5.4.3.3 Comparison

In order to validate the performance of our BERT, the whole test is repeated by using the Agilent Data Generator/Analyzer 81200 as the BER tester. The testing setup is shown in Figure 5.15.

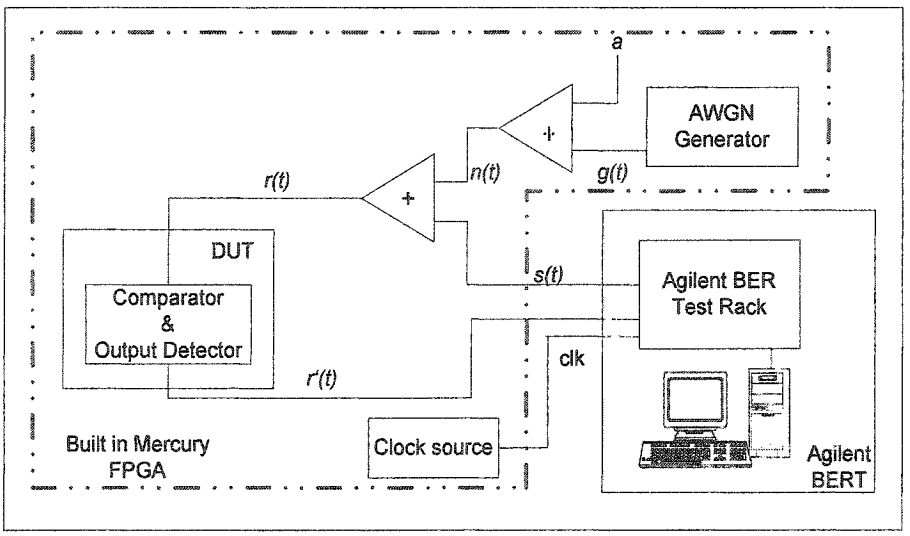

Figure 5.15 BER Testing Setup with Agilent BER Tester

For synchronization purposes, same clock source is used in both the AWGN generator and Agilent BER tester. The measured result is shown in Table 5.4.

Scaling Factor a	2	3	4	5	6	7	8
Transmitted Bits	12448	12448	12448	20000	1000000	1000000	1000000
SNR (dB)	0	3.53	6.02	7.96	9.54	10.88	12.04
Measured BER	1.65e-1	6.61e-2	2.40e-2	5.32e-3	1.31e-3	1.83e-4	1.78e-5

