Elimination of Traditional Functional Testing of Interface Timings at Intel

Mike Tripp, T.M. Mak, Anne Meixner

Intel Corporation

Abstract

This paper summarizes the Design for Test (DFT) circuitry and test methods that enabled Intel to shift away from traditional functional testing of I/O's. This shift was one of the key enablers for Automatic Test Equipment (ATE) re-use and the move to lower capability (& cost) structural test platforms. Specific examples include circuit implementations from the Pentium® 4 processor, High Volume Manufacturing (HVM) data, and evolutionary changes to address key learnings. We close with indications of how this can be extended to cover the next generation High Speed Serial like interfaces.

1. The Last Straw: Source Synchronous Interfaces

Historically, I/O performance has been guaranteed by functionally testing each I/O Signal with respect to its data sheet specifications. In order to guarantee we never shipped a bad device, the test criteria was "tightened" (made more stringent) with respect to the device specification to account for uncertainty in the test. This "tightening" of the test limits is usually described as "guardbanding". To minimize yield impact, the design had to have sufficient margin to its published specification to accommodate the guardbands. So to minimize impact on interface performance (i.e., minimize the guardbands) we required high pin count testers with accurate edge placement capabilities. This not only drove up the cost of testing but more importantly the test guardbands resulted in a performance penalty that approached 25% of the interface period (i.e., 2.5ns on 100 MHz interface). Therefore, we were giving up interface performance because of our traditional test methods.

In a Source Synchronous (SS) interface the receiving agent captures data based on a strobe or clock that is provided by the driving agent along with the data. The address and data portion of the Pentium 4 Processor Front Side Bus is an example of a Source Synchronous Interface [1]. The critical timing parameters on an SS interface are all skews between the output or input signal and its associated strobe. For example on the data bus they are: Tvbd (data output valid before strobe), Tvad (data output valid after strobe), Tsuss (input setup time to strobe), and Thss (input hold time after strobe). One of the consequences of this signaling architecture is that only differential or same cycle jitter impacts true performance and common mode jitter or cycle to cycle variations does not impact the performance of the interface.

As described in [2], these properties of SS interfaces introduce additional test methodology complexities for the "common clock" based ATE equipment (i.e, all that was available when this technology was being established in '95 to '98) and resulted in an increase in the size of the guardbands that further degraded our "testable" performance. We developed a "Sliding Window" methodology (surprisingly similar to the "Sweeping Window" method described in [2]) to address the cycle by cycle requirement. While the "Sliding Window" method is still used today during design validation; it was too costly to implement in High Volume Manufacturing (HVM).

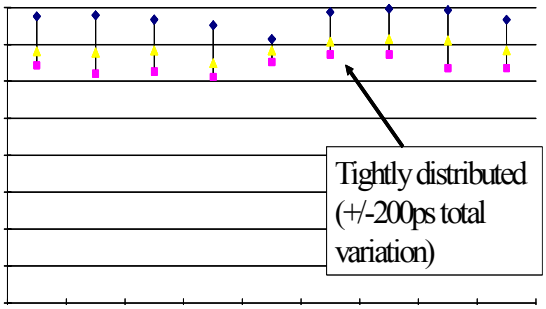

Figure 1: Characterization data from Source Synchronous interface on Pentium® II Processor

Figure 1 shows data from a SS interface on the back side level 2 cache bus of a Intel® Pentium® II processor. The Y axis is in 100's of pico-seconds and

on the X axis each set of results represents the min, max, and average value for a specific SS output parameter for the same group of signals on a set of devices from various process corners (slow, typical, fast), at hot and cold temperature and high and low Vcc. As can be seen, a consequence of the SS architecture is that the timings are independent of Silicon process skew (systematic variation that impacts the entire die or wafer), temperature, and Vcc. This uniform distribution of the critical timing parameters leads to an all or nothing yield situation, making it virtually impossible to trade a slight degradation in yield for higher interface performance. This eliminated a technique we sometimes used to minimize the impact of guardbands on the older "common clock" interfaces.

