LabStation: A Low Cost, Scalable, High Throughput Characterization Environment

Pradeep Batra, Adrian Torres, William Ng, Arun Vaidyanath, Philip Yeung, Tsunwai Gary Yip
Rambus Inc. Sunnyvale, California USA
www.rambus.com

***Abstract* — This paper presents a characterization environment that can reduce the time-to-market of IO cell technologies, such as DDR2, DDR3, XDR™, Mobile XDR™, PCI Express and various standard SERDES. The environment offers a convenient tool set for the bring-up and validation of a new test chip. The system-like environment is also used for determining the performance of the test chip in a system similar to the target product. As such the environment is useful for debugging when problems arise during the production phase. Low cost, scalability, and ease of deployment are some other attractive features of the characterization environment.**

I. Introduction

Integrating cells from libraries of IO technology into an ASIC design flow is one way to shorten the time-to-market. Successful integration of the IPs, such as XDR, FlexIO™ [1], DDR3, and SERDES PHYs, requires the resulting design to meet specifications not only at the chip level, but also at the system level. This paper describes the hardware and software architecture of the LabStation™ Characterization Environment used to test an IO cell technology in a system under test. The environment is known for its fully automated high throughput testing capability [2]. The portability of the LabStation software and the flexibility of its hardware have successfully supported the development of various high bandwidth memory subsystems for game consoles [3] and consumer products such as digital TVs and projectors.

In many cases the system under test, by design, has most or all of the system features of the final product. Consequently engineers can continue to use the environment for debugging when problems arise in the production phase. The modular nature of LabStation software and hardware allows easy transport. Hence engineering teams at multiple geographical locations can easily work together using duplicate set ups.

The scope of this paper includes

1. an overview of the LabStation environment,
2. architecture of LabStation software,
3. architecture of the interfaces for controlling the test chip and system under test,
4. an introduction of the language for test scripts, and
5. the design approaches that make the test and power platforms reusable in testing different IO subsystems.

Benefits of the high throughput testing capability will be illustrated in the presentation of characterization results over a wide range of process-voltage-temperature conditions.

II. Overview of Characterization Environment

Figure 1 shows the key hardware and software elements in the LabStation Characterization Environment. From the GUI of the LabStation software, commands for control and testing are initiated and sent through the USB port of a computer to a command translator called the LabStation Interface Module (LIM). The LIM outputs are compliant with I2C, SPI, JTAG, and customer-specific protocols to support a wide range of components on the test board. The serial port, GPIB, and Ethernet are also used to control equipment during testing. A test chip can be maintained at any desired temperature between -40C and +125C without putting the test platform in a temperature chamber [2].

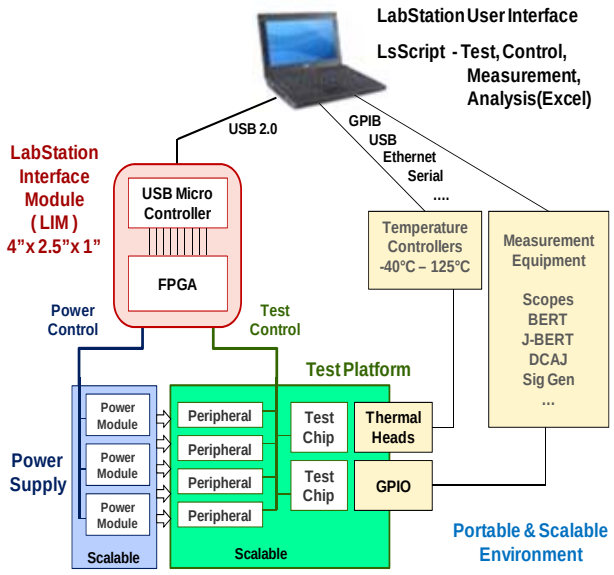

Figure 1: LabStation Characterization Environment.

The architecture of the environment is designed to control up to 255 devices on a test platform, thus allowing the complexity of the test board to scale with the development and production of the silicon and system. The architecture is useful for silicon bring-up, characterization, and debugging. In addition to the test chip silicon, its package design and the PCB design are directly transferable to production.

III. LabStation Software

The user environment is constructed from three software elements: the LabStation application, configuration file of the

test board, and a test script written in a language called LsScript.

The contents of the LabStation application are illustrated in Figure 2. LabStation has information about different test systems and test chips supported by the characterization environment. Examples include test systems and test chips for XDR, DDR3 and PCI Express technologies. Each test board typically has an EEPROM holding the hardware configuration of the board. The characterization environment is initialized at the startup of the LabStation application. LabStation utilizes the LIM to access the configuration file stored in the EEPROM of the test board. According to the identifiers found in the configuration file, LabStation instantiates the appropriate test features which are displayed in the LabStation GUI. During automated testing, the test features are accessible by commands written in LsScript.