Table 5.4 BER measurement result of Agilent BERT

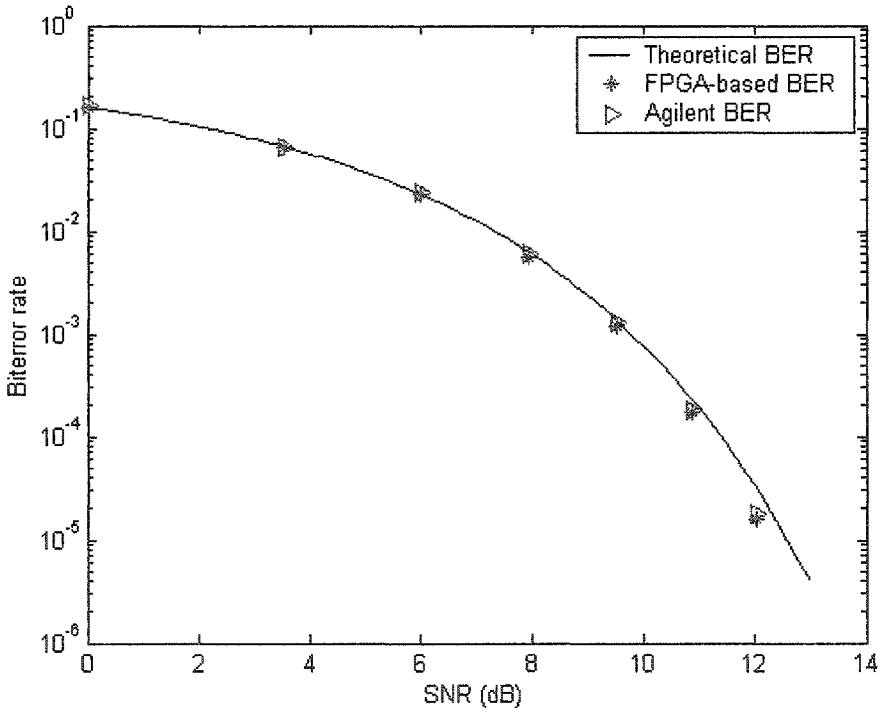

Figure 5.16 Plot of Measured BER and Theoretical BER for Baseband Transmission

To validate the proposed BERT, the BER versus SNR curved is plotted based on the measurements in Table 5.3, Table 5.4 and compared with the theoretical results, as shown in Figure 5.16. Both measured and theoretical results are closely matched with each other which prove that both the proposed BERT and noise

generator work flawlessly. Notice that as the SNR increases, both measured BERs are lower than the theoretical value. It is mainly due to the limited number of transmitted bits. By increasing the length of the test pattern, the measured BER will become closer to the theoretical value. Moreover, the accuracy of the comparator in DUT also affects the measured results. In our test setup, 0.5 is artificially chosen as the threshold value. By carefully adjusting the threshold value, a better performance can be obtained.

5.5 New Proposal for Jitter Measurement

As discussed in Section 3.0, Yongquan Fan's BER testing scheme provides a flexible and low cost solution for BER testing. By implementing the BERT on the FPGA chip, it decreases the complexity of the test setup. Moreover, the flexibility is also increased since adding extra functions can be achieved by re-programming the BERT on the FPGA core, with no hardware modifications of the test setup. In addition to that, Gaussian noise generator is included to this BERT schemes.

Based on the same design methodology, other noise or jitter source can be added to the BERT. They can be either internally designed on the FPGA core or externally connected to it as shown in Figure 5.17. Due to the complexity of the jitter measurements, different jitter sources are required to inject the jitter to different parts of the system. As discussed in Section 5.3, a sine wave generator is used to inject the PJ to the clock source of the Pattern Generator in the XAUI standard. FPGA-based AWGN generator is used as the RJ source and injects the Gaussian noise into the test data. A low pass filter is also included to provide the DJ.