This data also highlights another property of SS interfaces; you can easily define groups of signals that are designed to behave identically to one another. These groupings correspond to the signals associated with a given strobe (or set of strobes in the case of the Pentium® 4 processor). These signals are designed with a high degree of matching and are physically located in close proximity to each other and their strobe(s). Later in this paper we describe how this matching can effectively be used to identify any individual signal that is impacted by a defect.

2. AC IO Loopback: Self-contained, Cycle by cycle, I/O Timing test

Just as we were primed for a shift to a new I/O test methodology, we were also developing plans to reduce tester capital spending and move to the lower capability structural testers [3]. To enable this shift, we developed an I/O test methodology [4] that only required an accurate clock source and did not require probing the individual signals. Since the method relies on a loop in the I/O buffer and addresses guaranteeing the AC Timing parameters, the methodology was labeled AC IO Loopback.

AC IO Loopback is a significant enhancement over a simple I/O loopback scheme [5] that was targeted primarily at screening Stuck@ (hard) failures. In [6] the authors describe enhancements to IBM's IO Wrap test methodology to address delay failures in the 10's of nano-seconds, our methodology was targeted to address impacts in the 100's of pico-seconds. Even with this screening goal, AC IO Loopback is a "Defect Based Test Method" as it does not directly measure a single specification. The measurements are directly related to functional parameters, but it relies on detecting "unexpected" performance in the I/O Loop to detect devices that are "likely" not to meet their AC Timing specifications.

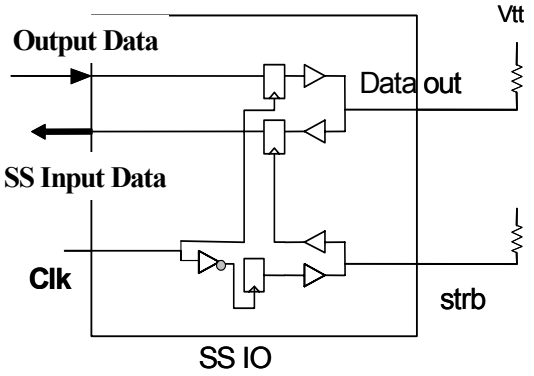

Figure 2: Block diagram of bi-directional Source Synchronous data signal and associated strobe

Figure 2 depicts a generic block diagram of Source Synchronous (SS) I/O interface and Figure 3 is the associated timing diagram. For SS interfaces there are two physical data path loops, one for the data signal and one for the strobe and both are required for the buffers operation.

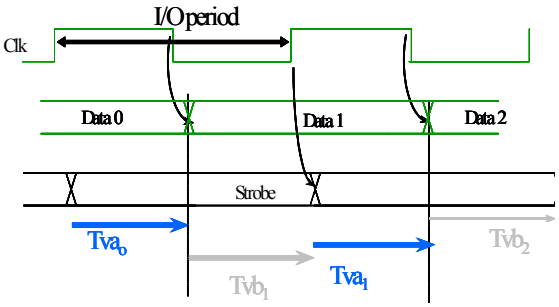

Figure 3: Timing diagram for Source Synchronous Interface

In the example in figure 2 the data is launched from the driver based on a falling edge of Clk and, mid-cycle, the Strobe is launched from its driver based on the rising edge of the same Clk. Given matched drivers and matched path lengths this places the Strobe in the middle of the data valid area providing for equal Tvb and Tva. At the end of the data loop, the output arrives at the receiver input latch and is captured in the latch when the strobe arrives. Just like in the system, in order for the data to be captured robustly it needs to be

stable at the input latch long enough before the strobe to satisfy the Tsuss and held long enough after the strobe to satisfy the Thss.

The input parameters (Tsuss & Thss) are very small, typically totaling less than 100ps (i.e., performance of a single latch). The drivers are matched devices and data and strobe paths are also matched. Therefore valid data can potentially be captured at frequencies up to 10 GHz! It is possible to detect defects that result in an increase in the critical timing parameters by observing a reduction in this maximum frequency. However, this approach could not really be used as the test method since 10 GHz clocks weren't readily available. So we designed and incorporated DFT circuitry to insert a known imbalance (stress) into one of the paths and used this to determine if the interface was operating as expected.

