2016 21st IEEE European Test Symposium (ETS)

Practices in High-Speed IO Testing

An Embedded Tutorial

Salem Abdennadher
Intel Corporation
Folsom, California, USA
salem.abdennadher@intel.com

Saghir A Shaikh
Broadcom Ltd.
San Diego, California, USA
saghir.shaikh@broadcom.com

***Abstract*—With advances in VLSI technology, process, packaging, and architecture, SoC dies continue to increase in complexity. These advances have resulted in an unprecedented rise in design marginalities, manufacturing flaws and customer returns in SoCs with High-Speed IO circuits. This situation presents a challenge to develop sophisticated but low-cost test solutions. DFT-based test methods offer solutions to this challenge. This tutorial paper provides a summary of industry practices in DFT-based approaches to testing High-Speed IOs and their comparison with the specification-based tests.**

Keywords—I/O, DFT; BIST; ATE; Jitter; IO Loopback; Eye-Margining, PCIe, DDR

I. Introduction

The System-on-Chip (SoC) die area occupied by I/O and High-Speed IO, in particular, has been increasing with each process generation; some estimates place this as high as 30% for recent process generations. Composed of analog and digital circuitry, High-Speed IO lack of formal fault models which provide logic and memory arrays a mean to assess test coverage. Subsequently structured systematic test methods have not been developed to focus on manufacturing defects. Characterizing the good and faulty behavior of high-speed IOs remains difficult because their behavior spans a continuum, unlike digital logic and arrays which can be characterized by discrete values of 0s and 1s (by their digital nature). Also, digital logic/array designs are amenable to reconfiguration through DFT logic to facilitate test. High-speed IO behavior can be very sensitive to even minor perturbations introduced by any circuitry added to observe/control the analog logic.

High-speed IO circuits vary in behavior based on the type of clocking architecture used. For Forward Clock (a.k.a. Source Synchronous) designs, a high-speed reference clock is forwarded from TX chip to RX chip. These IO circuits require separate clock channel and clock amplifier. For the Embedded Clock IOs, Clock Data Recovery (CDR) techniques extract the clock frequency and phase from the incoming data [1]. Common specifications between the High-Speed IOs lend to common test approaches. However, each IO architecture type requires customized implementation which can ensure adequate coverage. This tutorial presents testing methods for both clocking types of IOs as practiced in the industry.

Initially High-Speed IO testing focused on specification testing which verifies whether the High-Speed IO complies with the specs listed in a data book or the standards (e.g. PCI-SIG, JEDEC, MIPI Alliance…). To support functional and specification-based tests Automatic Test Equipment (ATE) requires the following: high-speed pattern generator and data capture capabilities, high timing accuracy (order of picoseconds), and well-controlled signal integrity. For products with multiple High-Speed IOs face, a specification/functional approach lead to an expensive test cost. Typically SoCs are tested with reduced pin-count and low-speed testers. Hence, to reduce test cost, multiple companies pursue DFT-based test methods which permit the usage of low-cost testers [2-8]. Designing I/Os with loopback provides a basis for all DFT-based test methodology enabling the use of such low-cost ATEs. The next section describes the several loopback schemes and the associated DFT requirements. Later sections cover timing-margining, tests specific to CDRs and voltage margining approaches.

Many High-Speed IO interfaces operate at GHz frequencies--well beyond the signaling frequency supported by the low-cost testers used for today's SoCs--creating the need for an even faster tester to perform manufacturing defect tests. Fast testers are expensive testers, so there is a definite motivation to define some other means to test the IO interface at full speed.

II. I/O Loopback

Having I/O loopback modes in the design mitigates the need for the costly high-speed testers to support manufacturing test. Loopback modes can also support silicon characterization with bench equipment and system level characterization/test. . Internal Loopbacks are implemented on-chip to reduce the reliance on the load board and to enable in-field system testing. There are four differences schemes for implementing the Loopback: (i) analog near-end; (ii) digital near-end; (iii) analog far-end; and (iv) digital far-end.

A. I/O Loopback for Basic Functionality

High-speed design with IO loopback capability can be tested at-speed without connecting to the tester channel. In loopback schemes, the output of the system is connected to the input forming a data transfer loop. To enable a basic loopback test, one needs the ability to apply a pattern and the mechanism for checking the expected results. Patterns can be simple (1010…) or complex (10100…, or pseudo-random).

978-1-4673-9659-2/16/$31.00 ©2016 IEEE

In a near-end analog loopback, data exits the TX and enters the RX. This connection can be made internal to the DUT or on a load board i.e. external to the DUT. As the connection is looped externally, from the Analog Front End (AFE) perspective, the operation is identical to a normal transmit and receive sequence. Figure 1(a) shows an example of the external loopback. For DC parametric, analog near-end loopback implementations typically requires relays on the load board. An internal loopback to support this path may also be present on the die to support at speed testing at wafer testing.