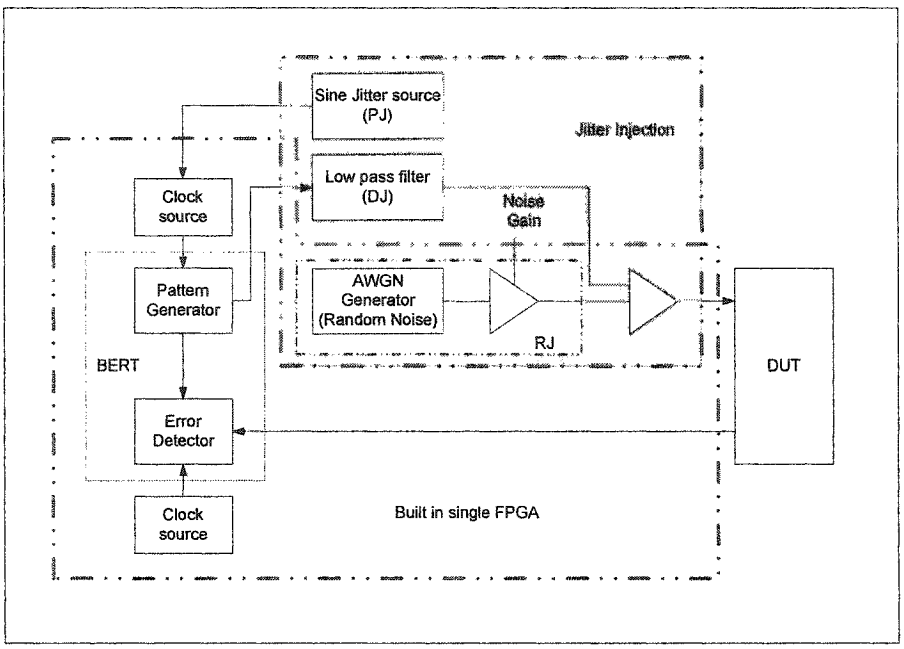

Figure 5.17 Proposed Jitter Test Setup for Fibre Channel

Using the industrial standard BER tester such as Agilent 81200 BERT for jitter measurement is difficult due to the physical limitations. Random jitter requires injecting the Gaussian noise into the clock module directly. controlled by the computer. A specialized clock module provided by the manufacturer is needed to achieve this requirement. As a result, the flexibility is limited and the testing cost is increased. The situation becomes worse if different communication standards or protocols need to be tested. All these issues can be solved by applying Yongquan Fan's proposed BER testing schemes since the whole tester is based on customer design.

Notice that some major issues still need to be addressed to use the proposed BERT for jitter measurement. One of the important issues here is the availability of the jitter source. Although the proposed BERT has shown no compatibility problem, parts of the jitter source such as DJ can be easily implemented by the DSP functions in the FPGA. RJ and PJ may require some customized ASIC designs. Integration of parts of the jitter source into the BERT using FPGA will decrease the testing cost and ease the testing processes.

Secondly, the data analysis power is limited in the current design. Only basic BER measurement is performed and BERT scans and eye diagram are still not available due to limitations of the FPGA devices. More studies in this area are needed in order to enable the jitter measurements.

Although using the proposed BERT for jitter measurement is not currently feasible, this new BER testing scheme provides the opportunity to explore the possibility of performing the jitter measurement on the simple FPGA design. It provides satisfactory results in the BER measurement. Based on the same design methodology, jitter measurement can be accomplished in a cheaper and more flexible way.

5.6 Conclusions

As the next generation high speed interconnect standard such as PCI-Express, InfiniBand and Fibre Channel are increasingly adapted in the industry, all the communication performance metrics become the concern in the system level testing. In this chapter, we discussed issues such as BER, jitter and the available industrial measurement methods. It is obvious that the current solutions do not fit the large scale systems and require a high setup and testing cost. FPGA-based BER testing scheme provides a flexible method to solve this problem by generating the basic BER tester with the Gaussian Noise generator on an FPGA. We validate this testing scheme by comparing the performance with the Agilent BER Tester. Although lots of modifications are needed to enable it for jitter measurement, the proposed BER tester provides an opportunity to tackle this problem in a cheaper and easier fashion.

Chapter 6 - Conclusions

6.1 Conclusions

In this thesis, we discussed some of the challenges raised in the modern system-level testing including the high test speed and flexibility. The newly developed point-to-point switching interconnect standards such as Fiber Channel and InfiniBand impose the high frequency requirement in the system testing. As ITRS predicts, system-level testing becomes more difficult and expensive to accomplish due to the complexity of modern systems. The existing boundary-scan test circuitry is inadequate due to its topological limitation. Traditional ad-hoc or informal design methodology as shown in the PCI, has proven inefficient. Instead, a well-defined test interface should be taken into considered during the early design stage in order to provide efficient and fast testing circuitry, as shown in the AMBA.

A layered design methodology for the system test circuitry has been presented in this thesis and demonstrated in the IBA network system in order to show its advantages, including flexibility and reusability. They are achieved by partitioning the system into the layered model and by integrating the test circuitry into each layer according to the test requirements. Although it is logical to apply this design methodology to the complex network based system, it is equally applicable to the wider class of systems, including System On Chip (SoC).