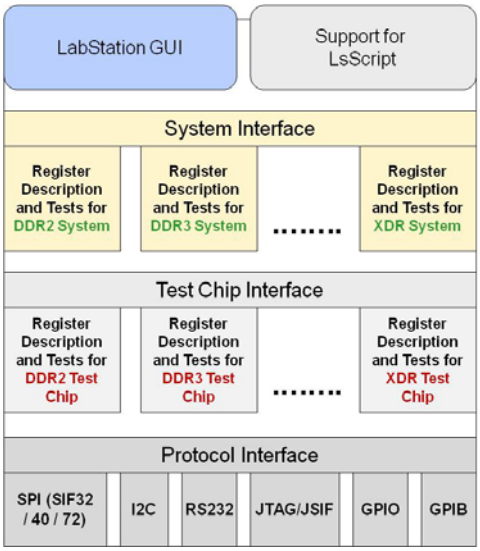

Figure 2: Software architecture of LabStation application file.

In addition to specifying the system and test chip for the technology to be characterized, the configuration file also contains identifiers for peripheral devices on the test board and power supply board that support the testing. A number of protocols, as shown in Figure 2, are supported so that LabStation can transmit commands and data to different types of peripheral devices. One example is to use the I2C interface to set up a CK505 compliant clock chip to output an 800 MHz clock for testing DDR3 memory at 1.6 Gbps. Table 1 lists some of the components supported by LabStation via standard interfaces like I2C, SPI, GPIO, GPIB, and JTAG. Their functions include, but are not limited to, sensing, control, and signal generation.

Besides the standard protocols like I2C and SPI, LabStation also supports non-standard protocols. An example is the JSIF, which is a serial interface protocol for communication with Rambus test chips over JTAG.

IV. Advantages of Automated Initialization

The above description of the process for creating a characterization environment is transparent to the user and does not require any user input. The test features are available in the LabStation GUI and also accessible during automated testing using commands written in LsScript.

By the use of identifiers in a configuration file, the LabStation Characterization Environment is easily portable among test platforms for various IO cell technologies. The environment has been used successfully to characterize PCI Express, XDR, DDR3, Mobile XDR, and other IO cell technology.

Additionally, consistency in the test environment is maintained across different test platforms, thus minimizing the need for re-development. The LabStation application is continuously updated to include new test platforms and test chips. The architecture of the software is scalable to support different levels of hardware complexity. The LabStation software is written in C++ and has a layered architecture which makes it easy to add support for new protocols, hardware devices, and hardware interfaces.

ADT7301	Temp Sensor	Analog Devices	MAX1153	Temp Sensor	MAXIM
AD5231	Digital Pot	Analog Devices	MAX1240	ADC	MAXIM
AD5235	Digital Pot	Analog Devices	MAX1286	ADC	MAXIM
Ad7416-18	Temp Sensor	Analog Devices	MAX4691	MUX	MAXIM
AD7814	Temp Sensor	Analog Devices	MAX5122	DAC	MAXIM
ADS7817	ADC	Texas Instrument	MAX5171	DAC	MAXIM
ADT7461	Temp Sensor	Analog Devices	MAX6642	Temp Sensor	MAXIM
ADT7467	Temp Sensor	Analog Devices	MAX6957	LED Display	MAXIM
AT34C02B	EEPROM	ATMEL	MCP23008	IO Expander	MicroChip
AT25128	EEPROM	ATMEL	MCP23016	IO Expander	MicroChip
BR24L04-W	EEPROM	ROHM	MK1493	Clock Generator	ICS
CDCDLP223	Clock Synthesier	Texas Instrument	MPC9230	Clock Synthesier	IDT
CDCE706	Clock Synthesier	Texas Instrument	PCA9540	I2c Mux	Philips
CY22150	Clock Generator	Cypress	PCA9546A	I2c Switch	Philips
GLK12232	Matrix Display	Matrix Orbital	PCA9547	I2c Mux	Philips
PCA9548	I2c Switch	Philips	PCA9556	I2c GPIO	Philips
ICS9FG104	Clock Generator	ICS	RPS-10	Remote Power Swit	WTI
ICS9249	XDR clock Generato	ICS	RTC-8564	Real Timme Clock	EPSON
ICS954309	Timing control hub	ICS	S75100	Temp Controller	TTC
ICS954321	Timing control hub	ICS	SLG505YC264	Clock Synthesier	SILEGO
MA24AA64	EEPROM	MicroChip	SLG63103	Clock Generator	SILEGO

Table 1: Examples of components supported by LabStation Characterization Environment.

