XAPP537 (v1.1) November 29, 2004

MultiBERT IP Toolkit for Serial Backplane Signal Integrity Validation

Author: Delfin Rodillas

Summary

For many years, backplanes have been used to physically and electrically interconnect components of complex systems. Standard backplanes such as VMEbus and CompactPCI have supported integration of system components from different vendors. The electrical interconnect has traditionally taken the form of parallel buses, but the proliferation of high-speed serial technology is changing this practice.

Today's serial backplane implementations support line rates ranging from 622 Mbps to 3.125 Gbps and are now approaching speeds in excess of 10 Gbps. A significant recent development is the emergence of standards to define serial backplanes. The Advanced Telecom Computing Architecture (ATCA) specification is one example that has gained industry traction. Whether proprietary or standards-based, serial backplanes present a very demanding signaling environment with high signal density, multiple connectors, and substantial trace lengths. Proving and characterizing the performance of any high-speed serial solution is critical, and MultiBERT provides a means of accomplishing this with Xilinx Multi-Gigabit Transceivers (MGTs).

Introduction

The fundamental nature of MultiBERT is the simultaneous operation of a large number of serial connection tests over an ATCA backplane. MultiBERT is based on the Xilinx ATCA 10G 4X4 Line Card (hereafter called “ATCA Card”) and, as illustrated in Figure 1, requires management and reporting at a central host computer. The MultiBERT GUI runs on the host computer and supports control of the serial test parameters. It also reports test statistics on a cumulative and per-connection basis. It should be noted that while MultiBERT was developed for an ATCA environment, the same concepts can be used to validate proprietary serial backplane implementations.

NOTICE OF DISCLAIMER: Xilinx is providing this design, code, or information "as is." By providing the design, code, or information as one possible implementation of this feature, application, or standard, Xilinx makes no representation that this implementation is free from any claims of infringement. You are responsible for obtaining any rights you may require for your implementation. Xilinx expressly disclaims any warranty whatsoever with respect to the adequacy of the implementation, including but not limited to any warranties or representations that this implementation is free from claims of infringement and any implied warranties of merchantability or fitness for a particular purpose.

