DATA MINING ANALYTICS FOR SERDES HSIO VALIDATION

MOVING BEYOND PASS/FAIL

BY STEPHANIE AKIMOFF

ASSET

Stephanie Akimoff

Stephanie Akimoff, senior application engineer, is currently the worldwide technical lead for validation and customer issues for the development team working on ASSET InterTech's HSIO design validation tool for the ScanWorks® platform for embedded instruments. Previous to ASSET, she worked for Intel® Corporation for seven years. Stephanie is a published author and well respected in the field of signal integrity. She graduated from Pacific Lutheran University with a double major in electrical engineering and computer engineering.

Table of Contents

Table of Figures

Table of Tables

Executive Summary

Following best practices to validate serdes high-speed I/O (HSIO) on motherboards can sometimes create unexpected problems for engineers. Many engineers – in particular, signal integrity engineers, I/O validation engineers and production engineers – have employed the testing methods recommended by chip suppliers, but they end up with a mountain of data that holds little meaning for them. And all that data is difficult to analyze. In fact, the sheer volume of the data collected can become part of the problem. Too often, engineers settle for a cursory pass/fail judgement call on the validity of the HSIO being tested with a single test or over a small sample size instead of performing statistical analysis and data mining on test results (multiple tests on multiple boards). Data mining can reveal which test configurations should be tested to validate a platform and which test configurations don't require testing. This reduces validation testing, accelerates development and delivers new platforms to market sooner.

Validation engineers at system suppliers are challenged by chip vendors who recommend comparative statistical validation testing on HSIO operating margins, as well as alternative components, plug-in boards and many other aspects of a design. In development alone, for example, data must be gathered over multiple prototype motherboards assembled on multiple printed circuit boards (PCB). The intent is to validate that the operating margins on HSIO buses is sufficient so that there will not be a risk of failure or significant performance degradation as the design progresses through the various phases of development and assembly. Later, beyond design and development, these same tests or a subset of them are re-used to gather additional data and revalidate the design during the later phases of the motherboard's life cycle. Gathering the data generated throughout this process is not a problem. Not performing data mining is a major problem that can wreak havoc on engineering timelines, sometimes delaying a product's delivery to the marketplace or allowing a design flaw on production boards.

Not performing data mining is a major problem that can wreak havoc on engineering timelines, sometimes delaying a product's delivery to the marketplace

From the perspective of signal integrity engineers, I/O or memory validation engineers, and production engineers, this eBook explains that data mining and the application of analytics to a validation test database gathered over the life cycle of a system will generate significant value in

terms of cost and time savings for the system supplier. With the right suite of analytic tools at their disposal, engineers are able to identify trends, quantify risk factors and make decisions to ensure the financial success of a product. This eBook examines the foundational steps engineers can take early in the validation process so that effective data mining and analytics can take place throughout all phases of the life cycle of the system.

The Shortcomings of Pass/Fail Validation Testing

For several years, chip makers have pointed out the shortcomings of overly simplistic pass/fail testing by recommending that various statistical methods should be applied to fully validate HSIO operation and performance on motherboards. There are many good reasons for this.

A single high-speed I/O pass/fail test on one platform performed only one time does not give an appropriate level of understanding of the signal's quality. A board could ostensibly pass a single validation test, but actually be approaching perilously close to failure. Passing one test is a simple snapshot, predicated on the conditions present when the test was run. It is possible for a platform to pass the predefined mask on one test, but fail on subsequent tests with the same conditions due to reflections, crosstalk, ISI, power integrity issues and other challenges. Running the same test on multiple boards would uncover these issues.

The database of information resulting from multiple tests over several platforms will give a much clearer indication of the confidence designers can have in how the design will perform over time. Building a baseline database of information has the added benefit of being utilized over the product life cycle by providing a foundation to compare over time.

Statistical Validation

Statistical methods allow engineers to evaluate risk factors associated with various HSIO operational metrics. For example, intermittent low margins found as a result of additional testing might not decrease HSIO performance to the point where it is unacceptable. A common terminology used to describe the methodology of running multiple tests on multiple boards is "NxN" testing, where "N" is the number of times the same test is run per board across "N" number of boards. For example, 5x5 means running the test five times on five different boards, producing 25 test results. Statistical methods will quantify the risk associated with manufacturing a design with HSIOs. With this information the manufacturer could decide to either manufacture the design, assuming the risk factors are low, or re-spin the board to correct the flaw(s) found during validation. Correcting major design flaws during development is typically far less costly than manufacturing a flawed design that jeopardizes the success of the product. Simple pass/fail x testing which only produces a test sample size of one would not quantify the risk factors associated with the design the way that a 5x5 test method would because the 5x5 method produces a sample size of 25 tests.

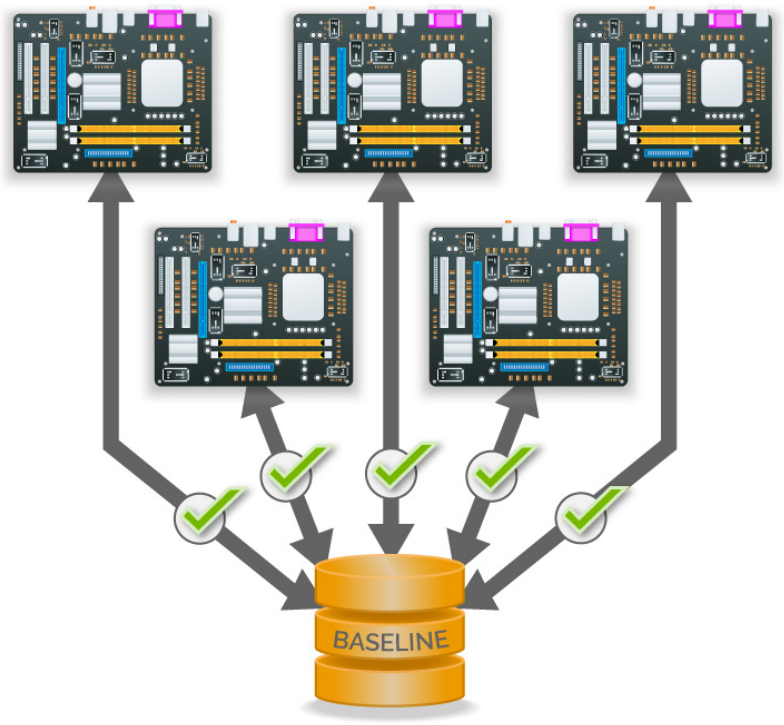


Figure 1: Statistical N x N testing is done over multiple circuit boards assembled by multiple fabricators.