Three additions to the simple loop back scheme are necessary to support AC IO Loopback.
1. A capability to stimulate the loops with controlled data patterns. The minimum requirement is a toggling pattern to generate transitions since we are targeting delay, timing, defects;
2. A capability to stress the loop to failure;
3. A capability to detect when the loop is passing (expected data is being captured) and when it is failing (opposite data is being captured).

3. Pentium® 4 processor Implementation:

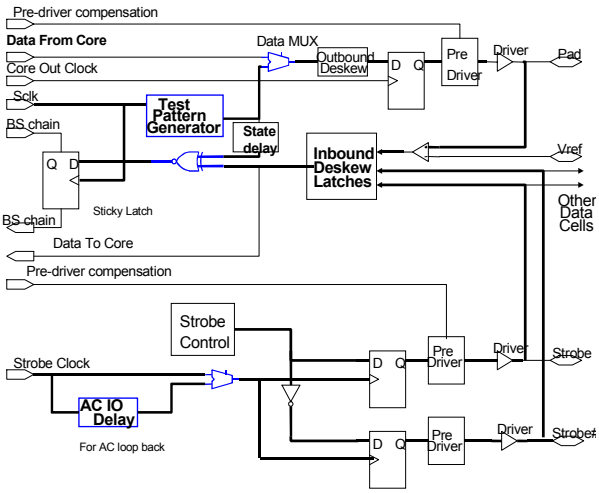

Figure 4: Block diagram of Pentium® 4 processor data signal and strobes

Figure 4 is representative of the data bus on the Pentium® 4 processor and includes the DFT circuitry to support AC IO Loopback [7]. In addition to the DFT, this diagram contains driver compensation [8], a differential comparator on the input and an additional strobe (strobe#) as compared to the simple generic diagram in Figure 2. In order to keep the maximum data rate on the strobe equal to the data signals and not rely on a 50% duty cycle strobe, the Pentium® 4 processor chose to have 2 strobes (Strobe, Strobe#) associated with each data signal which are used to capture even or odd data bits respectively [1].

The essential elements of the Pentium® 4 processor AC IO Loopback DFT implementation are:

- Per Pad two bit pattern generation, accessible via a JTAG controlled scan chain;
- A timing stress mechanism that can shift the position of the strobe generation (relative to the data generation) consisting of a delay chain programmable via a TAP controlled test configuration chain;
- A compare of expected value with a sticky bit mechanism accessible via Boundary scan as the pass/fail detection;
- Ability to exercise this circuitry over 1000's of cycles.

Since we implemented a unidirectional stress mechanism on the Pentium® 4 processor (i.e, could only delay the strobe generation with respect to its nominal position), we measured the following two points for each SS signal group :
•First Fail (FF): The first signal(s) within a signal group to fail at least one cycle
•All Fail (AF): All signal(s) within a signal group fail for all cycles.

Figure 5 is a simplified representation of a scope measurement showing the "eye" diagram for two consecutive data bits on 3 separate signals synchronized in absolute time (X-axis). The multiple waveforms forming the data valid "eye" represent variations across multiple cycles in the relative position of the data with respect to the strobe due to various sources of differential jitter (i.e., noise on Vcc or Vss , pattern dependencies, or defects). The dotted vertical lines represent the position of the strobes (nominal, shifted to FF, and shifted to AF).