Since the developed system test interface circuitry is based on the layered design methodology, it maximizes the reuse of the system design. Further, topological limitations are eliminated and all system requirements can be fulfilled. Moreover,

the proposed test interface is flexible and can easily expand its functionality to other operations including in-circuit programming and device upgrading. We have shown in the Sensor Networks example how to achieve scalability and low power consumption. These goals are difficult to accomplish if existing testing standards such as MTM-Bus are applied, while reusing the existing system communication mechanism can meet these requirements easily.

Due to the development of the new high bandwidth interconnect architectures, general high frequency communication concerns including BER and jitter also need to be considered in the system level testing. The situation is even worse here, since efficient industrial jitter measurement and injection instruments are still missing in the current stage. In this thesis, basic BER testing, jitter measurement and injection are described and a new FPGA-based BER testing scheme is examined. Although BER tester is still in the early development stages, it shows the potential of providing a cost effective integrated BER testing for industrial use. Jitter measurement can also be achieved based on this testing scheme with the use of the external deterministic and probabilistic jitter source.

6.2 Future Work

To show that the layered design approach can be effectively used for the system-level test circuitry design, it needs to be applied to a wider class of systems. Moreover, advancing the BER testing functionality and system integration is needed for our proposed jitter measurement scheme.

References

[1] *International Technology Roadmap for Semiconductors, Test and Equipment*, 2001.Edition, ITRS.

[2] The Peripheral Component Interconnect Special Interest Group, *PCI Local Bus Specification, Revision 2.1*, June 1, 1999.

[3] The Peripheral Component Interconnect Special Interest Group, *PCI-X Protocol Addendum to the PCI Local Bus Specification, Revision 2.0*, July 29, 2002.

[4] The Peripheral Component Interconnect Special Interest Group, *PCI Express™ Base Specification, Revision 1.0a*, April 15, 2003.

[5] ARM, Limited, *AMBA Specification, Revision 2.0*, May 1999.

[6] InfiniBand Trade Association, *InfiniBand Architecture Specification Volume 1, Release 1.0*, October 24, 2000.

[7] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the World with Wireless Sensor Networks", *Proc. of IEEE International Conference on Acoustic, Speech and Signal Processing*, Vol. 4, 2001, pp 2003-2036.

[8] G. Minassian, "Home Phone Line Networks: The Next Networking Challenge", *Electron. Prod. Design*, Nov. 1998, pp. C15-C21.

[9] IEEE Std 1149.1, *IEEE Standard Test Access Port and Boundary-Scan Architecture*, June 14, 2001.

[10] W. J. Hughes, "System Level Boundary Scan in a Highly Integrated Switch", *Proc. of International Test Conference*, 1997, pp. 636-639.

[11] IEEE Std 1149.5, *IEEE Standard for Module Test and Maintenance Bus (MTM-Bus) Protocol*, Jan 24, 1996.

[12] N. Cardoso, C. Almeida and J.C. da Silva, "A System Level Boundary Scan Controller Board for VME Applications [to CERN Experiments]", *Proc. of*

European Test Workshop, 2000, pp. 153 –158.

[13] IEEE P1500 Web Site, http://grouper.ieee.org/groups/1500/.

[14] D. Flynn, "AMBA: Enabling Reusable On-Chip Designs", *IEEE Micro*, Vol. 17, Issue 4, July/August 1997, pp.20-27.

[15] D. Flynn, "Modular Bus Support On-Chip Testability", *IEEE Colloquium: System Design for Testability*, No 1995/083.

[16] M. Nahvi and A. Ivanov, "A Network on Chip for Test Access: A Packet Switching Communication-Based Test Access Mechanism for system Chips", *Micronet Annual Workshop,* April 2002, pp. 51-52.

[17] M. Sgroi, M. Sheets, A. Mihal, K. Keutzer, S. Malik, J. Rabaey and A. Sangiovanni-Vincentelli, "Addressing the System-on-a-Chip Interconnect Woes Through Communication-Based Design", *Proc. of Design Automation Conference,* June, 2001, pp. 667-672.