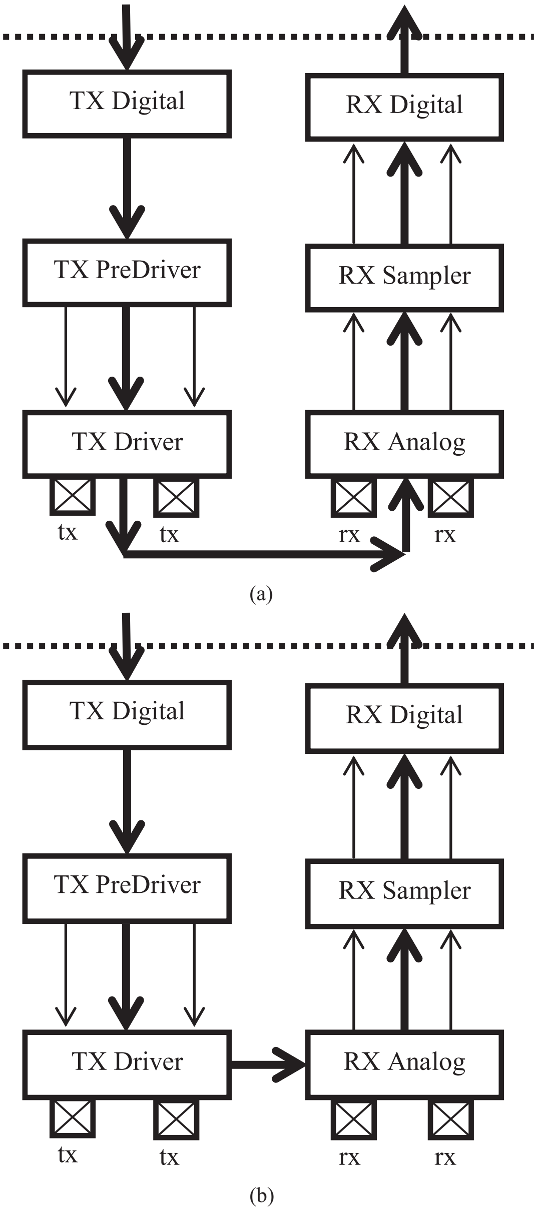

Fig. 1. Analog near-end loopback: (a) external (b) internal.

In analog near-end internal loopback mode TX loopbacks to the RX via switches also. In normal functional mode, the switches are off. In the loopback mode, the switches are turned on, and the data signals are sent from TX to RX. The switches are located on the both ends of the TX pads and RX pads to minimize pad-capacitances and cross-talk. Any intermediate nodes between the two nodes are pulled to ground to reduce crosstalk. Attention to signal integrity of all these connections is of utmost importance; especially if both external and internal analog near end loopback is needed.

In Digital near end loopback, the TX data is loop backed to RX bypassing the analog all together as shown in Figure 2. This loopback scheme is easy to implement and does not cause and perturbation to the analog circuitry. Digital near-end loopback can be used for core functional testing without cycle slip issues usually seen in analog near-end loopback.

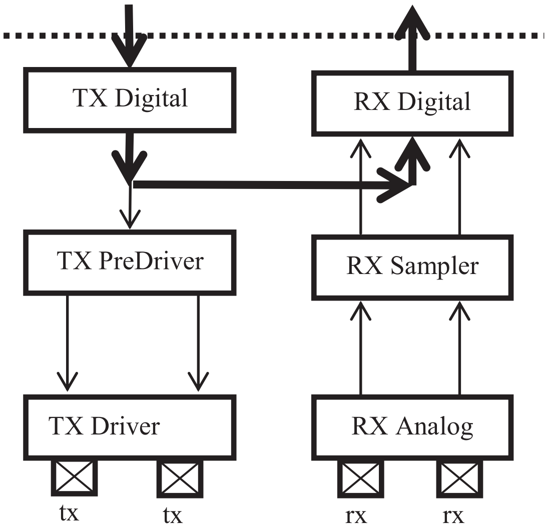

Fig. 2. Digital near-end loopback.

In an analog far-end loopback, Figure 3, data that received by the Analog Front End (AFE) is looped back onto its transmitter bypassing the digital portion of the IO. In this scheme, the tester or lab equipment receive and compare the transmitted data with the original data sent to AFE. This scheme can be very helpful at first silicon as it does not require the whole DUT to be functional.

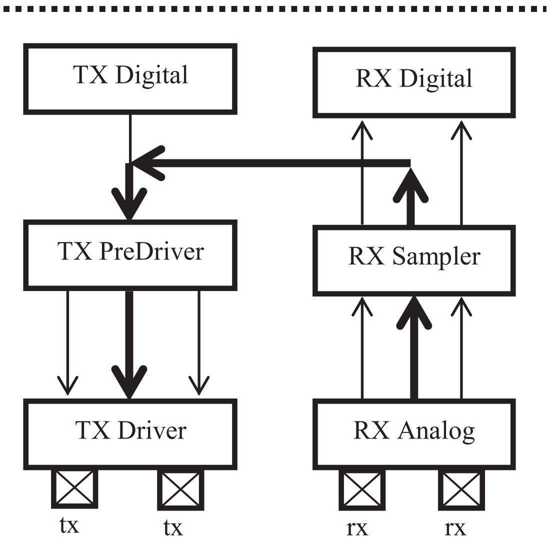

Fig. 3. Analog far-end loopback.

Digital far-end loopback mode is shown in Figure 4. This loopback scheme involves feeding the IO received data back into the transmitter. This way the data is to be transmitted out again on the TX interface after being extracted and re-clocked by the core receiver logic. Digital far-end loopback involves an external pattern generator to pump in the specific data pattern. In digital far-end loopback, the AFE just functions like normal transmit and receive operations. The looping is happening within the core, and the loopback data appear as standard data to the TX. Digital far-end loopback is used extensively for Bit Error Rate Testing (BERT).