V. LabStation Interface Module

The LabStation Interface Module (LIM) is a smart device physically connected to a USB port of the PC on one end, and to the test platform and power supply board on the other (see Figure 1). It is designed to translate commands sent from LabStation into protocols understandable by the intended receiving devices on the test platform and power supply board. Data can also be transmitted back to LabStation via the module.

The LabStation software talks to the LIM using a custom packet protocol optimized for high throughput. This protocol takes advantage of the burst mode of the USB interface by sending a large number of concatenated commands in one write operation.

There are multiple protocol engines in the LIM to translate information in the packets into commands compliant with different interface standards. For example, the LIM decodes the custom packets and translates them to SPI commands to control the DAC's in the power supplies. The JSIF engine outputs commands for programming the registers in the test chip.

VI. Other Interfaces

LabStation supports two other interfaces commonly used in testing. They are the RS232 port and GPIB bus. At startup LabStation looks for any instruments connected to the GPIB bus and automatically populates the GUI with the controls for the instruments. It also looks at the RS232 port for temperature controllers. Once found, the devices are added to the GUI without any user intervention.

VII. LabStation GUI

The LabStation GUI as shown in Figure 3 consists of four panels. The "Catalog" panel at the upper left displays the hierarchical organization of the registers and commands available in the test environment. To the right is the "Details" panel for navigating and executing selected commands and operations. Frequently used commands and operations can be stored in the "Command Buffer" in the middle for repeating use. The bottom part of the GUI is the action log which contains a line by line record of the commands and operations executed since the launch of the GUI. This record is in the format of the LsScript language and therefore can be conveniently converted to a test script.

VIII. LsScript

LsScript is the scripting language built into LabStation. The language syntax is based on register access. Register names are specified in the same hierarchical way as specified in the architectural specification of the device in the system under test. Subfield access for the registers is supported. Polling register subfields with specified time out is allowed to make scripts easy to write.

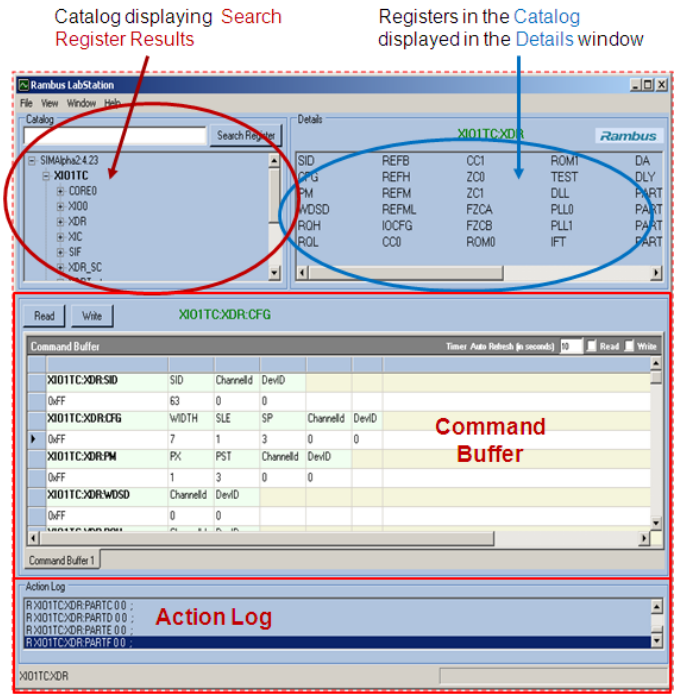

Figure 3: Different interface panels in a LabStation GUI.

LsScript uses a C like syntax and supports common features like logic and math operators, for loops, variables, and procedures. The language has built in high level variable types for statistical calculations, user input parsing, process control, and string processing. Figure 4 shows a sample LsScript highlighting various language features. LabStation has a built in source level LsScript debugger, which allows for easy debugging by the use of breakpoints, single stepping, and reading/writing LsScript variables.

For the protection of the IP and algorithms in the scripts, LsScripts can be encrypted for distribution to OEM's and manufacturers for production testing. Encrypted LsScripts can also have an expiration date.

```
; script to run a test across vdd of 900mv to 1250mv and measuring current

local QW qw_vdd_val 900 ;
for($qw_vdd_val = 900 ; ($qw_vdd_val <= 1250) ; $qw_vdd_val = ($qw_vdd_val + 50))

{
 W ECMB:vdd:voltage ($qw_vdd_val)                ; change vdd voltage
 wait 100                                        ; wait 100 ms to let voltage settle
 run testchip:noisetest:start call               ; run the be noise test
 print ELFCON "Current = " %( ECMB:Vdd:Current ) \n ; print the current in log file
                                                    and on console
 run testchip:noisetest:stop call                ; stop the noise test
}
```

Figure 4: Sample test script highlighting LsScript features.

IX. Design of Test and Power Platforms

Figure 5 is an illustration of the different sections of a test board for characterizing memory interfaces. The system under test includes the memory PHY in a controller test chip and the interfaces in two simulated DRAM devices.