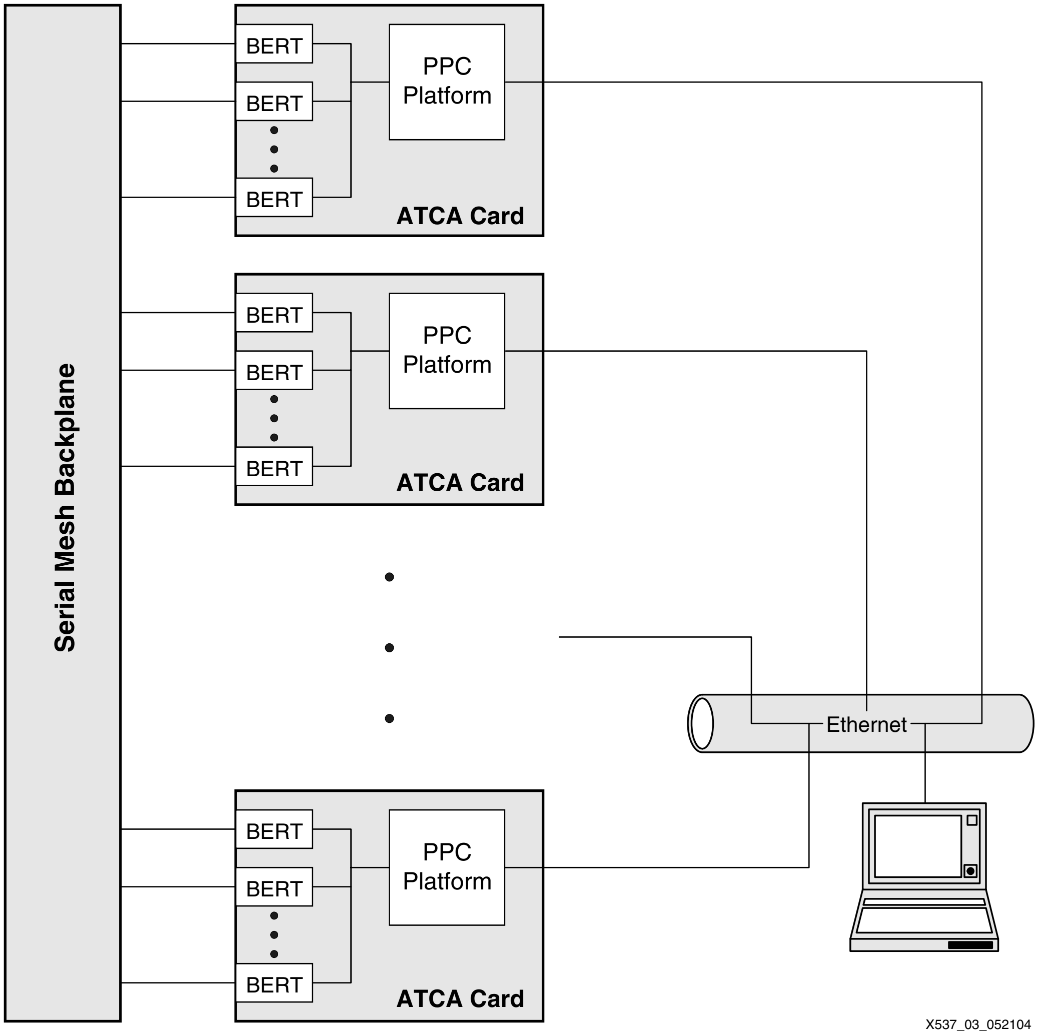


Figure 1: **MultiBERT System**

Terminology

Certain terms are used with a specific meaning in this document. These terms are defined in Table 1, and acronyms are provided in Table 2.

Table 1: **Definition of Terms**

Term	Definition
8B/10B Encoding	A standard encoding technique in which 8-bit characters are mapped to 10-bit codes to provide a transition-rich, DC-balanced signal.
IOCTL	A general purpose Linux system call used for implementing the interface between a user application and a device driver.
RAMdisk	A minimal Linux file system implemented in RAM.

Table 2: **Acronym Definitions**

Acronym	Definition
ATCA	Advanced Telecom Computing Architecture
BERT	Bit Error Rate Test (Tester, Testing)
FIFO	First-In First-Out (Memory)
GPIO	General Purpose I/O
GUI	Graphical User Interface
MGT	Multi-Gigabit Transceiver
PRBS	Pseudo-Random Binary Sequence
UART	Universal Asynchronous Receiver-Transmitter

Features

The following list outlines the high-level features of MultiBERT:

- Up to 128 simultaneous full duplex serial connections, in a 16-slot full mesh chassis.
- Automatic slot identification and backplane modeling, with provision for user-defined backplane models.
- Control of the following settings through the MultiBERT GUI, running on Windows:
  - Eight different serial test data patterns
  - Two different clock settings for each reference clock
  - Use of MGT serial and parallel loopback