The speed and sensitivity of HSIO have reached a point where effective HSIO validation can only be performed by statistical analysis using an appropriate sample size (Figure 1). In addition, each individual route of a bus on a motherboard and each lane and port on each bus should be

tested. Finally, each test should be performed multiple times. The objective of all this testing is to reveal the quality of all of the HSIOs on the motherboard in order to understand the risk of failure and/or degraded performance for each bus.

Running one test across one board (1x1 testing) is not good enough for validation. There is so much more information that could be discovered with statistical validation that will not be discovered by passing one eye mask test one time. Using a 1x1 test method, the test results are only good for just that one instance, whereas if a proper NxN test suite were completed under the proper conditions, an invaluable database would be provided for the entire life cycle of the platform.

Who's Testing What and When?

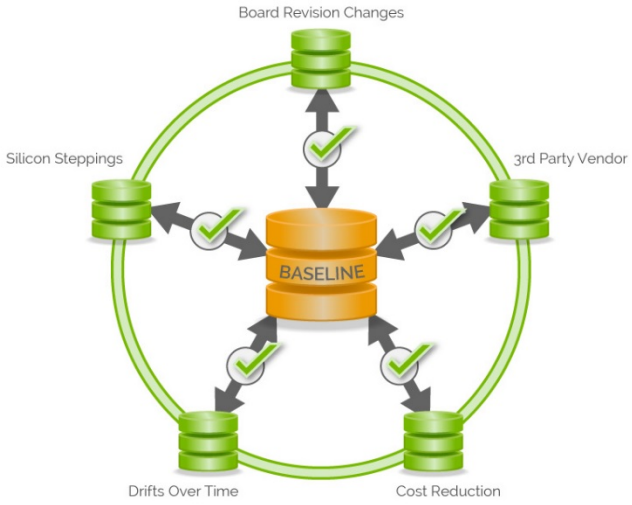


Figure 2: Any of several different events can trigger another round of validation testing on a design.

Any of several distinct events, such as design revisions, a new version of a processor, different memory configuration, changing an I/O device or firmware, changes in the equalization settings etc., can trigger validation testing on a motherboard design over the course of its life cycle (Figure 2). Typically, this testing is performed by at least three types of engineers: signal integrity engineers, I/O or memory validation engineers, and production engineers. During a validation project, each type of engineer will have his or her own set of concerns that will occupy the majority of their time and attention. As a result, each type of engineer will be challenged in a different way by the volume of data generated by statistical validation.

Signal Integrity Engineers

During development, signal integrity engineers will validate the signal integrity on HSIO buses on a relatively small number of manufactured prototypes of a motherboard design before it moves into the next phases of testing, typically I/O or memory validation, which in turn is followed by more validation testing by the production team (Figure 3). In addition, signal integrity engineers will analyze the signaling margin relative to the eye mask on some or all of the lanes that make up the HSIOs on the board.

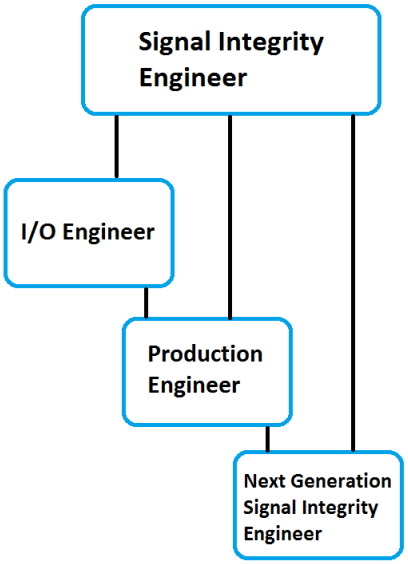


Figure 3: Validation testing is performed throughout a system's life cycle

The eye mask margin analysis (Figure 4) is important to signal integrity engineers because the margin between acceptable and unacceptable performance will tend to decrease as the design moves through the different phases of development as well as later throughout the product's life cycle. Manufacturing will likely not implement the same components on an assembled motherboard as those that were implemented on the prototypes that were used during design validation. Prototype boards generally have greater margins due to the higher quality components than the components used during production. Components employed during production are usually inferior to those that designers specify for prototypes. This may cause the margin relative to the eye mask to decrease from that which was found on prototypes.