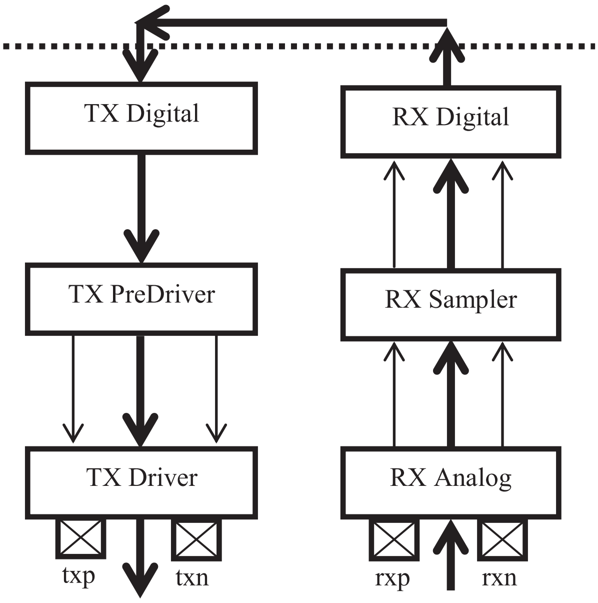

Fig. 4. Digital far-end loopback.

B. *DFT requirements to support I/O Loopback implementation*

To fully support test or characterization, loopback schemes require the following: pattern generation, pattern capture and comparison, re-start after the completion of the training sequence. In a complete BIST solution, these resources are implemented on-chip. Figure 5 shows the implementation of the loopback schemes on a PCIe system.

The on-chip IO Test Engine generates and captures the pattern and compares using the Multi-Input Signature Register (MISR) block. This Built-in Self-Test (BIST) scheme can flag any errors in the signals paths and provides a pass/fail report to the tester. Such a basic BIST schemes are designed to work with works either both the internal as well as external loopback schemes. Both external and internal loopback schemes need careful consideration regarding the signal integrity at the load board. Internal Loopbacks require careful design of RF stubs to limit the reflection at the system level. As described earlier, the loopback schemes are very prevalent as they can because during all the testing phases; e.g. sort, class, device validation, debug and platform testing.

By themselves testing with a loopback schemes verifies at speed functionality; i.e. it is a gross test of timing and voltage levels. To cover the specifications, designers implement additional DFT features. The following section describes a few practical schemes for the coverage of timing and voltage specs.

III. Timing Margining Tests

The eye-opening and jitter comprise the two critical measurements carried out to determine the timing of the high-speed signals. Jitter is the deviation of a timing event of a signal from its ideal position. Data errors result when this deviation extends past the sampling point at the receiver. Three types of jitter measurements include (i) intrinsic jitter, (ii) jitter transfer, and (iii) jitter tolerance tests. These jitter measurements can be performed with a high-speed costly tester or instruments. For manufacturing test, DFT has enabled lower cost solutions.

Eye-diagram is an accumulative measurement of jitter over the time. Data pulses of the signal stream are a layover on top of each other over a period resulting in an eye-shaped diagram. Ideally, the data eye shows no jitter as Figure 6 (a) illustrates.

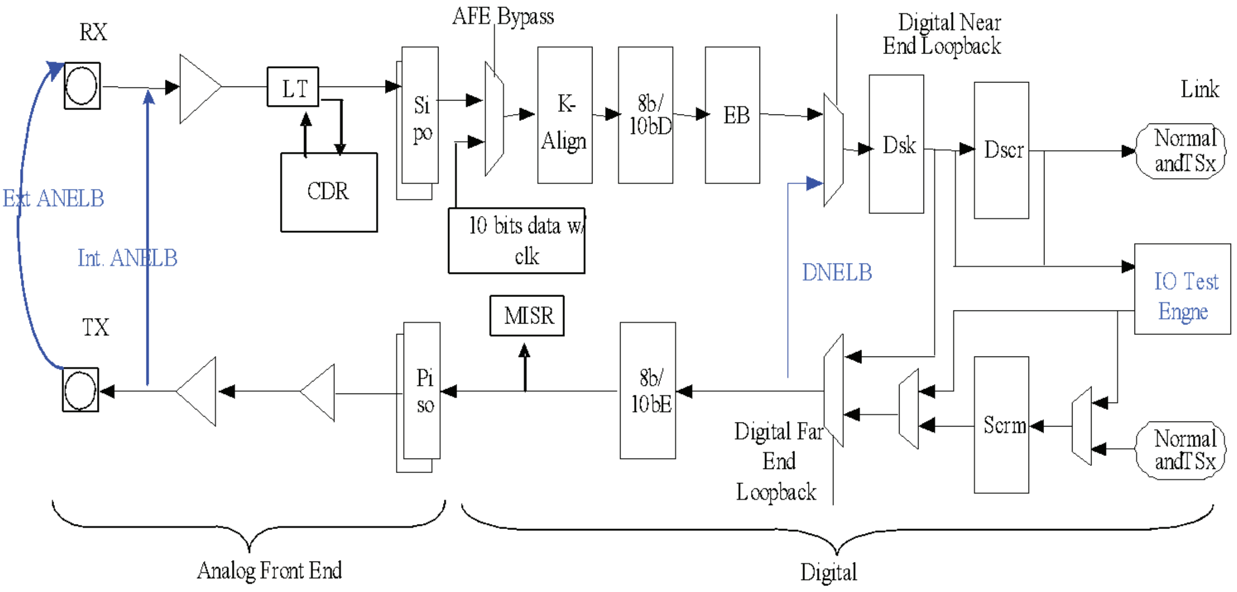

Fig. 5. Loopback BIST implementation in a typical PCIe interface.