  - MGT transmit polarity
- Control of the following settings through FPGA compile-time configuration:
  - MGT Transmit Pre-Emphasis (TX_PREEMPHASIS)
  - MGT Transmit Differential Voltage (TX_DIFF_CTRL)
  - MGT Reference Clock Source (REF_CLK_VSEL):
    - REFCLK (Up to 2.0 Gbps)

  - BREFCLK (Up to 3.125 Gbps[1])
- Reporting of the following statistics through the MultiBERT GUI:
  - Aggregate data transferred and error count
  - Per-connection data rate and error rate
- Aggressor Mode, in which some channels are configured to drive fixed patterns for the purpose of creating a harsh electrical environment for ongoing tests.

Architecture

The MultiBERT system consists of the following components:

- ATCA Backplane
- ATCA 10G 4X4 Line Card
- OSCAR & OSCAR Device Driver
- MultiBERT Platform & Linux
- Embedded MultiBERT Application / Network Server
- MultiBERT GUI

ATCA Backplane

The ATCA specification defines a number of standard backplanes, including the following:

- Full mesh up to 16 slots
- Replicated mesh for 8 or fewer slots
- Star and dual-star configurations

In each case, connections between individual cards consist of from one (1X) to four (4X) bi-directional serial links. The relevant characteristic that must be taken into account in MultiBERT is the specific routing between slots. This must be modeled in order to identify the specific MGTs that constitute the endpoints of each connection.