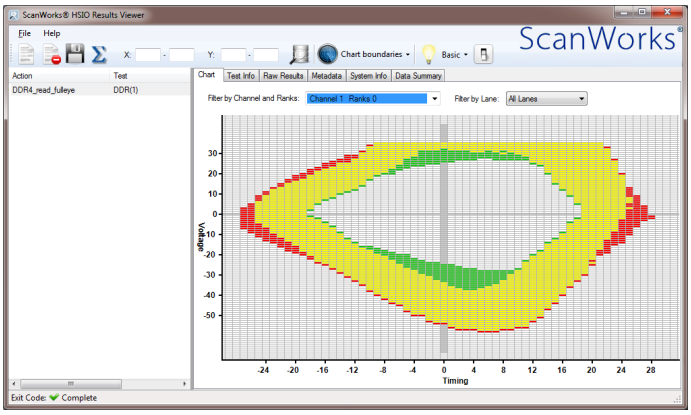


Figure 4: An eye mask analysis in a validation test results viewer

Moreover, the board will be subject to a much wider spectrum of environmental conditions in user installations than it experienced in the controlled environment of the design lab. These and other factors will erode the margin to the HSIO's eye mask throughout the motherboard's life

cycles. Knowing this, many engineers typically require that the signal margin on the board's HSIOs found in the design lab must be sufficiently wide as to accommodate some erosion of the signal margin as the motherboard progresses through the various phases of its life cycle. Therefore, the signal margin for a motherboard that is operational in the field will likely be significantly less than the signal margin on a design prototype in the validation lab.

As a first step toward validating the HSIOs on a board design, signal integrity engineers should establish a series of baseline validation measurements. These baselines will be used as the basis of comparison for all test data generated by subsequent validation testing. Deviations from the baseline measurements can have a number of effects, which differ depending on the type of engineer evaluating the information. For example, certain third-party vendor part numbers might not be approved for a platform due to low margins, the equalization settings might need to be adjusted to a sweet spot, etc.

Validation tests should be performed utilizing the devices that will be sold with the system. Once these devices are known, a complete set should be set aside and marked with the firmware, revision and any other pertinent information at the time the tests took place, since these factors could also affect operating margins.

Performing eye mask margin analysis on all lanes on all HSIOs using statistical test procedures generates a huge amount of data that must be analyzed by engineers. Following the validation work performed by the signal integrity engineer, I/O or memory validation engineers, and production engineers also perform validation testing. With a validation data mining tool at their disposal, I/O and memory validation engineers, as well as production engineers are able to take advantage of the validation testing already performed by signal integrity engineers and, as a consequence, reduce the number and time of the validation tests they perform, streamlining the entire process and accelerating the product's time-to-market.

I/O and Memory Validation Engineers

I/O validation and memory validation engineers are typically charged with evaluating the effects that third-party plug-in modules will have on the performance of a motherboard before it moves into manufacturing. The intent is to identify any issues that, for example, a certain device or a plug-in module, such as a dual inline memory module (DIMM), might cause when connected to

one of the platform's HSIO buses. Plug-in modules or devices that maintain sufficient margin at the eye mask could be approved for manufacturing or recommended by the system supplier to users. But, if a device were to significantly decrease the margins at an HSIO's eye mask, the system supplier might eliminate that part number from the platform's list of approved suppliers.

The I/O or memory validation engineer begins by examining the validation data compiled by the signal integrity engineer. In addition to data mining and re-using the validation test plan created by the signal integrity engineer, the I/O validation engineer will also conduct additional tests on the devices he or she is validating. These tests will generate another wave of validation data which must be analyzed on its own. The performance of alternative I/O devices and their effects on the rest of the platform's operation are then compared with their golden device counterparts. For example, a memory validation engineer might eliminate a specific DIMM's part number from a certain supplier because it did not meet the margin criteria. The signal integrity, I/O and validation engineers all use the same NxN methodology for validation testing.

Production Engineers

Before a board design migrates into volume production, production engineers also perform validation tests to determine which alternative components and/or materials could be used in manufacturing without significantly decreasing the eye mask margins. Often, these alternative components and materials are less costly than those employed by the design department, so production engineers must validate all alternatives to ensure that they do not jeopardize the performance and functionality of the board. Production engineers could qualify several devices from different suppliers for the volume manufacturing of a platform.

Ideally, the production engineer would be able to mine the validation data and test plan generated earlier by signal integrity, and I/O or memory validation engineers. Production engineers may re-use some portion or the entire validation database previously compiled and they will add new data to this database as they perform validation tests on alternative components and materials.

In addition, production engineers periodically validate a sample of production boards to ensure that variations in manufacturing processes have not caused unacceptable margins to the eye masks on the HSIOs. Over time, the performance of components on production boards can drift

out of their acceptable range, causing the platform itself to drift out of range. These periodic validation tests will also add data to the database associated with a platform design.

Mining Validation Test Data

Data only becomes valuable when it can be analyzed. In the case of validation testing, analyzing and mining the data compiled early in the life cycle of the product can actually reduce validation test costs throughout the entirety of the product's life.

Turning a vast database of validation data compiled and collected over all of the phases of a system design into useful information is quite a challenge unless the database can be mined with analytic tools.

Turning a vast database of validation data compiled and collected over all of the phases of a system design into useful information is quite a challenge unless the database can be mined with analytic tools. Collecting data is just the first step. Retaining all of the validation data collected by signal integrity engineers, I/O validation engineers and production engineers and then performing analytics on it can yield many process and economic benefits to the system supplier. For example, all of the validation tests applied by signal integrity engineers on a design early in development need not be repeated later. The signal integrity engineer will discover the test configuration that exercises the most critical aspects of the design and places the most stress on certain aspects of the system. These critical test configurations might be repeated later by I/O validation and production engineers while other test configurations that do not provide insight into critical aspects of the system need not be repeated. In addition, as more validation data is added to the database, the number of design re-spins for subsequent designs can be reduced because developers will learn which design practices are most effective and which should be avoided. With a rich database to draw on, the time it takes to validate successive generations of a design or closely related designs can be reduced significantly.

What to test?

Before a validation process can begin, engineers must decide how much or how little of a design will be validated. The signal integrity engineer typically sets up the validation process for the first generation of a design. He or she will have analyzed the design and, for example, will know

the longest and shortest lanes of the bus that is being considered for validation testing. He/she should also be aware of any glaringly obvious routing issues that might be in question. Examples of these issues would be HSIO lanes that are longer or shorter than others, lanes that have been routed too close to other signals, lanes routed through a noisy power plane, inadequate ground vias near the lanes and other issues that will affect signal margins.