The two basic eye opening measurements exist eye-width and eye-height. This section focuses on eye-width. TX circuit and transmission path characteristics cause the eye-diagram to close and manifest into reduced eye-width as shown in Figure 6 (b). Note Figure 6 (c) depicts an eye that is too narrow, i.e. fails to be detected. The transmitter circuit should generate as little jitter as possible. Testing the High-Speed IO timing performance requires measuring jitter from the transmitter output. In a loopback path, one cannot explicitly separate TX from RX from transmission path effects. Jitter testing requires the jitter attenuation and inducing mechanisms to mimic the effects of the transmission channels and to ensure the RX circuit's ability to detect the minimum eye width. Testing for the jitter is one of the most common methods to do timing test.

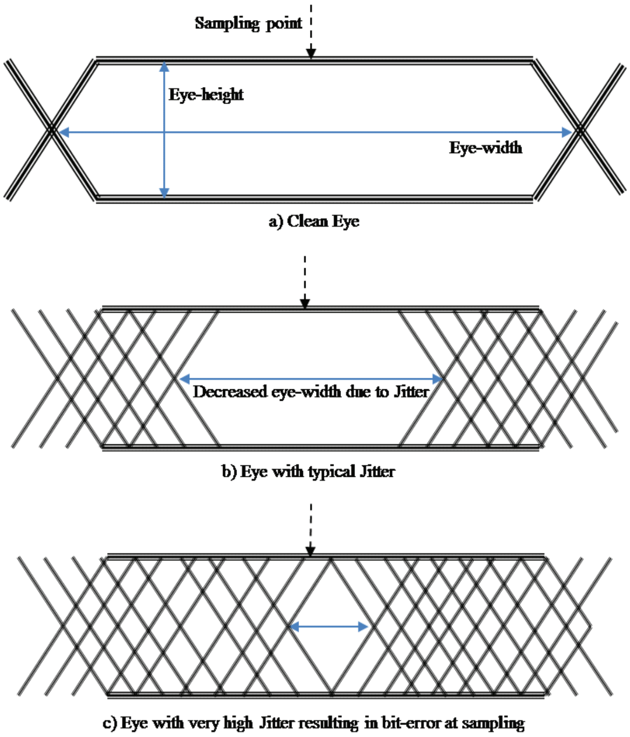
Sampling point
Eye-height
Eye-width
a) Clean Eye
Decreased eye-width due to Jitter
b) Eye with typical Jitter
c) Eye with very high Jitter resulting in bit-error at sampling

Fig. 6. Eye-diagram measurements and Jitter affects.

Typical jitter measurement techniques can be carried out with instruments, e.g., an oscilloscope with eye-diagram template, bit-error-rate-analyzer, spectrum analysis, time-interval-analyzer. Typical jitter injection techniques include phase modulation and waveshaping using an arbitrary waveform generator (AWG). Jitter testing using instruments described is too costly for the volume productions. Existing AWGs on low-cost ATE do not have adequate sampling rates and bandwidth. Also, these techniques take several seconds to measure and induce further jitter [9]. Moreover, instruments for the measurement techniques on commonly used ATE systems for SoC products suffer from accuracy, resolution and bandwidth limitations. Thus, for test coverage of timing related errors, DFT-based methodologies such as Timing Margining and DFT for Data Clock Recovery are used.

A. *Timing Margining Test for High-Speed IO*

The industry has embraced Defect-based IO screens using the Timing Margining [1-2] methodology as an alternative to traditional functional timing test. Timing Margining measures the amount of margin within a data eye. For a serial interface such as Intel's QPI link, the width (in ps) between the data eye edges is measured. Ideally, the data eye width would be 1 UI (Unit Interval). In reality, because of jitter, circuit non-idealities, setup/hold time, trace length mismatch and clock recovery inaccuracies, the data eye width is reduced, and the margins on both sides are unequal.

For a serial link interface, Figure 7 shows a simplified diagram of a typical timing margining implementation. In a normal mode, data is fed from the core, serialized and is transmitted out. The forwarded clock is also transmitted. In Timing Margining mode, both the data and the forwarded clock transmitters to loopback to the receivers. The received data is captured, sampled and compared to the sent data to qualify a part as pass or fail. The looped back clock feeds a Delay Locked Loop (DLL) and a Phase Interpolator (PI) to adjust precisely the strobe in the center of the data eye. During Timing Margining, the PI setting is manually overridden, and the clock is forced to move left and right until it reaches the edges of the eye. When an edge is reached, the comparison block records a fail, and the current PI setting is scanned out.

The DFT circuitry shown here enables a Timing Margining test. Figure 8 illustrates a simulated data eye where the sampling strobe is moving to find the failure points at the right and left edges. The width of an eye can be obtained through these failure points. This test measures the TX Intrinsic Jitter because if there were excessive noise produced by the TX driver circuitry, a small eye would be measured. Test methods based on DFT Timing margining covers the same faults as jitter tolerance testing would detect.