ATCA 10G 4X4 Line Card

The ATCA Card is essentially a reference design supporting all basic aspects of ATCA, including the high-speed serial interfaces and management functions. The central component is a Virtex-II Pro™ device (XC2VP50 or XC2VP70) with 16 of its MGTs connected to the backplane. This provides connection to four ports (4X ATCA) on four channels.

The ATCA Card also supports a 10/100 Ethernet interface, a System ACE™ Compact Flash interface, and a configuration selection switch. There are both 125 MHz and 156.25 MHz reference clocks available for the MGTs (2.5 Gbps and 3.125 Gbps[2]).

MultiBERT Platform and Linux

The MultiBERT platform implemented in the Virtex-II Pro device is shown in Figure 2. The primary features are an Ethernet interface, GPIO to read the ATCA Hardware Address for slot identification, and a serial test module described in the next section. A UART is also available for debugging.

[1] The ATCA 4x4 card is rated at 2.5Gbps. Operation of the card at 3.125G may result in errors for particular slot configurations. Please consult the RocketIO Transceiver User Guide at http://www.xilinx.com/bvdocs/userguides/ug024.pdf for more information on higher speed designs up to 3.125 Gbps.

[2] See footnote 1.

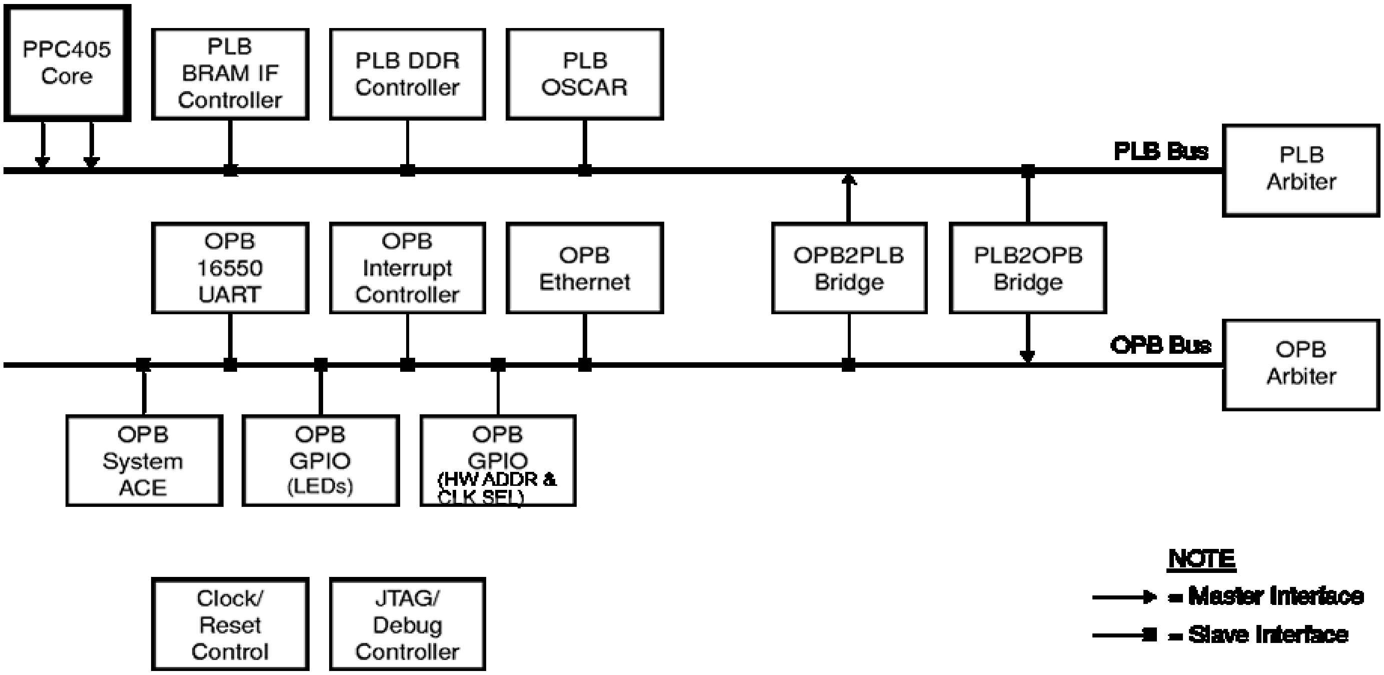


Figure 2: **MultiBERT Platform**

The MultiBERT platform is implemented in EDK 6.2 with the resource utilization defined in Table 3. The 10/100 Ethernet MAC core used in the MultiBERT reference design is the evaluation version included with EDK. This evaluation version will stop operating after six to eight hours of run time. In order to run extended tests, you must purchase and install the full 10/100 Ethernet MAC core.