In addition, most new product development projects cannot completely follow the reference design guide provided by the silicon supplier. The question for the new product designer is where can the design guide's rules be compromised and by how much? Tools are available that will give the signal integrity engineer this information. The amount of validation testing should scale upward to the degree to which the rules in the design guide have been compromised. If the design guide was followed explicitly, minimal validation of the margins on the HSIOs may be acceptable. But, if the rules have been violated in several places, more validation needs to be done. Since the evaluation is accomplished through statistical analysis, multiple tests must be performed so that the sample size is large enough to arrive at a high confidence level in the statistical analysis.

Since testing is accomplished through statistical analysis, multiple tests must be performed so that the sample size is large enough to arrive at a high level of confidence.

Establishing baselines

As the first step toward validating the HSIOs on a board design, engineers should establish a series of baseline measurements. These baselines will be used as the basis of comparison for all data generated by subsequent validation testing. Deviations from the baseline measurements can have a number of effects, which differ depending on the type of engineer evaluating the information. For example, the signal integrity engineer might decide that design changes are needed or not needed. I/O and memory validation engineers might eliminate a specific device. And a production engineer could qualify several devices from different suppliers. Choosing the test equipment should be dependent upon the components the system supplier will be selling with the product. A set of these devices should be set aside and marked with the firmware, revision and any other pertinent information that might affect margins. Multiple components might need to be set aside to maximize the amount of stress on the bus.

One of the critical tasks of the signal integrity engineer is defining the 'golden system' parameters. The golden system is defined as the baseline against which all subsequent signal integrity measurements will be compared. This golden system definition should be as detailed as possible to assure apples-to-apples comparisons. This would include BIOS version, board SKU, third-party add-in card vendor, part number etc. This definition should be as close to the production platform as possible.

Another critical task of the signal integrity engineer is to create a test plan such that other teams can follow the same validation steps in testing. This includes finding a test configuration that is the most stressful for the platform being validated. It is best to find one configuration so that the tests can be run in both the read and write directions when put into a sequence and there is less handling of the hardware. Both the golden system and the most stressful configuration are part of the test plan for the different buses that should be reused by subsequent validation teams.

Beyond Pass/Fail

To find the platform configurations that results in the most stress, validation engineers must apply statistical methods to a database of test results of sufficient size to provide a statistically valid conclusion.

Historically, validation engineers did not have the statistical tools they needed to see beyond the scope of a single configuration of the platform being tested. Then came statistical tools using the NxN test methodology. Unfortunately, without data mining, the amount of data derived from such testing is substantial and analysis of it cumbersome. Simply examining whether the HSIO buses pass or fail a test suite is not statistically predictive of operating performance over time. Without analytic tools, the only way to compare test result files has been to look at them one result at a time. With this methodology, some important information can be missed. Viewing the individual data, however, is still the best way to debug any issues. For debugging purposes, more granular test results are better. For data mining, the more data that can be seen at the same time, the easier it is to see trends.

One of the first goals of validation engineers is to identify the test scenario or the configuration of the platform that results in placing the most stress on a particular HSIO bus because this will

show the design team where the operating margins are at a minimum level. The less margin there is, the more likely a HSIO could have a data miscompare. But, in order to find the platform configurations that results in the most stress, the validation engineers must apply statistical methods to a database of test results of sufficient size to provide a statistically valid conclusion. By identifying early in the design phase the platform configurations that result in placing the most stress on a HSIO, successive validation testing by I/O validation engineers and production engineers can focus on the test configurations that are most vulnerable to failures.

Use Cases: The Limitations of Pass/Fail Validation

The following use cases illustrate the limitations of pass/fail validation testing. The example cases use six different configurations of the platform to determine which configurations place the most stress on a particular HSIO. Each configuration used the same set of DIMMs and the same unit under test (UUT). The parameters examined were: + Voltage, - Voltage, + Timing and – Timing. Each configuration had a different effect on these four parameters. This process will identify which test configuration is needed for the margin parameter. It will also show that one test configuration is not best for each parameter.

The same tests were performed on two different platforms to determine whether the stress placed on the HSIO bus is specific to a particular platform or whether it is caused by the configuration of the platform. Both platforms contained one SoC CPU. The HSIO bus validated was the DDR memory bus. Platform 1 consists of two channels of DDR3 memory while Platform 2 consists of four channels of DDR4 memory. Both platforms were tested with DIMMs from the same memory vendor.

Platform 1: Test Case 1

Test Case 1 contains one DDR3 DIMM on one channel. This is the most common test case. This configuration produces the best-case test results and, as a result, generally should not be used in validation testing. Tests were performed in the read direction and results were last pass. Table 1 shows the results of one test. They indicate the number of steps that were tested while incrementing/decrementing the voltage/timing parameter. The test starts at the trained value for the bus and then progresses one step at a time until it reaches a data miscompare. The step size is dependent upon the bus, the direction of the test, voltage or timing of the bus.