Two modes of eye width measurement exist-- an automated and a manual method. The automated method uses a built-in state machine (FSM) to adjust the Data PI; the manual method is programmed through software by the user. Both modes control the PIs in the same manner to measure timing margin. The auto-margin FSM can be implemented to measure the width of the data eye through automated mechanisms to allow for shorter test times. As the strobe point is moved a single bit-error could indicate the edge of the eye; some implementations may want to count the number of bit errors. Therefore, the BIST engine often includes a programmable maximum error count to set the number of errors required to identify a fail point. The eye width measurements can then be read through each IO's configuration registers. The actual correlation between DFT results and the real jitter measurement can be established during the first silicon characterization.

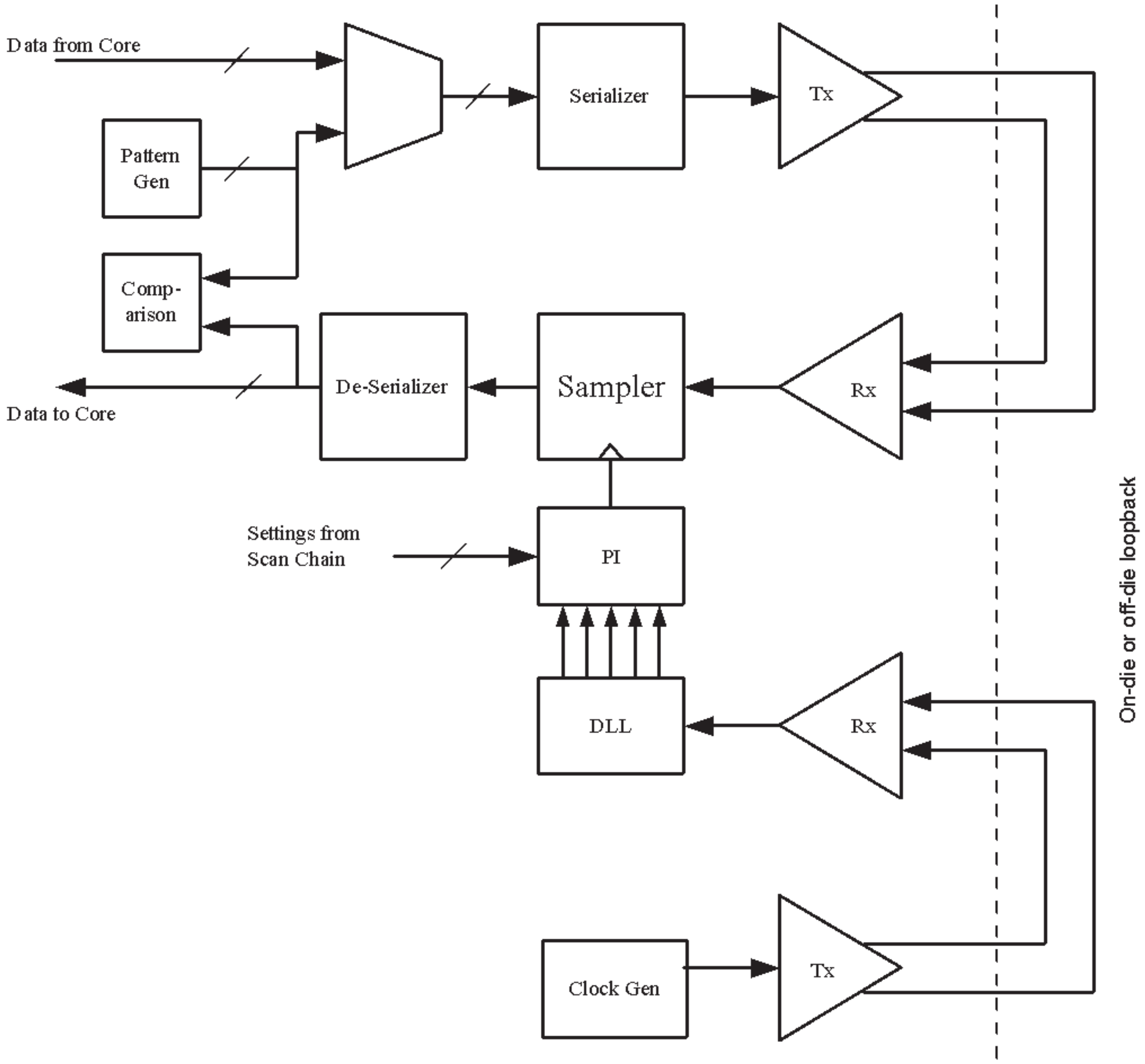

Fig. 7. Time margining general approach for source synchronous (SS) IO’s [3].