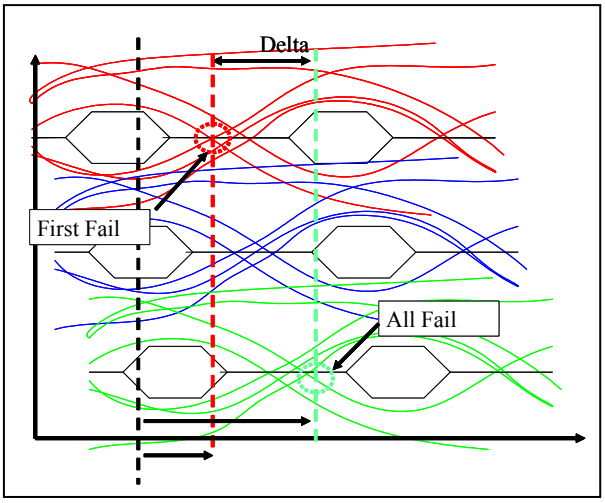

Figure 5: AC IO Loopback Measurements

The "First Fail" (FF) value is the minimum amount the strobe had to be delayed from its nominal position resulting in incorrect data being captured in the input latch for at least one signal of the signal group. This corresponds to the worst case Tva & Th of the signal group. For a centered strobe interface like the Pentium® 4 processor: Expected Delay = 0.5 x Period – (Tva-Th).

The "All Fail" (AF) value is the maximum amount the strobe had to be delayed from its nominal position to result in ALL cycles failing on ALL data signals within the signal group. This corresponds to worst case Tvb & Tsu of the next cycle. Expected Delay = Period – (Tvb-Tsu).

The product specifications for the SS Tsu and Th take into account some amount of shift in the position of the Tsu/Th window corresponding to variations in process skew of the full population of devices to be shipped. So we further improved the accuracy of the screen by using the difference between the FF and AF for each individual device (Delta in figure 5). The Delta corresponds to the maximum width of the Tsu/Th window across the specific signal group of the specific device being tested.

The Pentium® 4 processor stress mechanism consisted of series of delay elements that could be sequentially added into the Strobe Clock generation path. So in its simplest form the FF and AF measurements could be the number of delay elements that were required to be added for each of the measurements. Since this implementation used un-compensated delay elements, the magnitude of the individual delay elements varied about a 100ps over the normal silicon process variation[1]. Since we expected to use several of these elements in each of the measurements, the 100ps uncertainty, per element, exceeded our defect screening accuracy goals. To eliminate the die to die variation, we developed an elaborate algorithm based on over-stressing the loop and changing the IO period to get back to a passing condition to calibrate a delay element down to +/-25ps.

Since some of the signals were being loaded by the tester probes, we incorporated external loops on our Tester Interface Unit to get similar loading on the un-probed signals. Unfortunately this required extra DFT that we missed, and quickly became evident when silicon arrived.

4. Learnings from First Silicon:

The Strobe Signals were included in the set of signals that were unprobed and had external loops. When the first device was tested we learned of an unexpected interaction between the Scan implementation and the external loops. Data was getting out to the signals connected to the strobes and causing the input buffers to no longer function as expected while scanning in data. So the external loops were quickly disabled on the Test Interface Unit (aka Load Board) and DFT was added on the next stepping of the device to ensure unused drivers could be completely disabled during Scan testing.

As part of the design validation, we characterized the size of the delay elements and the with-in die variation between the individual elements on a given delay chain. The data in Table 1 shows that there was significant variation among the cells within a single delay chain. Since our methodology calculated the total delay based on accurately characterizing a single element, this could result in significant measurement

Table 1: Delay Cell Process variation

Single Cell Variation	Within Chain	Across Chains
Avg sigma	21 ps	18 ps
Max sigma	38 ps	33 ps

[1] The temperature and Vcc are fixed and constant during testing, so the variation due to these factors was not significant.

error. Table 1 also shows the variation of the delay elements across the six delay chains[2] implemented on the Pentium® 4 processor was almost equal to the with-in chain variation. Therefore, we increased our accuracy of the delay cell estimate by averaging across the six delay chains. This greatly helped our kill limits, without it we would have had unreasonable yield loss.

When we correlated the Pentium® 4 processor AC IO loopback FF measurements to functional measurements (based on the "Sliding Window" technique), we were a little disappointed. The AC IO loopback measurements showed slightly worse accuracy than the functional measurements. The cause was obvious and we had just overlooked it. While the individual errors were small, we multiplied them in our methodology (+/- 30-40psec on individual delay elements results in 150-200psec errors for 5 elements).