Table 1: Test Case 1 Results

	+ Voltage	- Voltage	+ Timing	- Timing
Channel 0	20	-22	17	-19

The pass/fail results are also shown in a results viewer below (Figure 5). Timing is on the X or horizontal axis and voltage on the Y or vertical axis. The colors on the X and Y axis indicate the following:

- **Green**: all lanes passed
- **Yellow**: at least one lane failed while the rest of the lanes passed
- **Red**: all lanes failed

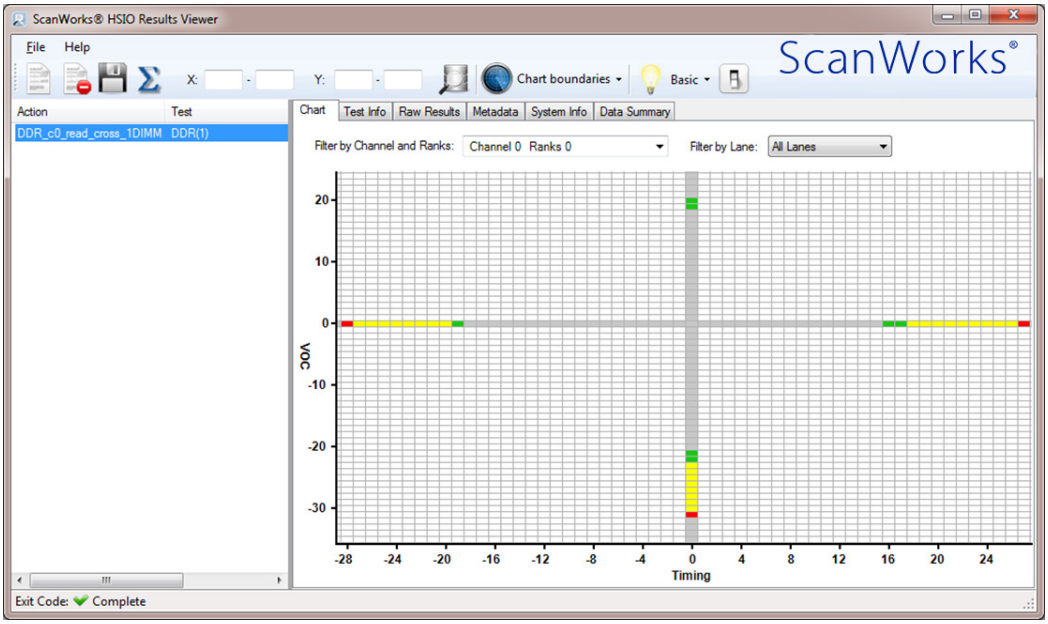


Figure 5: Test Case 1 pass/fail results shown in a results viewer.

Platform 1: Test Case 2

The platform in Test Case 2 is configured with two DDR3 DIMMs, one DIMM per channel (1DPC) and two channels populated. DIMM 1 is in Channel 0 and DIMM 2 is in Channel 1. The tests were performed in the read direction. Results are last pass. Table 2 shows the test results from one test.

Table 2: Test Case 2 Results

	+ Voltage	- Voltage	+ Timing	- Timing
channel 0	20	-22	17	-20
channel 1	20	-20	17	-19

The pass/fail results are shown in a results viewer below (Figure 6). The first screen capture shows results gathered from Channel 0, while the second screen shows results from Channel 1.