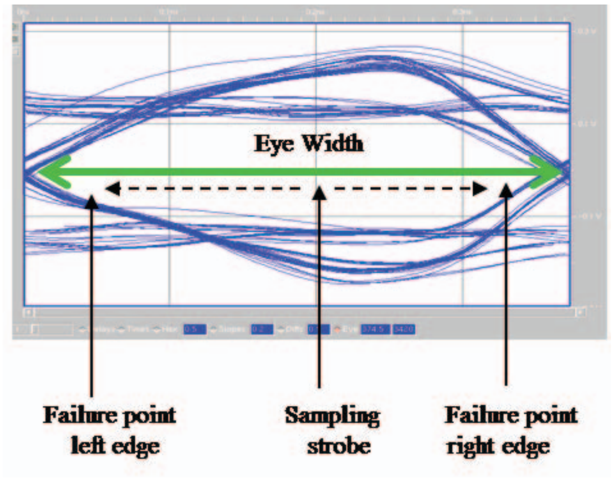

Fig. 8. Measure of eye-width by adjusting the sampling strobe

This timing margining concept has been applied to various single-ended I/O configurations such as common-clock, DDR, CMOS, source-synchronous, as well as serial interfaces such as Serial ATA and PCIe.. Most serial interfaces already have a phase interpolator in the receiver circuitry, which can be re-used to perform timing stress on the received data eye as described above. Timing Margining can also be applied in non-loopback configuration by sending data from one chip to another chip if a comparison pattern protocol has been implemented such as IBIST [7].

B. DFT-Based Test Methods for Data Clock Recovery

Many High-Speed IOs requires Clock Data Recovery (CDR) circuits and other synchronization circuits to correct for any clock/data skews. CDR performance dictates the overall I/O system performance, such as Bit-Error Rate (BER). Specific DFT based test methods are used to test these synchronization circuits. CDRs track the optimal position of the recovered or forwarded clock. This CDR characteristic must be fully tested to ensure the tracking range is appropriate. Hence, the whole tracking range must be stressed since the production test environment is different from the end-user system environment.

For Phase Interpolator CDR Implementations, the production test method should purposely stress the Phase Interpolators (PI) to its limit. High-speed IO that uses an embedded clock approach has a DLL and PI in the receiver as illustrated in Figure 9, but do not have a DLL in the transmitter. Most implementations use more than one PI to capture multiple phases of data within one clock cycle. One structural test approach for the PI is to have the PIs test each other. The fundamental concept of the PI test is to compare the delay through two PIs while a known delay setting offset is applied on the PIs. The total delay is the sum of the Delay DLL plus the PI under test. The delay difference is verified by observing the phase relation of the PI outputs with a Phase Detector (PD).

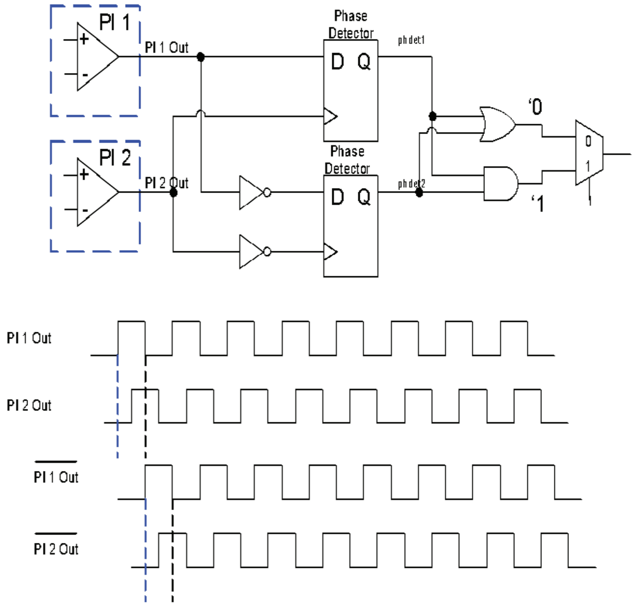

Fig. 9. PI mutual test method [10].

The graph in Figure 10 illustrates the delay curves through PI 1 and PI 2 as their settings are swept while preserving an offset between the PI settings. Ideally, with perfectly linear delay versus setting relations and no defect, the curves are parallel, the phase relation is preserved, and the PI delay output remains low. However, if PI 2 has a linearity defect somewhere along its curve (as shown on the graph), the phase relation changes and the PI Delay goes high. In this example the PI Delay is “sticky”, so it remains high, which is a “Fail”.

C. *DFT-based method for DDL based designs*

Some High-Speed IOs use Digital Delay Lines (DLL) for synchronization purposes. These DLL’s generate accurately-shifted clocks that are necessary to communicate with the external device. In a typical High-Speed IOs, there are multiple embedded DDL having a core function of a buffer. Defects in certain delay line can disrupt the expected functionality of the synchronization mechanism hence the need to exercise all DLL in a production test. Loopback and timing margining tests exercise only a fraction of these DLL’s, to facilitate an exhaustive DLL test, logic needs to be added.

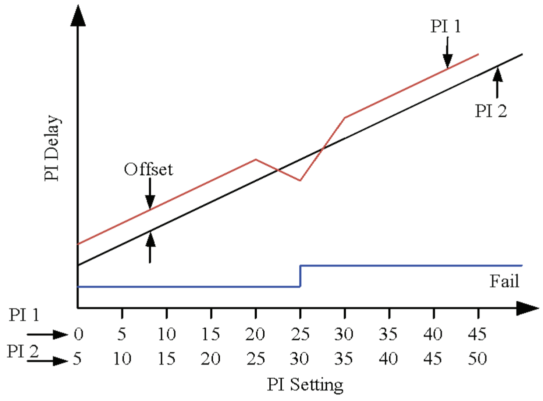

Fig. 10. PI settings for lead/lag mutual test [10].