Like all test boards used in the LabStation environment, the board in Figure 5 consists of four sections, which are the

1. system under test (gold),
2. power supplies (yellow),
3. control signals (green lines), and
4. peripherals (white), such as clock devices.

The system under test is unique to the IO cell technology being characterized. There are two approaches used in designing the system. One approach is to emulate the constraints of a production system such as layer counts, package and PCB design rules, and PCB layout area of the system. Such a design can be transferred to production directly if it meets system specifications. The characterization results from the development phase become a knowledge base for debugging when a problem arises during production. Figure 6 shows the system under test that is transferable to a product.

For the development of new IO cell technology, e.g. a 16 Gbps data link for building a 1 TBps memory subsystem [4], a different design approach is used to create a system suitable for characterizing the performance limits of the technology. The design constraints are often very different from those of a production system.

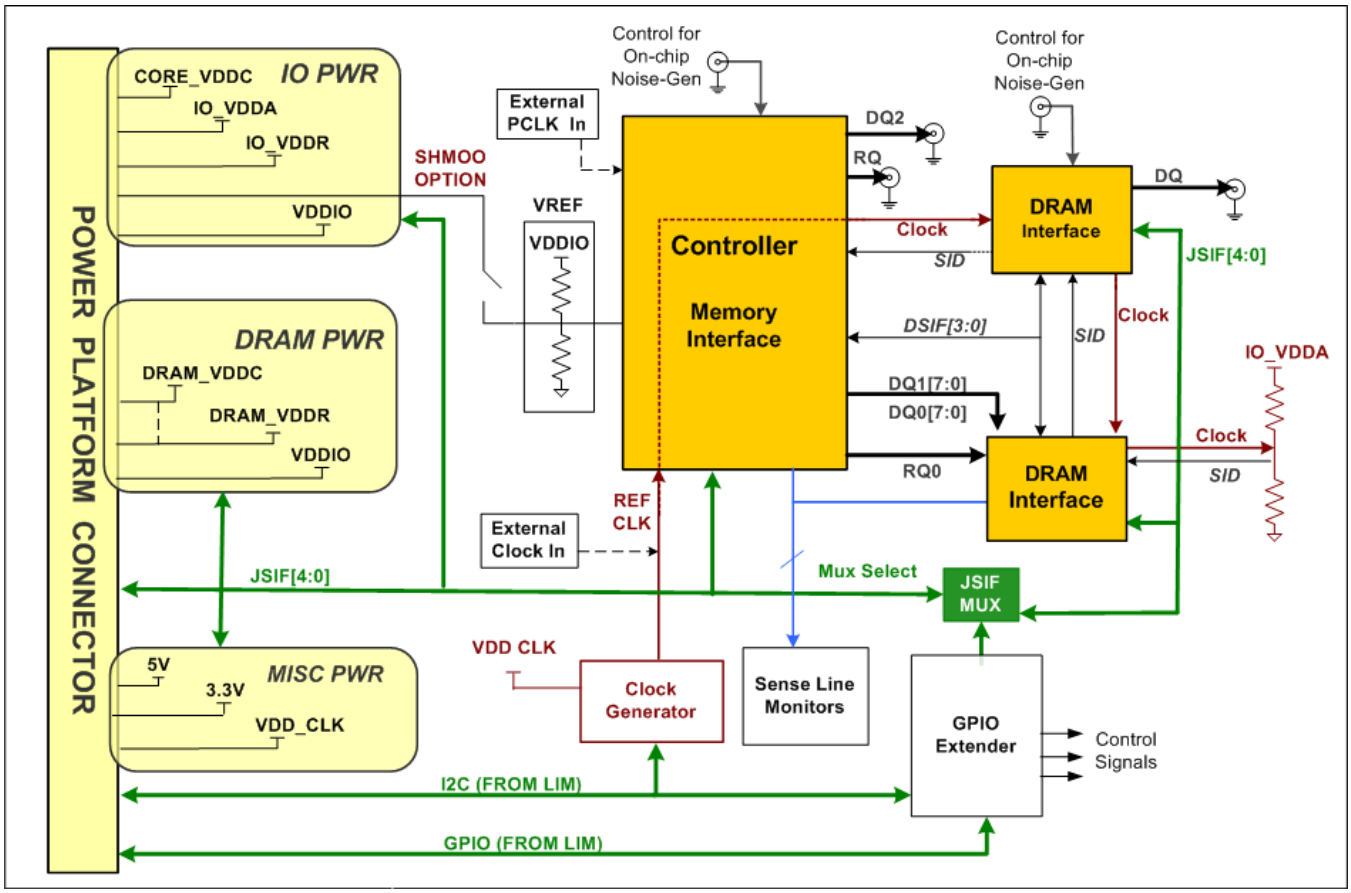

Figure 5: Four parts of a test board: system under test in gold, power in yellow, control in green and peripherals in white.