We also validated this methodology on a tester, similar to what is currently used in HVM, that had enough pins to probe the strobe signals directly. We used the tester to measure the magnitude of the stress directly, instead of estimating it from a single element. Since this is a single channel measurement, it relies on linearity and repeatability versus absolute edge placement accuracy. We have found that these types of measurements can be made with greater than 50ps of accuracy on testers that advertise overall tester accuracy (OTA) of 300 psec or worse. Using this methodology, we were able to detect ~ 100 psec "defects" that were inserted (via "defect masks") in a set of Pentium® 4 processor test devices built to characterize this methodology and other test methodologies .

So for this implementation:

Good news: **Functional Tester accuracy without touching the signals,**

Bad news: **Functional Tester accuracy unless you touch the signals.**

The accuracy discussion above is actually an overly pessimistic view as it only applies to the FF delay value. The Delta measurement (AF-FF) was done by varying the period (this was possible because the stress was fixed and did not vary with frequency). This had an accuracy approaching the step size of the tester clock source (<25 psec). Random defects in individual signals within the signal group will show up in this measurement. The absolute FF value is used to screen defects that result in shifts in the strobe(s) from the expected position (centered, in this case). It also understates the functional tester error as the actual HVM methodology for these interfaces on our existing testers had a significantly larger error.

After establishing test limits (based on characterization data and product specifications), we instituted this test in HVM.

As part of our qualification methodology, we collected a sample of rejects from the failures and analyzed them in detail and compared them to the statistics we obtained from data logs from the normal HVM failures. Table 5 summarizes the failure breakdown from the sample and from the HVM data logs.

Table 2: Breakdown of AC IO Failures

	Sample of AC IO failures	Sample %	HVM % of AC IO failures
Basic Lpbk	1	11	25
FF or AF count	4	44	25
Delta > 200ps	0	0	2
DFT fails	3	33	26
Algorithmic/Flakey Fails	1	11	22
Total tested	9		267170

As you can see, the "true" failures are split between basic loopback or gross AC IO loopback failures, meaning their failures were based on delay count only and did not require the time consuming single element calibration mentioned earlier. You can also see there are only a very small number of devices that failed the pico-second based limit in the HVM data (Delta failures), so it was not surprising we didn't have any in our small sample. Based on these results, the lengthy frequency searches in the AC IO Loopback tests were a victim of the Test Time Reduction (TTR) process and were discontinued, leaving in the tests for FF and AF delay element counts.

One final learning related to the test time. The HVM test program was based on the design validation algorithm which used the capture memory on the tester. This resulted in extra fail data but also in a relatively expensive test, longer than the historical HVM functional test time. A more optimized HVM "first fail" version that is being implemented on the

[2] The Pentium® 4 processor has four signal group for the data bus and two for the address bus.

latest microprocessor is estimated to be slightly less than historical functional test time budgets.

5. Evolution and Adaptations of AC IO Loopback

Based on the results from the Pentium® 4 processor, addressing test time was a higher priority than improving accuracy on our current generation of interfaces. The AC IO Loopback measurement process is comprised of the following steps:

1. Setting up the test.
2. Executing the test algorithm
3. Checking the results

Enhancements have been made in each of the steps to reduce test time. For example using single long scan chain to setup the measurements and check the results can take significant time. This is particularly true for products which have a large number of I/O's like our chipsets. The load and unload time can be minimized by separating and optimizing the input and output scan chains. In addition, the time to load the pattern generator can be reduced by enabling a default pattern (toggling, 101010….).

Since there are multiple measurements associated AC IO Loopback the test time can be minimized by automating the execution of the algorithm by implementing it in BIST logic on the die. The BIST circuitry can indicate that the AC IO loopback parameter (FF, AF, or Delta) has been found or it can perform the compare to predetermined values and simply report pass or fail condition. There were several motivations for implementing a BIST solution for AC IO Loopback:

1. Reduce time for generating and debugging test patterns which had previously been manually written;
2. Permit easy Pre-Silicon validation of patterns using RTL simulation;
3. Decrease test time in HVM;
4. The test patterns can be more compressed due to less activity at the pins.