In the ring oscillator mode, all DLL’s are connected into a single ring [11]. This ring is achieved by connecting the digital test output of one delay line to the digital test input of another delay line. The digital test output of the last Delay line is connected to an NAND gate and looped back to the digital test input of the first DDL to form an oscillator as shown in Figure 11.

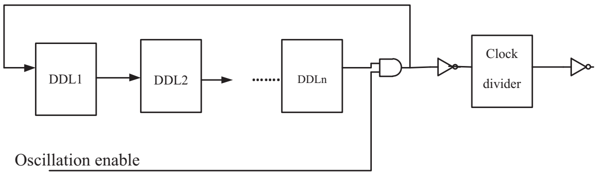

Fig. 11. DDL Oscillation BIST [11].

IV. Voltage Margining Tests

While timing-margining and synchronization test techniques target timing specification and eye width measurement voltage margining targets eye height measurement. To provide relevant results, Voltage Margining measures the data eye for logic high and logic low while running at speed. The standard and optimum location for such a measurement is at the center of the eye with respect to time. As in timing margin, the DFT implementation reuses existing TX and RX circuitry to support the assessment of voltage margin. The sensitivity of a receiver is a measure of its ability to pick up weak signals. The higher value of the sensitivity of the receiver indicates the greater ability of the receiver to pick up the weak signals. The receiver sensitivity is defined as the weakest signal that produces an acceptable Bit Error Rate. The typical method to test for receiver sensitivity is to apply a minimum detectable signal input voltage to the receiver through the ATE. However, this method is not ideal for extremely high sensitivity receivers because it is hard to generate these minimum detectable signals using existing ATE.

DFT based voltage margining shrinks the data eye going into the RX till bit errors are observed. The data eye is modified by either changing the TX driver strength or by varying the common modes of the signal going into the receiver pads by different amounts thus shrinking the eye as shown in Figure 12.

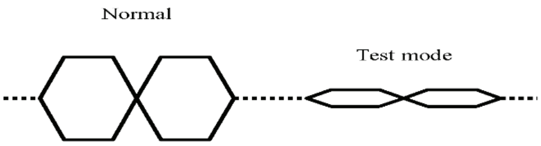

Fig. 12. Receiver stimulus for sensitivity test.

A. Voltage Margining using TX & RX Compensation Circuitry

The TX compensation circuitry adjusts the termination resistor (R-comp) and the driver current bias (I-comp) to compensate for process variation. However, with adequate range, this circuitry can be re-used to perform a voltage margin test. It can be used for test purposes if it has enough range. This DFT-enabled capability requires controllable termination (R-comp) and transmitter with output swing control (I-comp) to stress the receiver's sensitivity as shown in Figure13. To characterize the receiver for HVM, we want to make sure it works for the minimum and maximum eye requirements. The TX-eye height needs to be reduced below the TX's minimum specifications to stress the RX minimum specifications in loopback mode. This test method consists of degrading the data eye until failure.

Taking PCIe High-Speed IO as an example, the PCI Express specification is for a nominal differential impedance of 100 Ω + or - 10%, or 50 Ω single ended. The best case signal occurs when the termination at each end is the same, in this case, 50 Ω. When the termination is different, reflections occur and degrade the signal. If the RX and TX R-comp have a range of 40-70 Ω, then setting the RX and TX at opposite ends of the range should reduce the eye height greatly due to reflections. Similarly, I-comp can be used to adjust the height of the data eye or voltage swing of the TX. For test purposes, the TX-eye height can be reduced with I-comp to stress the RX voltage sensitivity. The receiver needs to detect a sound signal down to 175mV, well below the minimum differential Transmitter output of 800mV.

B. Voltage Margining using Common Mode Control

The input common mode can be applied to the RX to measure the available voltage margin on the RX data eye. The test varies the RX input voltage swing till eye is closed. This scheme is shown in Figure 14.

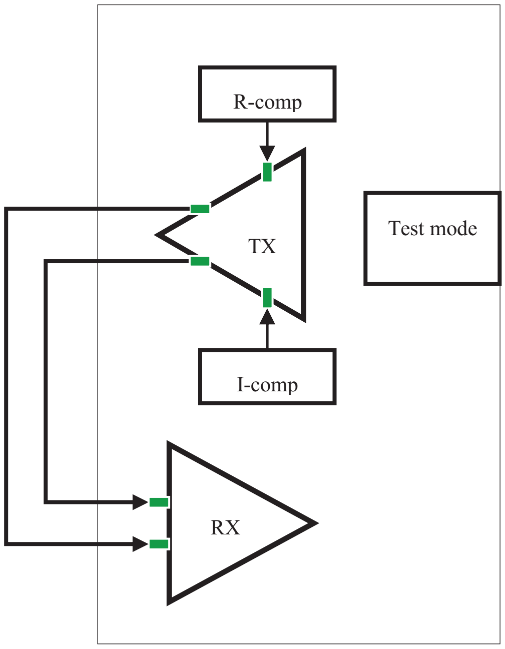

Fig. 13. Voltage margining using compensation control [10].