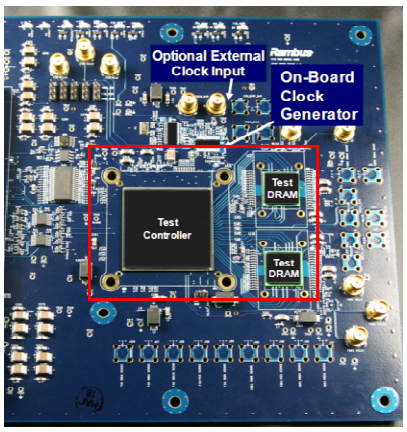

Figure 6: A test board with a system under test (area in red box) that is transferable to a product.

The power supply is reusable and scalable because of its modular nature. On a separate power platform attached to the test board are the power modules that generate different voltage rails. The power platform can support up to twelve modules. The output voltage of each module is programmable and can support different current capacities.

The power platform is configurable and able to support different test boards. Additionally it reduces the complexity of the test board to solely distributing power planes as required by the system under test and peripherals. Figure 7 shows the power platform used in the characterization of a multi-protocol IO cell designed for data links ranging from 1.0 to 6.375 Gbps. More about the features and design of the power platform is described in a companion paper.

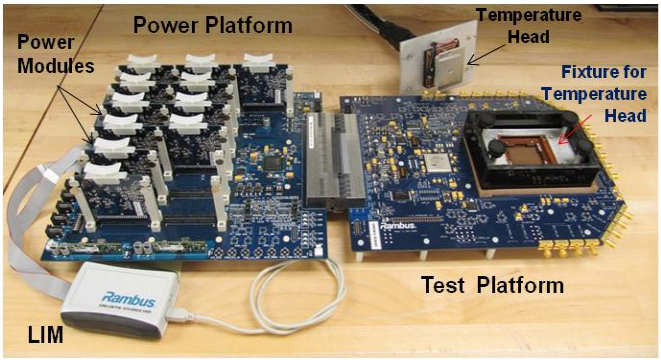

Figure 7: Power and test platforms, LIM and temperature head.

The control signals (in green) in Figure 5 are generated by the protocol engines in the LIM according to the commands from the computer running the LabStation application. The JTAG compliant serial interface, JSIF, uses a custom protocol to transmit WRITE and READ commands to the registers in the test chips. One type of WRITE command is for configuring the test chip; an example is to set the loop

bandwidth of a PLL. Another type is for initiating a test. An example is to instruct the state machine in the XDR memory subsystem to determine the optimal transmit and receive timing for a data link. The READ commands retrieve from the test chip information that defines the operating conditions and data collected during characterization. In addition to JSIF, LabStation uses standard control interfaces such as I2C, SPI, JTAG, and GPIO to configure the peripheral components.

The peripherals in Figure 5 are in white. They are devices supporting the system under test. LabStation supports many different kinds of devices. Some of them are shown in Table 1. They are common for most of the test boards, but often configured differently to meet the requirements of different systems under test. An example of configurability includes the frequency, swing and DC voltages of a reference clock signal for a memory subsystem under test.

A new IO cell technology often requires new JSIF commands for communication with the test chip. There is no need for redesign of the control interface because the JSIF engine in the LIM can be updated to incorporate the new requirements. Furthermore, when a test platform uses a new peripheral device, the LabStation application is updated to include its specifications without re-development of the software. These two types of updates allow LabStation to instantiate an environment that can support the characterization of the new technology without user intervention.

X. Characterization Results

The fully automated LabStation environment allows very high throughput in collecting data across a wide range of voltage and temperature conditions. In addition, register settings in the test chip can be changed to test against the specifications of the components and system under test. This section presents test data from the characterization of a XDR memory interface and a wide bandwidth serial link.

A. XDR Memory Interface

When designing a XDR memory interface according to a particular silicon process and rules, provisions are made for tuning the circuits to meet system specifications. Some clock alignment register settings are available for tuning the performance of each data pin in a 32-bit wide interface. The interface was tested for 22 clock alignment settings over the process, voltage and temperature shown in Table 2a. A total of 107,008 tests (Table 2b) were performed to find the optimal setting for read and write operations.

While sweeping through a large number of conditions and register setting is very time consuming, with automation, few resources are required to do this testing. Using only one low cost test platform, an operator changes the test chip once a day. The 107,000 tests were completed in four days.