A BIST implementation on a chipset product as been proven in the production test environment and reduced the test time by 10x (as compared to non-BIST solution on previous product)[9].

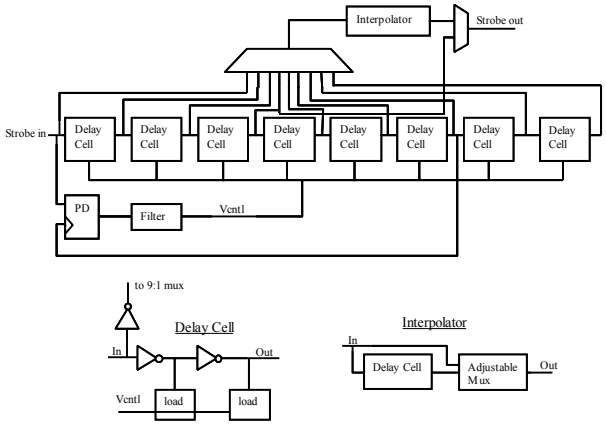

Figure 6: Delay Lock Loop

The inaccuracy in the Pentium® 4 processor implementation resulted from the variation in the delay elements due to process skew and random with-in die variation between the elements. This variation can be controlled by using a Delayed Lock Loop (DLL) as the timing stress mechanism. The simplest description of a delayed lock loop, Figure 6, is a Process, Vcc, & Temperature compensated delay chain. It operates by dividing a time period into an equal number of divisions. A feedback loop compensates the delay increment to achieve this division. Usually there is a coarse stepping of delay buffers and then an interpolator to achieve a finer resolution. Such implementations [10] increase the accuracy of the measurement and because there is no need to estimate the amount of delay, using a DLL reduces test time as there is no need to search for the First Fail point.

From a design perspective, a DLL requires more silicon area, a good portion of which is devoted to a capacitor for filtering power supply noise. Depending upon the interface design, a DLL may be present [11] and it may be just a matter of extending the range to meet the timing stress requirements.

6. Extension of AC IO Loopback technology

As discussed in [12], AC timings are just one part of contactless IO testing. The AC IO loopback methodology has been extended to the other areas, like DC tests. It has also been extended to cover advanced signaling technologies like Simultaneous Bi-

Directional signaling (SBD) and high speed differential signaling.

DC tests may be easily carried out using common full pin count ATEs. However, reduced pin count testing requires new methods to test for DC characteristics, such as Vin and Vout. We can run the IO loopback at low frequency, while we shift the threshold voltage of the receivers to detect the drive voltage of the drivers. We can use the programmable drive strength and the programmable termination to vary the output levels and determine if the receivers are detecting the proper logic state. Since the drivers and receivers are compared with each other (just like the AC IO loopback test), the presence of defects that impact DC specifications can be detected.

Per pin leakage is another DC test that we perform. As described in more detail in [13], the drivers are driven to a logic level while the IO loop is enabled. Once the drivers are switched off, the output is left to float. The leakage of the circuitry associated with the IO will eventually cause the voltage level to drift to the other state. After a pre-calculated time delay (allowing for normal leakage time constant), a sampling of the logic state by the receiver indicate if the circuitry has excessive leakage.

The AC IO loopback technique was also extended for other IO circuit types, e.g. simultaneous bidirectional (SBD) IO circuits as explained in [14], [15]. These interfaces are capable of transmitting and receiving signals at both ends of the signal line (figure 7), i.e. both transmitters at either end of the interface pairs are driving at the same time.