Table 3: **MultiBERT Platform Resource Utilization**

BRAM	LUT	DFF	DCM	GCLK	MGT	I/O
36	12000	11000	2	7	16	137

As shown in Figure 3, the embedded software running on the MultiBERT platform is based on Linux, specifically Monta Vista 2.4.18. A single System ACE image supports FPGA configuration and Linux kernel boot with an initial RAMDisk. If a user does not log on to Linux, the MultiBERT Server application will start automatically. During a Linux start-up, the IP address and Ethernet MAC address are derived from the slot number, as determined by reading the ATCA Hardware Address pins.

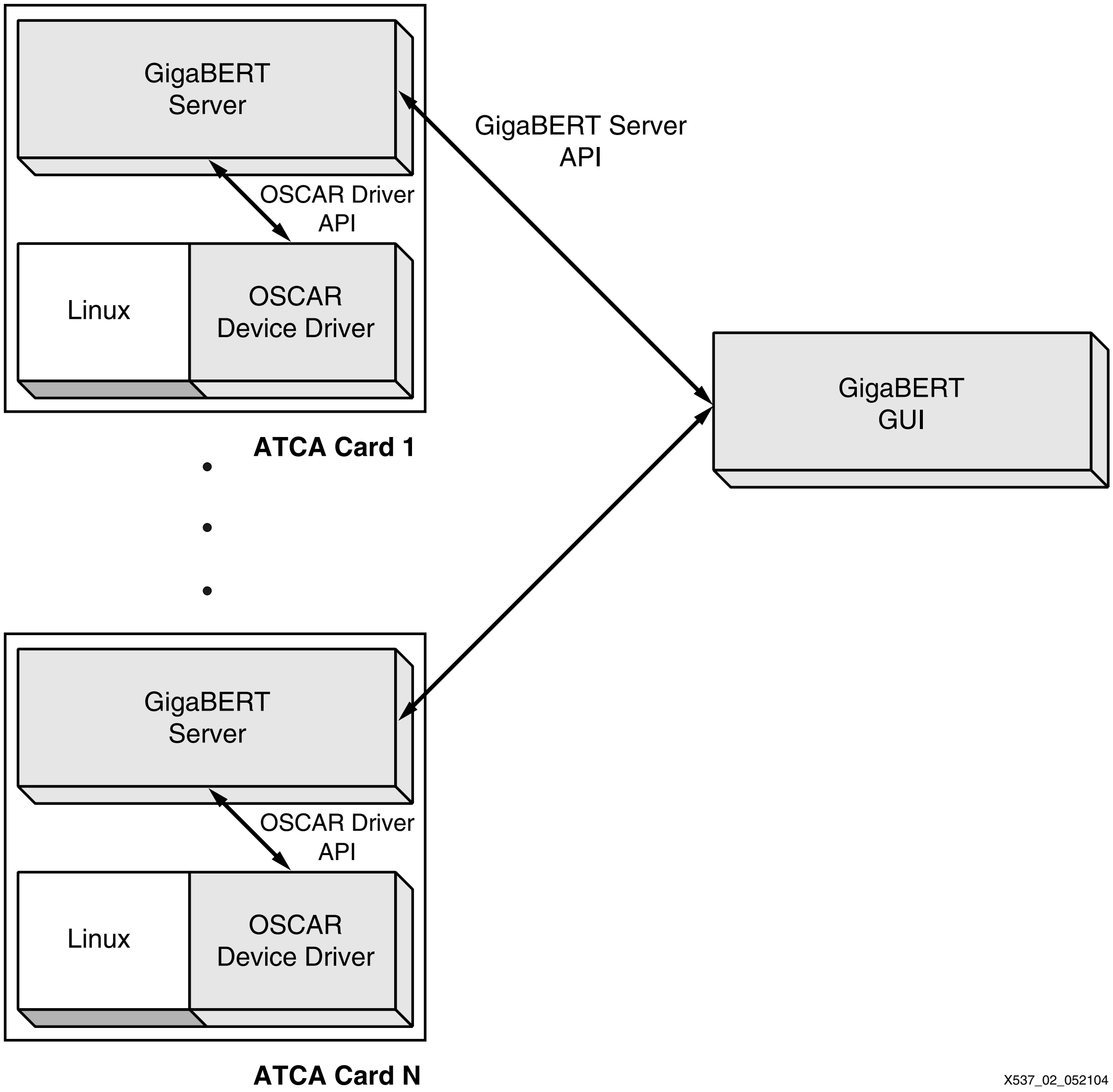


Figure 3: **MultiBERT Software**

OSCAR and Device Driver

OSCAR is a PLB peripheral that performs serial transmission and testing on up to 16 MGTs. (Strictly speaking, OSCAR is not a BERT since it uses 8B/10B encoding. This prevents determination of the exact number of bit errors.) A block diagram of OSCAR is illustrated in Figure 4.