One advantage of high throughput characterization is to allow visualization of the conditions for passing and failing using color coding on the test results. Figure 8 summarizes the results of 39,424 tests, in which the 32-bit interface was performing READ and WRITE operations for 22 settings of clock alignment register and 28 PVT conditions. With green for passing and red for failing, the register settings closest to the left edge of the graph are the optimal. Settings in middle of the graph will cause failure of the majority of data pins in the 32-bit interface.

Another advantage of high throughput is that the entire behavior of a system often can be discovered when a large volume of data covering the full range of specifications are available.

Process Splits	FF, FS, SF, SS
Temperature	0C – 95C
VDDR	0.900V – 1.100V
VDDA	1.455V – 1.545V
VTT	1.140V – 1.260V
Operations	READ, WRITE

(a)

Number of Devices (Processes)	4
Voltage-Temperature Combination	19
Data Pins per Device	32
Register Settings	22
Operations per Data Pin	2
Total Number of Tests	*107,008*

(b)

Table 2: (a) Test conditions, and (b) 107,008 tests on the XDR interface.

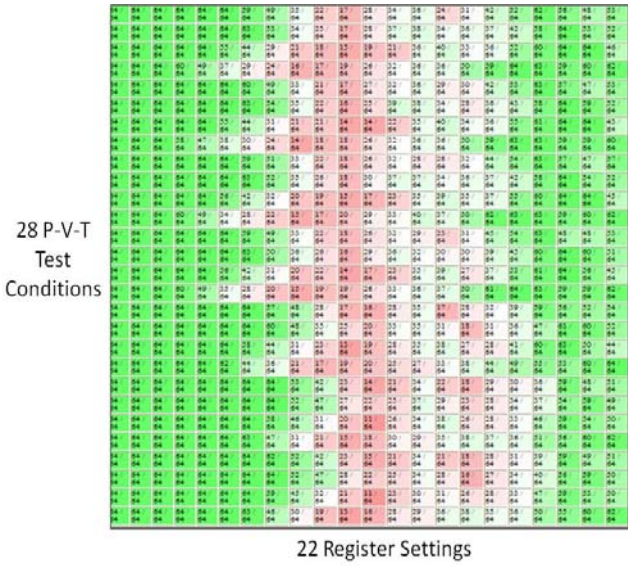

Figure 8: The graphical presentation of data from 39,424 READ and WRITE tests provides a high level view of passing (green) and failing (red) tests.

B. IO Cell for Multi-protocol Serial Link

An IO cell was designed to support 12 different standard serial link protocols. The data rate ranges from 1 to 6.375 Gbps. The design was tested at -40°C and 125°C to simulate the extreme environmental conditions of its applications. Figure 9 shows the total jitter when the cell was operating in the CEI6 mode at 6.375 Gbps.

In the highly automated LabStation environment, the results for 15 devices (3 each from 5 process splits) with 4 lanes per device and 9 different voltage-temperature conditions were collected in one day. Further analysis was also automated so that the results were quickly posted on a network for the design team to review.

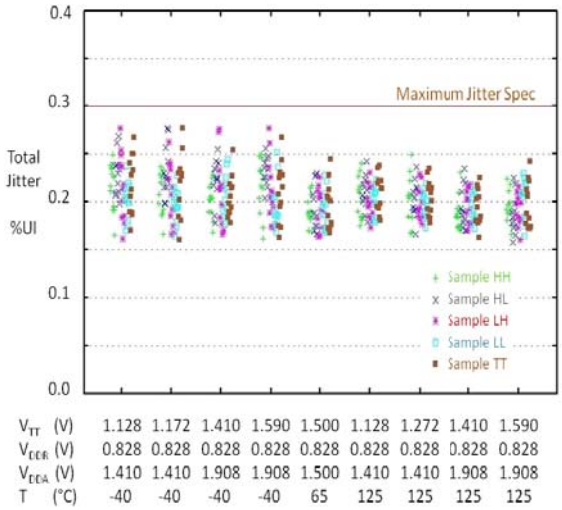

Figure 9: Total jitter at extreme temperatures and voltages.

temp (C)	vdd (V)	lane	RJ (rmsUI)	ISI (UI)	DJ (UI)	TJ (UI)	eye_image	image
25	1.20	4	0.0100	0.029	0.034	0.173		
0	1.26	4	0.0100	0.029	0.035	0.174		
125	1.08	4	0.0100	0.033	0.038	0.176		
125	1.26	4	0.0098	0.035	0.042	0.178		
0	1.08	4	0.0101	0.031	0.039	0.179		
0	1.26	0	0.0098	0.040	0.044	0.181		
25	1.20	0	0.0100	0.040	0.043	0.182		
0	1.08	0	0.0101	0.038	0.044	0.184		
125	1.26	0	0.0097	0.049	0.053	0.188		
125	1.08	0	0.0101	0.049	0.049	0.189		

Figure 10: Extra jitter information collected when testing the data links for compliance with total jitter specification.