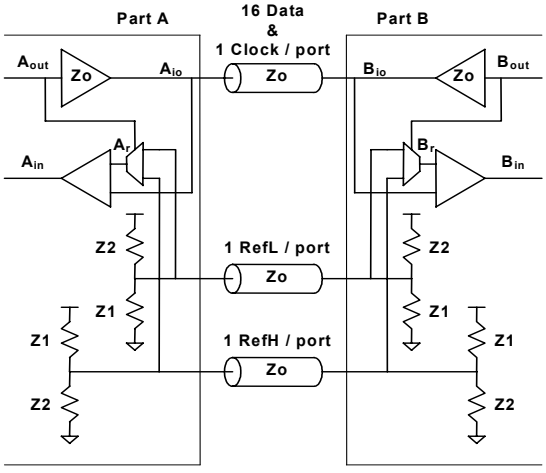

Figure 7: Bidirectional point-to-point connections between two ports

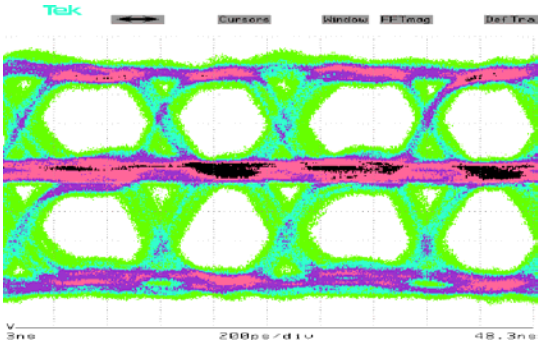

Figure 8: SBD eye diagram at 2.5Gb/s per wire

This actually poses problems for ATE as well. Two channels of resources have to be applied to a single signal pin while juggling with the signal integrity issues. When the signals collide mid way (or at any point of the transmission line), a tertiary level is created (see Oscilloscope trace capture in figure 8). The polarity of the data received has to be extracted from the threshold so that the proper logic level is sampled.

By controlling the threshold voltage of the receiver while changing the delay elements in the AC IO Loopback mechanism, we extracted the true dual loop data eye. Figure 9 represents a single stack of "eye diagrams", similar to what was captured in figure 8. However, figure 9 is what the receiver actually perceived and not what was observed at some intermediate point on the interconnect.

In figure 9, the X axis is picoseconds and the Y-axis is in Volts. The light inner regions are the passing points and the dark outer regions are the failing points. Note, there is a vertical gap between the failing points for the two "eyes" because the failing data for this region was not gathered. The procedure used to create the diagram in figure 9 was akin to an ATE's shmoo plot capability. The timing control was provided by the DFT circuitry, i.e., the delay elements on chip, and the threshold control was provided via a tester (although we are working on circuitry to also provide this on die in next generation implementations).

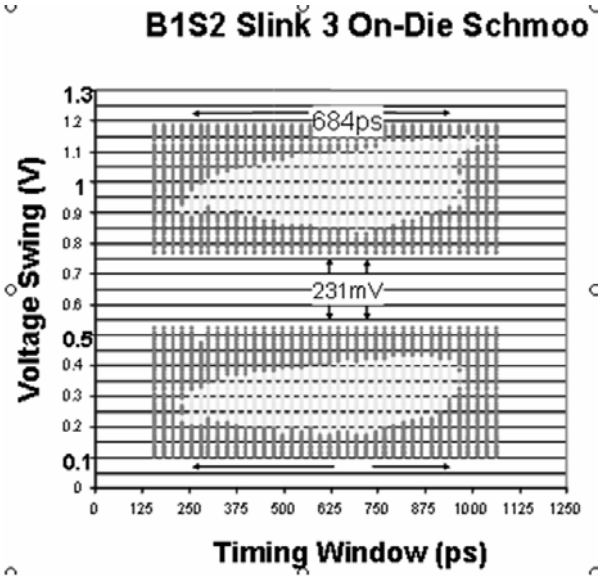

Figure 9: SBD Data eye diagram by shmooing threshold voltage and delay elements

Similar to SBD IO circuits, high speed serial link utilize low voltage, differential and embedded clock/derived clock signaling, running at multiple Gbps (e.g. S-ATA, PCI Express). At this signaling rate measuring the output data eye is a normal part of the test measurement procedure. While this is routinely carried out on oscilloscopes and high-end mixed signal ATEs during characterization, it is not economical to employ manual equipment or expensive ATE for HVM testing of future products that have hundreds of these high speed serial links. With the enhanced AC IOLB DFT circuits that control both the timing and threshold of the receivers, a data eye as observed by the receiver circuit can be extracted without the use of any precision analog resources on the tester.