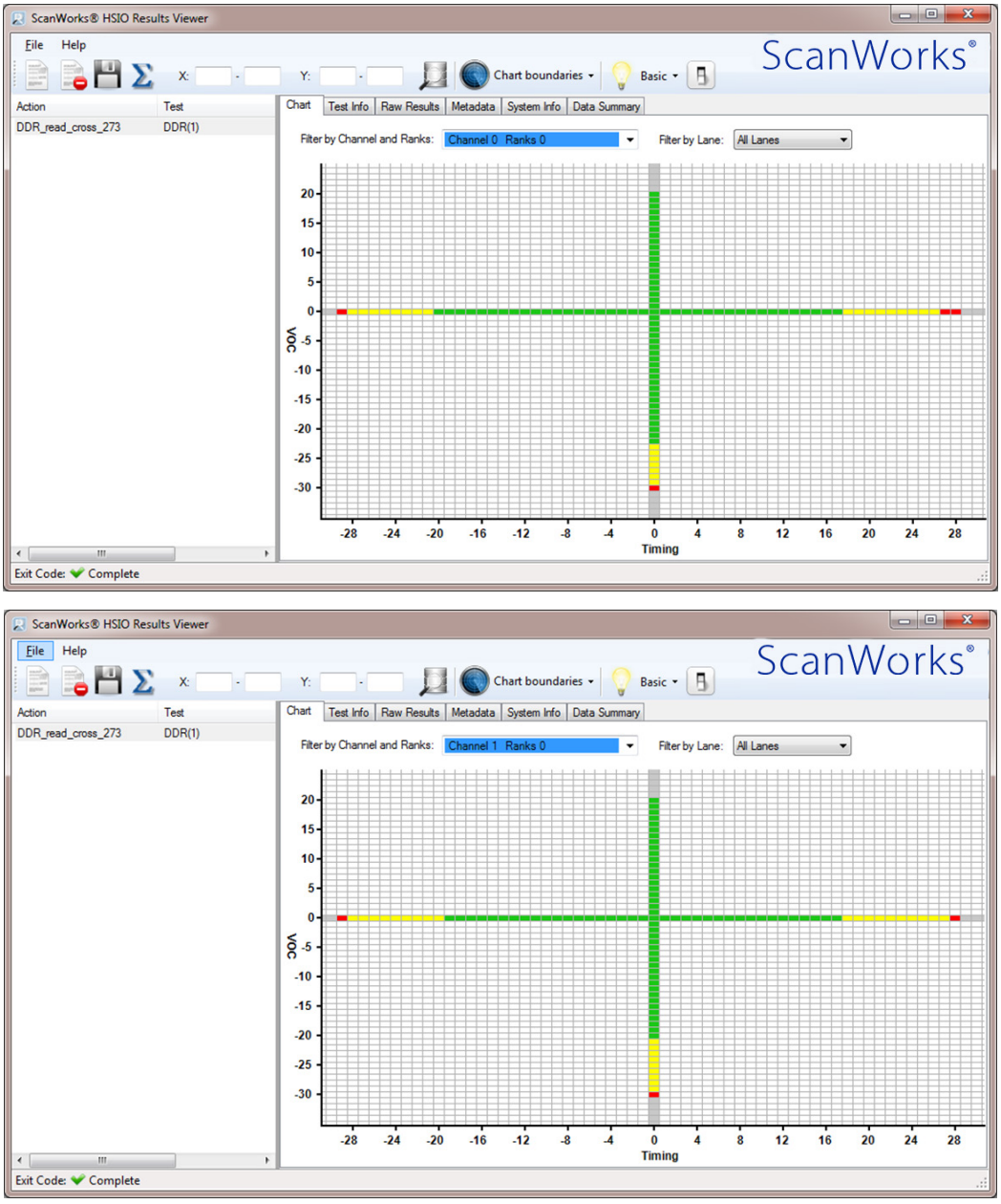


Figure 6: Test Case 2 results on two channels

Platform 1: Test Case 3

The platform in Test Case 3 is configured with two DDR3 DIMMs, two DIMMs per channel (2DPC), and one channel populated. The tests were performed in the read direction. The results are last pass. The table below shows the test results from one test.

Table 3: Test Case 3 Results

	+ Voltage	- Voltage	+ Timing	- Timing
channel 0	20	-17	12	-18

These pass/fail results are also shown in a results viewer below (Figure 7).

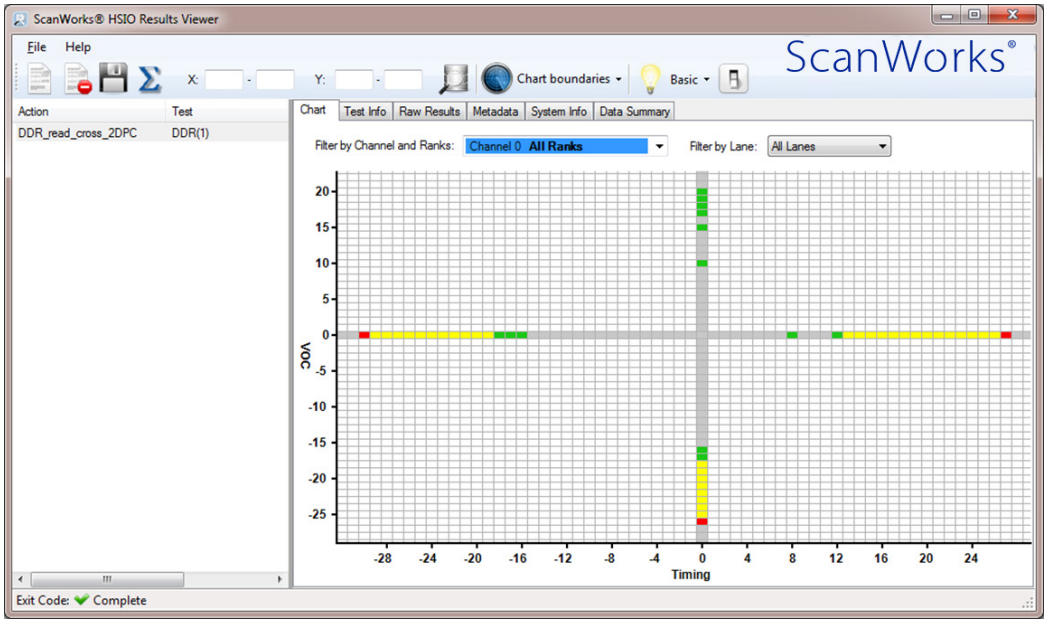


Figure 7: Test Case 3 pass/fail results

Table 4 summarizes the results of all three test cases. The top line is the default eye mask. All results should be greater than this value. Comparing all three test results, one can clearly see that the amount of margin stays relatively consistent on two of the parameters: +Voltage and –Timing. The +Voltage results are all 20. The –Timing results range is -18 to -20. The –Voltage showed a 23% decrease in margin from Test 1 to Test 3. Test 1 had a result of -22 and Test 3 had a result of -17. The +Timing shows a significant 30% reduction in margin from Test 1 to Test 3. Tests 1 and 2 had a result of 17, while Test 3 had a result of 12. Using the traditional pass/fail method of testing, this reduction in margin can be easily missed because the validation engineer would either look to see whether the tests passed or failed the eye mask or only examine the results of one pass/fail test type. If the wrong test type for the platform was performed, this decrease in margin would be missed.

Table 4: Summarized Results of the Three Cases

+ Voltage	- Voltage	+ Timing	- Timing	
13	**-13**	**10**	**-10**	Eye Mask
20	-22	17	-19	Test 1- 1DPC ch0
20	-22	17	-20	Test 2- 1DPC ch0
20	-20	17	-19	Test 2- 1DPC ch1
20	**-17**	**12**	-18	Test 3- 2DPC ch0

The validation engineer might want to duplicate the same tests outlined above on a different platform. Testing this second platform would show whether the test results from Platform 1 are limited only to Platform 1 or are consistent across a second platform.

Use Cases: The Benefits of Data Mining

Unlike the pass/fail test methodology shown above, which compares only one test result on three different platform configurations, statistical data mining enables a thorough analysis of more parameters than just voltage and timing. Now the range of the test results for each parameter can be examined. The range will tell the engineer whether the results are consistent across multiple tests. If the range is widely dispersed, such as greater than five margin steps, then this indicates that there might be jitter, reflections, power issues, pattern dependencies etc. on the bus. One way to see the worst case would be to increase the time spent at each margin point. This is known as the dwell time. Increasing the dwell time would increase the bit error rate (BER) for each margin point. This should narrow the range across multiple tests and increase the confidence level of each margin point. This should give worst case margins.