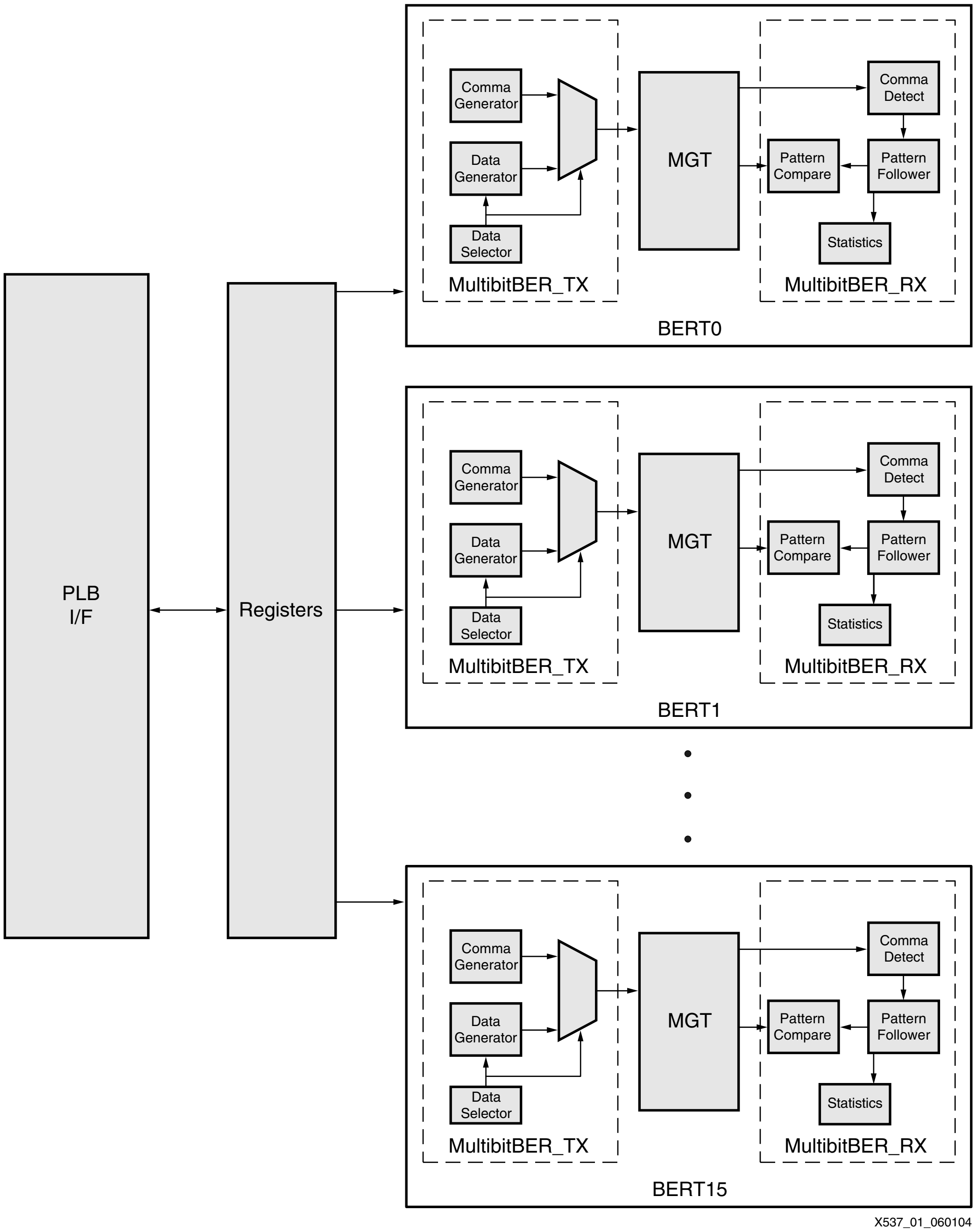


Figure 4: **OSCAR Block Diagram**

OSCAR supports the following settings through register programming:

- Serial test data patterns, all of which use 8B/10B encoding:
    - 8-bit increment
    - 8-bit decrement
    - 16-bit increment
    - 16-bit decrement
    - All zeroes
    - All ones
    - Alternating bytes of zeroes and ones (…00FF00FF…)
    - 16-bit PRBS (2^7-1 Polynomial)
- Per-channel enabling of Aggressor Mode, in which received data is not checked and transmitted data is a fixed un-encoded pattern of 20 logic '1' bits alternating with 20 logic '0' bits
- Choice of two different clock speeds for each reference clock:
    - REFCLK (62.5 MHz/ 1.25 Gbps or 100 MHz/ 2.0 Gbps)
    - BREFCLK (125 MHz/ 2.5 Gbps or 156.25 MHz/ 3.125 Gbps[1])
- Use of MGT internal loopback (serial or parallel)
- Inversion of receive and transmit polarity
- Transmit inhibit
- MGT power down

OSCAR also makes the various MGT status signals available through register access, indicating disparity errors, 8B/10B decode errors, etc. In addition, the following statistics are maintained:

- Total frames received (40-bit)
- Erroneous frames received (32-bit)
- MGT errors (32-bit)

Erroneous frames are those which are caused by a bit error in the channel, such as an incorrect received character, an 8B/10B decoding error, or a running disparity error. MGT errors are of a more systemic nature, such as receive FIFO overflow or a loss of clock synchronization.

OSCAR has a Linux device driver which uses IOCTL calls to support the following commands:

- GET_CHANNEL_STATUS
- START_TEST
- GET_STATISTICS
- END_TEST

The device driver also maintains 64-bit frame counts, and ensures that OSCAR registers are read sufficiently often to avoid hardware overflow.

[1] The ATCA 4x4 card is rated at 2.5Gbps. Operation of the card at 3.125G may result in errors for particular slot configurations. Please consult the RocketIO Transceiver User Guide at http://www.xilinx.com/bvdocs/userguides/ug024.pdf for more information on higher speed designs up to 3.125 Gbps.