7. Conclusions

We have described the Intel AC IO loopback test methodology, the Pentium® 4 processor implementation, and HVM experiences. As a result we were left wondering: Was the Pentium® 4 processor implementation over-designed or before its time? Just like delay testing for logic, we believe that the amount of delay testing that is required is inversely related to the margin in the design. The more margin, the less delay fallout will need to be screened. However, as interface performance continues to increase, margins continue to be reduced. In addition, we can't predict when an unexpected interaction between the design and silicon process will occur. So while we have tempered our quest for pico-second accuracy we have continued to perfect this methodology. We described some of our enhancements addressing both the accuracy and test cost issues. We described how we have successfully combined timing stress and voltage stress to generate "eye" diagrams as seen by the actual receiving circuitry. Lastly, we gave an indication of some of the efforts we have underway to develop extensions that can be used for HVM testing of the giga bit per second, serial like, interfaces being developed today.

8. Acknowledgements

This methodology or one of its many permutations has wide spread use within Intel and as such a large number of engineers have contributed to the evolution and learnings touched on in this paper. We appreciate and acknowledge their extra effort to make this happen. We want to specifically acknowledge Wai Yin Chan, Tim Frodsham, Alper Ilkbahar, Ei Jung Chen & Will Kimbro for helping us pioneer this effort on the first microprocessor and chipset.

9. References

1. Intel® Pentium® 4 Processor data sheet: ftp://download.intel.com/design/Pentium4/datashts/24988703.pdf

2. J. Cheng, "When Zero Picoseconds Edge Placement Accuracy is Not Enough", *Proc. Int. Test Conf. 2001,* pg 1134

3. Mike Mayberry, et. al., "Realizing the Benefits of Structural Test for Intel Microprocessors", *Proc. Int. Test Conf. 2002,* pg .

4. Tripp et. al., United States Patent Pending "Method and apparatus to structurally detect random defects that impact AC I/O timings in an Input/Output buffer"

5. United States Patent 5,621,739, Sine , et al. April 15, 1997 'Method and apparatus for buffer self-test and characterization'

6. P. Gillis et al, "Delay Test of Chip I/Os using LSSD Boundary Scan", *Proc. Int. Test Conf. 1998,* pp. 83-90

7. United States Patent: 6,477,674, Bates et al. Nov. 5, 2002 'Method and apparatus for conducting

input/output loop back tests using a local pattern generator and delay elements'

8. United States Patent 5,869,983 Ilkbahar , et al. February 9, 1999 'Method and apparatus for controlling compensated buffers'

9. United States Patent Application 20030005374, Fought, et al. January 2, 2003, "Method and apparatus for testing an I/O buffer"

10. H. Muljono, et.al., "A 400MT/s 6.4GB/s Multiprocessor Bus Interface", ISSCC paper, Feb 2003

11. United States Patent 6,348,826, Mooney , et al., February 19, 2002 "Digital variable-delay circuit having voltage-mixing interpolator and methods of testing input/output buffers using same"

12. S. Sunter*, B. Nadeau-Dostie, "Complete, Contactless I/O Testing—Reaching the Boundary in Minimizing Digital IC Testing Cost", *Proc. Int. Test Conf. 2002,* pg 446

13. T.Rahal-Arabi et al, "A JTAG Based AC Leakage Self Test", Proc. of VLSI Circuits Symposium, paper 18.4, June 2001

14. Haycock, et al. A 2.5Gb/s Bidirectional Signaling Technology, Hot Interconnects Symposium 5, 1997

15. Haycock , et al. Apparatus and methods for testing simultaneous bi-directional I/O circuits, United States Patent 6,348,811 February 19, 2002