The results of multiple tests are best viewed with a statistical data mining tool. The screen captures (Figure 8) were taken from the HSIO Validation Assistant (HVA) feature that is part of ASSET InterTech's ScanWorks® HSIO tool. A '5 x 5' test suite (i.e. five tests on five different systems) was performed on the same Platform 2 defined above and by using the same test case configurations described in the previously discussed Platform 1 Test Cases 1-3. The first set of tests was performed in the read direction. The test case configurations were:

- 1DIMM in channel 0
- 1DPC in channel 0 and channel 1 (two DIMMs in the system)
- 2DPC in channel 0.

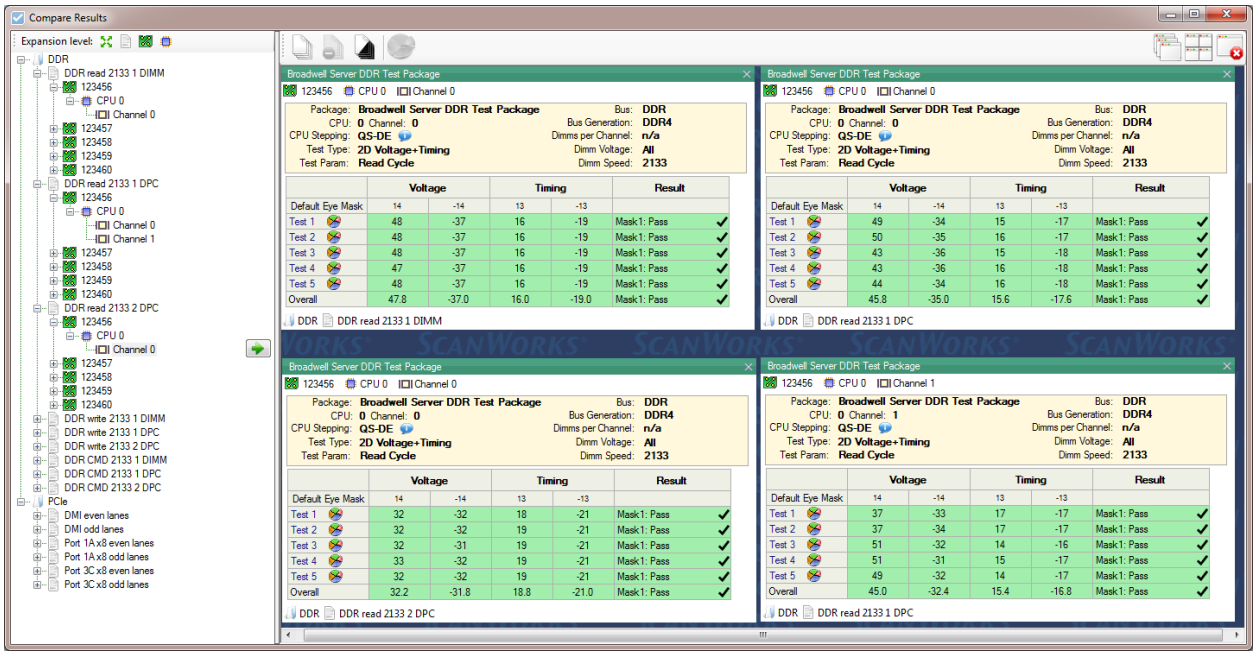


Figure 8: Platform 2 results for all test cases in the read direction

The results shown in the HVA tool are for one system on Platform 2. The upper left panel shows Test Case 1 (one DIMM in Channel 0), the lower left panel shows results for Test Case 3 (2 DPC in Channel 0). The two panels on the right of the viewer screen show test results for Test Case 2. The upper right panel shows 1 DPC in Channel 0 and the lower right shows 1 DPC in Channel 1.

The range across all five tests for Test Case 1 +Voltage was 47-48. This is a range of one margin value across five tests. The five test results for –Voltage, +Timing, and –Timing were the same across all tests. The results for all tests in Test Case 3 were also consistent across all four parameters with a change of only one margin value for voltage across all five tests. In other words, the results are tightly grouped.

In contrast, the test results for Test Case 2 in the right two panels show that the range for +Voltage is very wide. For Channel 0 the range is from 43 to 50 margin clicks; for Channel 1 the range of 37 to 51 is even greater or 27%!

To investigate the wide range of Test Case 2 (1DPC) voltage test results, the validation engineer has multiple options. The engineer could increase the dwell time of the test, increase the number of systems tested, increase the number of tests per board, etc. Unfortunately, these solutions will increase the amount of time needed to complete the validation. The original test methodology

was a 5 x 5 test suite. Any increase in test time on a single test for each of the 25 tests that will be performed would increase the resulting test time by 25x. Exploring the results of these options is beyond the scope of this paper.

At this point, validation engineers might rethink their options. Rather than increasing the test time, a dominant parameter might be sought to determine if there is a better test configuration. For example, voltage or timing might be the dominant factor. Looking at the entire set of test results shown in the HVA tool, the Test Case 3 result shows the least margin for the voltage parameters. The eye mask is +14 and -14. The individual test results with the least amount of margin for voltage are +32 and -31. This is +18 and -17 steps above the eye mask. Both the Test Case 3 and Test Case 2 results have a minimum margin point of -31 for –Voltage. While the minimum voltage for +Voltage is less for the Test Case 3 test configuration, it is still +18 steps above the eye mask. The timing margin results are much narrower in both directions. Therefore, it is the dominant factor in determining which configuration to use in validation testing. For Test Case 3, the timing test results with the least margins were +18 and -21. The eye mask for timing is +13/-13. This gives a margin above the eye masks of +5 and -8.