[18] IEEE Std 1451.1, *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators – Network Capable Application Processor (NCAP) Information Model,* April 18, 2000.

[19] IEEE Std 1451.2, *IEEE Standard for a Smart Transducer Interface for Sensors and Actuators to Microprocessor Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats,* Sept 25, 1998.

[20] J. Burch, J. Eidson and B. Hamilton, "The Design of Distributed Measurement Systems Based on IEEE1451 Standards and Distributed Time Services", May 2000, pp. 529-534.

[21] C. Girerd, S. Gardien, J. Burch, S. Katsanevas and J. Marteau, "Ethernet Network-based DAQ and Smart Sensors for the OPERA-Long baseline Neutrino Experiment", *Proc. of Nuclear Science Symposium Conference Record,* Oct 2000, pp. 15-20.

[22] IEEE Draft P802.3ae/D4.3, *Supplement to Carrier Sense Multiple Access with Collision Detection (CSMA/CD) Access Method & Physical Layer Specification,* April 2002.

[23] National Committee for Information Technology Standardization, T11.2/Project 1316-DT/Rev 4.4, "Fibre Channel – Methodology for Jitter and Signal Quality Specification", November 12, 2001.

[24] Y. Cai, S.A. Werner, G.J. Zhnag, M.J. Olsen and R.D. Brink, "Jitter Testing for Multi-Gigabit Backplane SerDes-Techniques to Decompose and Combine

Various Types of Jitter", *Proc. of International Test Conference,* 2002, pp.700 - 709.

[25] Aligent Technologies, *81250 Product Overview DataSheet,* Palo Alto, California, 2002.

[26] Bernd Laquai, "Testing of High Speed Serial IO Interfaces Based on Spectral Jitter Decomposition", January 2003.

[27] C. Metra, S.Di Francescantonio, T.M. Mak and B. Riccò, "Evaluation of Clock Distribution Networks' Most Likely Faults and Produced Effect", *Proc. of IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems,* Oct 2001, pp. 357-365.

[28] T.J. Yamaguchi, M. Soma, J. Nissen, D. Halter, R. Raina and M. Ishida, "Testing Clock Distribution Circuits Using an Analytic Signal Method", *Proc. of International Test Conference,* 2001, pp.323 -331.

[29] Masashi Shimanouchi, "An approach to consistent jitter modeling for various jitter aspects and measurement methods", *Proc. of International Test Conference,* Oct 2001, pp. 845-857.

[30] T.J. Yamaguchi, M. Soma, M. Ishida, H. Musha and L. Malarsie, "A New Method for Testing Jitter Tolerance of SerDes Devices Using Sinusoidal Jitter", *Proc. of International Test Conference,* 2002, pp. 717 -725.

[31] Y. Fan and Z. Zilic, "Efficient FPGA Implementation of High Speed AWGN Communication Channel Emulators", *Proc. of the First Annual Northeast Workshop on Circuits and Systems,* June 2003.

[32] Altera Corporation, *Mercury Programmable Logic Device Family Data Sheet,* San Jose, California, January 2003.

[33] Altera Corporation, *CDR Mercury Device, Application Note 130,* San Jose, California, Feb. 2003.

[34] Y. Fan and Z. Zilic, "Testing for Bit Error Rate in FPGA Communication Interfaces", *Poster Presentation at 11th ACM International Symposium on FPGA,* Feb 2003.

[35] H. Zimmermann, "OSI Reference Model – The ISO Model of Architecture for Open Systems Interconnection", *Proc. of IEEE Transactions on Communications COM-28,* No.4: April 1980.

[36] M.W. Chiang and Z. Zilic, "Layered approach to designing system test interfaces", *Proc. of VLSI Test Symposium,* April 2003, pp. 331-336.

[37] S. Mitra, E.J. McCluskey and S. Makar., "Design for testability and testing of IEEE 1149.1 TAP controller", *Proc. of VLSI Test Symposium,* April 2002, pp.247 -252.