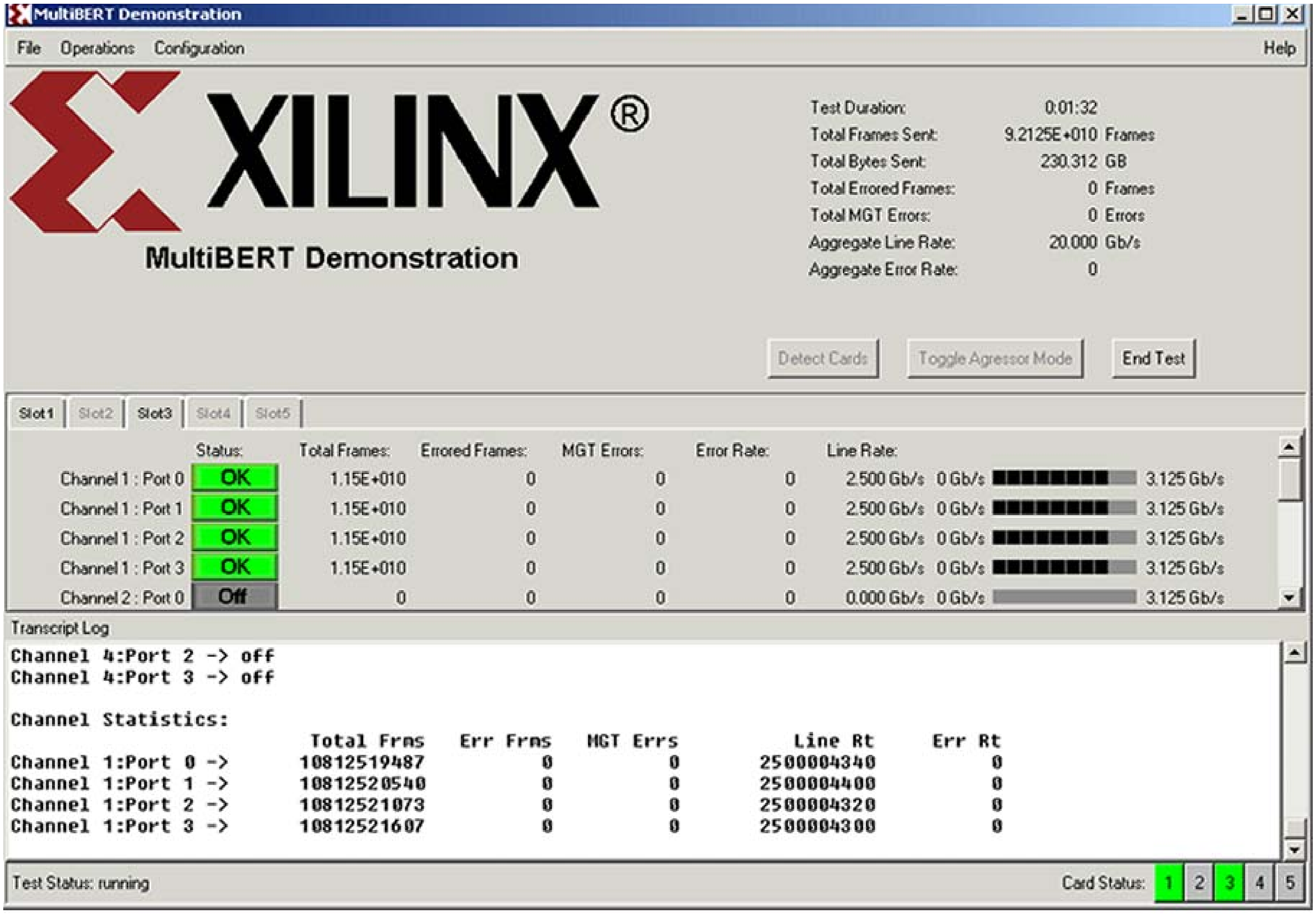
MultiBERT Server

The MultiBERT Server is an application running under Linux on the MultiBERT platform. The MultiBERT Server primarily responds to requests from the MultiBERT GUI via Ethernet, as follows:

- CARD_QUERY
- CHANNEL_STATUS_QUERY
- START_TEST
- REPORT_STATISTICS
- END_TEST

MultiBERT GUI

The MultiBERT GUI provides a simple graphical method for controlling tests and reporting results. It is implemented with Perl/TK, and runs under Windows 2000 and Windows XP. The main MultiBERT GUI screen is shown in Figure 5, and it uses tabs to allow viewing of all individual connections.



Test Duration: 0:01:32
Total Frames Sent: 9.2125E+010 Frames
Total Bytes Sent: 230.312 GB
Total Errored Frames: 0 Frames
Total MGT Errors: 0 Errors
Aggregate Line Rate: 20.000 Gb/s
Aggregate Error Rate: 0

	Status:	Total Frames:	Errored Frames:	MGT Errors:	Error Rate:	Line Rate:
Channel 1 : Port 0	OK	1.15E+010	0	0	0	2.500 Gb/s 0 Gb/s 3.125 Gb/s
Channel 1 : Port 1	OK	1.15E+010	0	0	0	2.500 Gb/s 0 Gb/s 3.125 Gb/s
Channel 1 : Port 2	OK	1.15E+010	0	0	0	2.500 Gb/s 0 Gb/s 3.125 Gb/s
Channel 1 : Port 3	OK	1.15E+010	0	0	0	2.500 Gb/s 0 Gb/s 3.125 Gb/s
Channel 2 : Port 0	Off	0	0	0	0	0.000 Gb/s 0 Gb/s 3.125 Gb/s

```
Channel 4:Port 2 -> off
Channel 4:Port 3 -> off

Channel Statistics:
                        Total Frms   Err Frms   MGT Errs          Line Rt     Err Rt
Channel 1:Port 0 ->    10812519487          0          0       2500004340          0
Channel 1:Port 1 ->    10812520540          0          0       2500004400          0
Channel 1:Port 2 ->    10812521073          0          0       2500004320          0
Channel 1:Port 3 ->    10812521607          0          0       2500004300          0
```