While collecting characterization results to confirm compliance to data sheet specifications, it is also important to collect as much information about the system under test as efficiently as possible. The additional information can be used for debug purposes and to provide in-depth understanding beyond the specification. Figure 10 shows detailed jitter results from testing the fiber channel protocol at 4.25 Gbps. The table shows the breakdown of different components of jitter (i.e. deterministic jitter and random jitter) and scope screen shots of eye diagrams and jitter histograms. These screen shots are extremely helpful in quickly indentifying design failures and test setup issues. Collecting the extra results during the normal characterization flow saves valuable time by avoiding the need to re-setup test equipment and reproduce failures. Figure 11 is a screen shot from an oscilloscope controlled by LabStation during characterization.

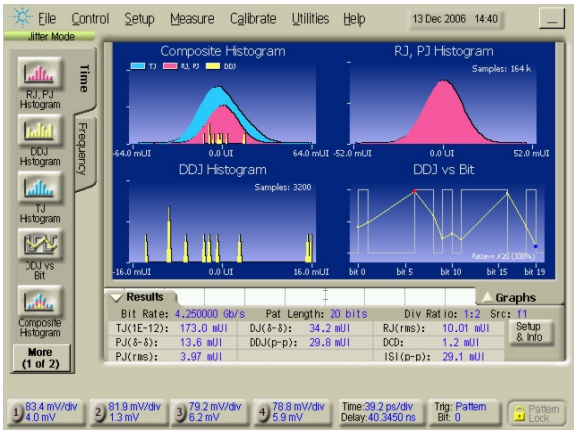

Figure 11: Results of jitter decomposition performed by a digital oscilloscope.

XI. Temperature Control

During characterization a temperature head is placed on top of the test chip to maintain the test temperature between -40°C and 125°C. LabStation communicates via a RS232 port with a temperature controller to set the test temperature (see Figures 1 and 7). The controller in turn supplies the power needed by a Peltier device to control the temperature of the test chip.

Figure 12 shows a temperature head mounted on a test board. The head is secured to the test socket using an adapter. The top cover and the plastic side walls together form an enclosure that protects the test chip and socket from the condensation of the moisture in the ambient air.

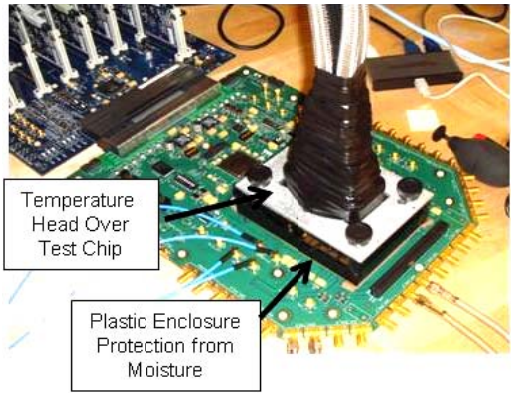

Figure 12: Temperature head with enclosure over test chip.

In the LabStation environment, the test platform and power supply need not to be placed inside a chamber with temperature and moisture control. Characterization at extreme

temperatures can be carried out on a standard test bench in a laboratory without expensive supporting facilities.

Besides validating the temperature specification of the test chip, data collected at various temperatures are useful for electrical-thermal co-design of the system under test [2].

XII. CONCLUSION

The LabStation characterization environment offers an effective set of automated test tools for validating the designs of memory subsystems and high speed serial links for production. It is particularly valuable for the consumer electronics industry to overcome difficulties associated with short development cycles and product lives.

In addition, the versatile environment can be easily updated to add features for the validation of new generations of IO technology for future products.

ACKNOWLEDGEMENT

The authors would like to thank Rambus senior management team for their support throughout the development of the LabStation Characterization Environment. We would also like to thank our partners, including Sony, IBM, Toshiba, TI and others, who provided much valuable feedback to the Labstation development team.

REFERENCES

[1] K. Chang, S. Pamarti, K. Kaviani, E. Alon, X. Shi, T.J. Chin, J. Shen, G. Yip, C. Madden, R. Schmitt, C. Yuan, F. Assaderaghi, M. Horowitz, “Clocking and Circuit Design for a Parallel I/O on a First-Generation CELL Processor”, *ISSCC* 2005, pp526-52.

[2] T. G. Yip, W. T. Beyene, G. Kollipara, W. Ng and J. Feng, “Electrical-Thermal Co-design of High Speed Links,” ECTC 2010, Las Vegas, June 2010.