[38] Aligent Technologies, *Agilent 81200 Data Generator/Analyzer User Guide, Revision 3,* RePalo Alto, California, 2002.

[39] M. Benabdenebi, W. Maroufi, and M. Marzouki, "CAS-BUS: a scalable and reconfigurable test access mechanism for systems on a chip", *Proc. of Design Automation and Test in Europe*, March 2000, pp. 141-145.

[40] E.J. Marinissen, R. Arendsen, G. Bos, H. Dingemanse, M. Lousberg and C. Wouters, "A structured and scalable mechanism for test access to embedded reusable cores", *Proc. of International Test Conference*, Oct 1998, pp. 284-293.

[41] Maxim Integrated Products, Inc, *An Introduction to Jitter in Communication Systems,* Sunnyvale, California, 2003.

[42] LeCroy Corporation, *Serial Data Analyzers SDA Series Data Sheet,* Chestnut Ridge, New York, 2003.

[43] Wavecrest Corporation, *Understanding Jitter,* Eden Prairie, Minnesota, 2001.

[44] G. Pottie and W. Kaiser, "Wireless Integrated Network Sensors," *Communications of the ACM*, vol. 43, no.5, May 2000, p.51-58.

[45] W. Ye, J. Heidemann and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", *Proc. of International Annual Joint Conference of the IEEE Computer and Communications Societies*, June, 2002.

[46] A. Woo and David Culler, "A Transmission Control Scheme for Media Access in Sensor Networks", *Proc. of ACM/IEEE Mobicom Conference,* 2001.

[47] W. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks", *Proc. of ACM/IEEE Mobicom Conference*, August 1999.

[48] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks", *Proc. of Hawaaian Int'l Conf. on Systems Science*, January 2000.

[49] J. Broch, D.A. Maltz, D.B. Johnson, Y.C. Hu, and J. Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", *Proc. of ACM/IEEE Mobicom Conference*, Oct. 1998.

[50] K. Sohrabi, J. Gao, V. Ailawadhi, G.J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network", *IEEE Personal Communications, vol. 7*, no. 5, October 2000, pp. 16-27.

[51] IEEE Draft P802.3ae/D3.2, *Analog Target Specification Annex 48B (Jitter Test Methods),* May 2001, pp. 6-14.

[52] IEEE Std 1532, *IEEE Standard for In-System Configuration of Programmable Devices*, Dec 28, 2001.

[53] C.Y. Wan, A.T. Campbell and L.Krishnamurthy, "PSFQ: A Reliable Transport Protocol for Wireless Sensor Networks", *Proc. of the 1st ACM International Workshop on Wireless Sensor Networks and Applications,* 2002, pp. 1-11.

[54] T11.2 Technical Committee, *draft proposed NCITS Technical Report for Information Technology - Fibre Channel -Methodology for Jitter Specification Technical Report REV 10*, June 9, 1999

[55] C. Intanagonwiwat, RC. Govindan, D. Estrin, "Direct Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", *Proc. of the Sixth Annual ACM International Conference on Mobile Computing and Network,* August 2000, pp. 56-67.

[56] Texas Instrument, *MSP430X4XX Family, User Guide*, Dallas, Texas, 2003.

[57] Y. Fan, *A Versatile FPGA-based High Speed Bit Error Rate Testing Scheme, Master Thesis,* McGill University, Montreal, Aug 2003.

[58] IEEE Draft P802.15.4/D18, *Standard for Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Low Rate Wireless Personal Area Networks (LR-WPANs),* Feb, 2003.

[59] The Official ZigBee Alliance Web Site, http://www.zigbee.org/.

[60] IAR Systems, *MSP430 IAR Embedded WorkbenchTH IDE, User Guide*, Feb, 2003.

[61] Texas Instrument, *Programming a Flash-Based MSP430 Using the JTAG Interface, Application Report,* Dallas, Texas, 2002.