Test Status: running

Figure 5: **MultiBERT GUI Main Screen**

The following backplane models are pre-defined and can be selected in the appropriate pull-down menu:

- Full Mesh

- 4-Slot Replicated Mesh
- 5-Slot Replicated Mesh

In addition, the MultiBERT GUI automatically looks for a file called "backplane.inc" in the current working directory and attempts to parse the contents as a Perl table defining a backplane model. The table format is analogous to the tables included in Section 6.6.3 of the ATCA specification [1][2]. You can also browse to locate other backplane definition files in the same format.

Enabling Aggressor Mode causes all inactive channels to drive the aggressor pattern instead of being switched off. In a typical aggressor mode scenario, only a few channels will actually be running tests and all other channels will be inactive. This scenario is created using a virtual backplane model, with only the test channels activated. (Begin with the complete model for the backplane in use and deactivate all of the channels that should be operating in Aggressor Mode, leaving only the test channels in place.)

Usage Guidelines

The menu options available under the MultiBERT GUI are summarized in Table 4. You can perform a basic test scenario as follows:

1. Install ATCA cards in the chassis.
2. Connect all ATCA cards to the host computer through an Ethernet switch.
3. Power on the chassis.
4. Start the MultiBERT GUI.
5. Perform the “Detect Cards” operation.
6. Set test options as desired.
7. Start logging, if required.
8. Start the test.
9. Stop the test after running for the desired duration.
10. Save the logged results, if required.

An Aggressor Mode scenario follows the same flow, except that a test-specific backplane model must be selected and Aggressor Mode enabled. Aggressor Mode is enabled using a button on the main GUI screen, as opposed to a menu option.

Table 4: **MultiBERT GUI Menu Options**

Top-level Menu	Options	Description
File	Various file operations	Test transcript, including final statistics, can be saved to a file
Operations	Detect Cards	Determine which cards are in which ATCA slots
	Start/Stop Test	Start or stop a test sequence
Configuration	IP Addresses	Manually enable/disable specific cards
	Test Options	Set test patterns, clock sources, etc.
	Backplane Interconnect	Select from the available backplane models
	Load Backplane Model	Browse to a file to load new backplane models

MultiBERT uses the ATCA Card front panel LEDs as follows:

- RED: MultiBERT Server failed to start up properly
- YELLOW: At least one serial test error has been detected
- GREEN: MultiBERT Server successfully started up
- BLUE: Board has successfully started up

Download Information

MultiBERT can be downloaded for free from:
http://www.xilinx.com/esp/networks_telecom/optical/xlnx_net/multibert.htm
The reference design consists of three components:

1. MultiBERT Platform EDK Project

2. MultiBERT Embedded Software, including device driver and MultiBERT Server application. The Linux kernel is implemented as a patch that users must apply after downloading the ML300 Monta Vista Preview Kit (Version 3.0)

3. MultiBERT GUI, in the form of Perl source code. Users must download the Active Perl distribution from http://www.activestate.com

The reference design also includes an example "backplane.inc" file, with a number of backplane models including some appropriate for Aggressor Mode usage.

References

1. AdvancedTCA Base Specification (PICMG 3.0 Revision 1.0, December 30, 2002)
2. AdvancedTCA Base Specification Engineering Change Request (ECR 3.0-1.0-001, January 12, 2004)

Customization

For information on MultiBERT customization, please contact AMIRIX Systems:

AMIRIX Systems Inc.

77 Chain Lake Drive

Halifax, NS, Canada B3S 1E1

www.amirix.com

Tel: (902) 450-1700

Toll Free: 1-866-426-4749

Fax: (902) 450-1704

E-mail: info@amirix.com

Revision History

The following table shows the revision history of this document.

Date	Version	Revision
05/28/04	1.0	Initial Xilinx release.
11/29/04	1.1	Added Aggressor Mode and ported to 10G 4X4 Line Card.