[3] P. Yeung, A. Torres, P. Batra, "Novel Test Infrastructure and Methodology Used for Accelerated Bring-Up and In-System Characterization of the Multi-Gigahertz Interfaces on the Cell Processor," date, pp.137, 2007 Design, Automation & Test in Europe Conference & Exhibition, 2007

[4] W.T. Beyene, C. Madden, J.H. Chun, H. Lee, Y. Frans, B. Leibowitz, K. Chang, N. Kim, T. Wu, T.G. Yip, R. Peregro, “Advanced Modeling and Accurate Characterization of a 16 Gb/s Memory Interface,” *IEEE Trans. On Advanced Packaging*, vol. 32, no. 2, pp. 306-327, May 2009.

Authors Biography

Pradeep Batra is currently a Senior Principal Engineer and Software Architect at Rambus Inc. He is the inventor of the LS Script language for LabStation. His current research focus is operating systems for mobile platforms. Since joining Rambus in 1992 he has developed software tools for bring-up, characterization and diagnostic of RDRAM, XDR, and DDR memory subsystems. His tools were instrumental to the development of products including Nintendo64, RDRAM based PC's, graphics cards, Playstation3, and TI DLP projectors. Pradeep received his BS in electrical engineering from IIT Delhi and his Masters in Electrical Engineering from UCLA. He has authored several papers and has over ten issued patents. Prior to joining Rambus he was a member of the SPARC processor design group at Sun Microsystems.

Arun Vaidyanath is a Senior Engineering Manager at Rambus Inc. His team is responsible for delivering system solutions and platforms for the validation of Rambus high speed IO and memory interfaces. His team has developed system platforms for products like the PS3 and DLP projectors as well as LabStation enabling platforms for the development of Rambus advanced IO technologies. Mr. Vaidyanath has over 15 years of experience in the areas of Signal Integrity and System engineering. He has a B.Tech in Electronics Engineering from the National Institute of Technology, Calicut, India and M.S. in Electrical and Computer Engineering from the University of Arizona. He has several papers and publications in technical conferences and journals in the areas of signal integrity and system engineering design

Adrian Torres is currently a Senior Manager of the Product Engineering Team at Rambus. He led the effort to develop the LabStation Interface Module and the modular power platform for the LabStation environment. Upon joining Rambus in 2003, his primary role was to co-develop test and validation platforms for the high speed interfaces, FlexIO processor bus and XDR memory bus, in the PlayStation 3 game console. He has also managed system teams in various Rambus advanced developments such as the recent Mobile XDR initiative. Presently he manages a team of product engineers to ensure Rambus PHY products are ready for full production. Prior to Rambus, he has 5 years of experience in testing, validation and in-system diagnostic tools development for Itanium based enterprise servers at HP. He received his B.S. in Electrical Engineering and B.S. in Computer Engineering from the University of California at Davis.

Philip Yeung joined Rambus Inc. in June 2002, where he currently is the Director of System Solutions and Quality Group responsible for all aspects of system technology, packaging, system engineering, testing, software and product engineering. Under his direction, the development of the LabStation environment began in 2003. Philip was the recipient of the Rambus Global Values and Leadership Award in 2009. He received his BS degree in electrical and computer engineering from California State University, Chico, MS degree in electrical and computer engineering from University of Wisconsin, Madison, and MBA degree from University of California, Davis. He was a senior hardware design engineer at HP General System Lab working on PA-RISC Unix server design, and was a design lead and manager at Centerpoint Broadband Technologies working on high speed line cards and backplanes for RF/optical telecom system.

William N. Ng is the Senior Manager for the System Technology Team at Rambus Inc. His team drives innovations in silicon package, test technologies, test software for mobile platforms and PCB design for high data rate systems. He developed LabStation based methodologies for characterizing silicon of high-speed serial and parallel interfaces, including XDR memory, PCI Express, CEI6, XAUI, fiber channel and SATA. His contributions to LabStation also include the automation of processing high volumes of data collected by the LabStation software. William received his BS in Electrical Engineering and Computer Science with Honors from the University of California, Berkeley. Prior to joining Rambus, William worked on the logic and micro-architecture for 3D graphic pipelines when he was with ATI Research. He was also with Chromatic Research and Velio Communications, in both design and customer support roles.

Tsunwai Gary Yip joined Rambus Inc. in 2000 and is currently a Senior Principal Engineer. He developed many system clock solutions for LabStation test platforms, and the temperature control for maintaining test chips between -40°C and 125°C without using a temperature chamber. His interests are in the areas of EMI/EMC of mobile systems, low phase noise clocks for 10 - 30 GHz data links, phase noise analysis of PLLs, and thermal analysis of system and components. He is also active in new technology trends in mobile handsets and platforms, consumer electronics and PC graphics. Dr. Yip is an Associate Fellow of the American Institute of Aeronautics and Astronautics. He was a recipient of the Ralph Teeter Award from the SAE International for outstanding young educator in aerospace technology and was recognized as a Recent Outstanding Alumnus by the University of Illinois, Urbana, where he received his Ph.D. and M.S. degrees.