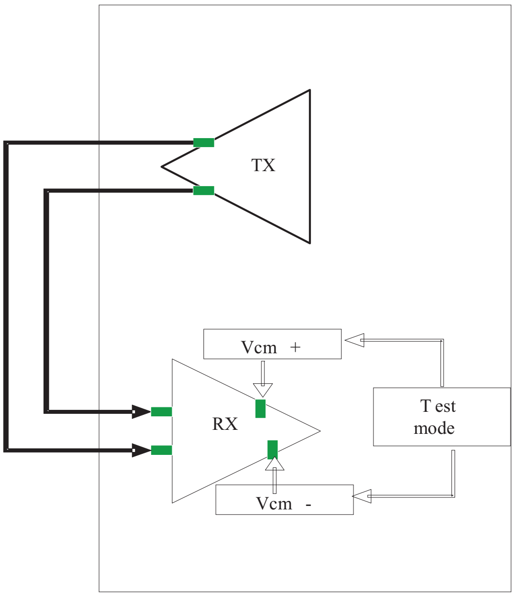

Fig. 14. Voltage-margining using common mode control [10].

C. Voltage Margining in DDR interfaces

In DDR interfaces, the Voltage Margining can be done by controlling the reference voltage VREFDQ. In DDR3 interface, the VREFDQ is supplied externally and is thus provide complete control using external sources. With DDR4 interfaces, on the other hand, VREFDQ is generated on the die, which means that the DDR4 internal VREFDQ must be controlled varied under the control of the memory controller.

The on-die VREFDQ control through a programmable configuration register mechanism is needed.

V. Summary

Loopback based margining tests had been broadly adopted in industry because they can test most High-Speed IO functionality without requiring costly ATE platforms. Although they cannot completely replace traditional jitter testing due to random jitter component in the IO, Duty cycle distortion in clock signals, the characteristic nonlinear behavior of the CDR, and jitter amplification, they are used effectively to detect common manufacturing defects and design marginality in a production environment.

The described DFT based on loop-back tests in this paper verifies device performance in five areas: transmitter at functional speed, receiver at functional speed, receiver jitter tolerance, receiver minimum detectable level, and transmitter implied jitter generation. The Table below maps the DFT based margining methods described to the corresponding specification based tests.

TABLE I. Summary of the DFT Methods

DFT Method	Specs or circuitry exercised that would impact a spec
Timing margining	DCR, jitter tolerance
Voltage margining	Receiver sensitivity
PI Lead/Lag	Phase interpolators
DDL oscillation BIST	Digital Delay Lines

This tutorial paper has described eye margining hooks verify at speed functionality, available timing and voltage margin in a system and are essential to ensure fruitful and reliable data transfer according to the interface specifications. These versatile features can have proven useful in High-Volume Manufacturing (HVM) test, system-level EV, and component debugs & characterization. These efficient approaches for High-Speed IO testability have been widely deployed. These approaches rely on modifying the architecture of the design to fit some BIST structures to test High-Speed IO on the low-cost tester. The DFT and BIST techniques are very helpful to perform a high-level evaluation of these IOs and to check both functionality and performance while reducing the test time and simplifying the ATE requirements. However, test schemes are application specific and do require an investment in test development and characterization to establish test threshold values.

ACKNOWLEDGMENT

Thanks to Dr. Anne Meixner for invaluable comments and suggestion to improve this paper.

References

[1] Sam Palermo, Special Topics in High-Speed Links Circuits and Systems, Lecture Slides at Texas A & M, Spring 2010.

[2] Mak, T. M., Tripp, M., and Meixner, A., “Testing Gbps interfaces without a gigahertz tester” Design & Test of Computers, IEEE Volume 21, Issue 4, July-Aug. 2004, Page(s):278 – 286.

[3] Provost, B. et al.; “AC IO loopback design for high-speed microprocessor IO test,” International Test Conference, 2004, Proceedings, Page(s):23 – 30.

[4] Sunter, S.; Roy, A.; “On-chip digital jitter measurement, from megahertz to gigahertz,” Design & Test of Computers, IEEE Volume 21, Issue 4, July-Aug. 2004,Page(s):314 – 321.

[5] Sunter, S.; Roy, A “Structural tests for jitter tolerance in SerDes receivers”, International Test Conference, 2005, Proceedings, Page(s):1 – 10.

[6] Robertson, I.; et al. “Testing high-speed, large-scale implementation of SerDes I/Os on chips used in throughput computing systems,” International Test Conference, Nov 2005, Page(s):1 – 8.

[7] Nejedlo, J.J.; “IBIST ™ (Interconnect Built-In Self-Test) architecture and methodology for PCI Express: intel's next-generation test and validation methodology for performance IO;” International Test Conference, 2003. Volume 1, Sept. 30-Oct. 2, 2003, Page(s):784 – 784.

[8] Agilent Technologies, “Jitter Analysis: The dual-Dirac Model, RJ/DJ, and Q-Scale,” Available at www.agilent.com.

[9] Y. Cai et al., “Jitter Testing for Gigabit Serial Communication Transceivers,” IEEE Design & Test of Computers, Vol. 19, Page(s) 66-74, Jan-Feb. 2002.

[10] Anne Meixner, Akira Kakizawa, Benoit Provost, Serge Bedwani, et. al. “External Loopback Testing Experiences with High-Speed Serial Interfaces,” Proc. Int’l Test Conf. (ITC 08), IEEE CS Press, Los Alamitos, Calif., 2008.

[11] Octavian Petre, Hans Kerchoff, “On-chip Tap Delay Measurements for a Digital Delay-Line Used in High-Speed Inter-Chip Data Communications”11th Asian Test Symposium ATS 2002.