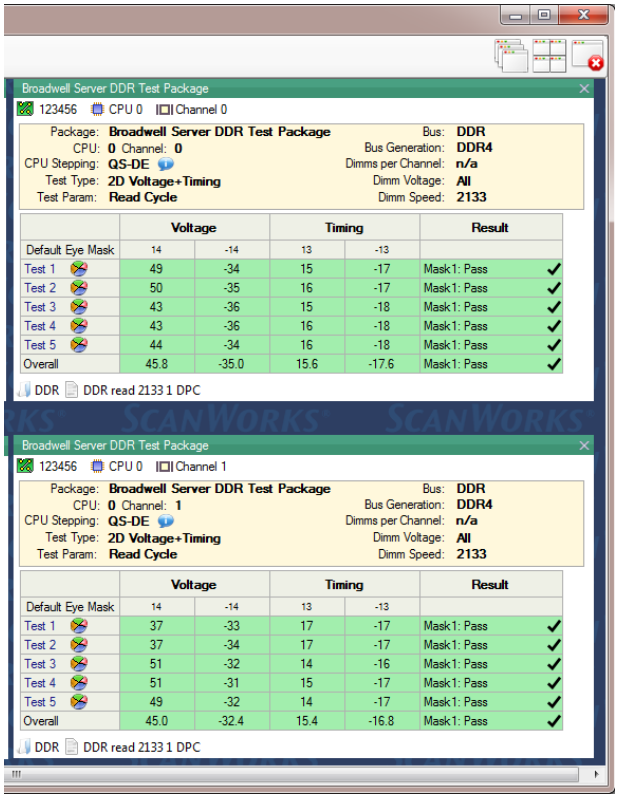


Figure 9: Statistical validation test results configured with Test Case 2

For Test Case 2, in Figure 9 for the sake of clarity, the amount of margin above the eye masks on Channel 0 are +2 and -4. For Channel 1, the amount of margin above the eye mask is just one margin step for +Timing and -3 steps for –Timing. Therefore, Test Case 2 should be considered for validation testing. Unfortunately, this is the configuration that has the wide variance in voltage testing.

Recall for Platform 1, using just the single 1 x 1 pass/fail test method, the best test configuration for validation testing was the Test Case 3 configuration. The display below (Figure 10) examines how tight the range is for all the parameters. Notice all the results are very tight for all test results in a test configuration. In looking at the timing parameters, it still looks as though the Test Case 3 is the best test configuration for validation testing for this platform.

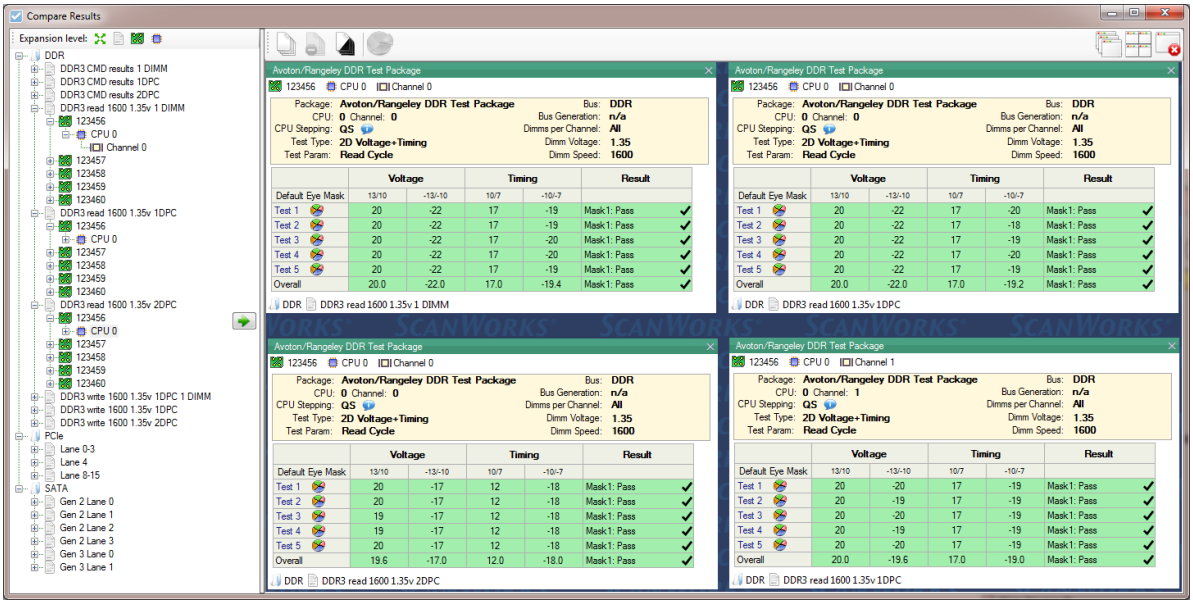


Figure 10: Platform 1 all test case results in the read direction

What these results show is that in the read direction, the two different platforms tested need two different test configurations. The goal is to find one configuration for both the read and the write directions on one platform, not to see if there is one configuration that works best for all platforms. The validation engineer is interested in finding one configuration so it can be used to test both the read and write directions on the platform being tested. What this proves is that each platform reacts differently to the test configurations. The engineer needs to determine which test configuration is best for the platform.

Further tests in the write direction will show which configuration for the different platforms should be used by all stages of validation, including signal integrity engineering, IO and memory validation, and production engineering. A tremendous amount of test time can be eliminated by identifying through data mining the platform configurations most appropriate for subsequent validation testing and by eliminating those configurations that do not yield the worst-case eyes.

The results in Figure 11 are for Platform 1 in the write direction using the identical configurations used in the read direction. The Test Case 1 shows the least amount of margin in the +Timing direction at 16 steps. This is 6 steps of margin above the eye mask. The Test Case 3 +Timing worst case margin is 18 steps or 8 steps above the eye mask. There are two margin steps separating the 1 DPC and Test Case 3 test results. These results are very close and within the normal range of up to five margin steps that was discussed earlier. Compare that to the discussion above where the +Timing margin for Platform 1's Test Case 3 configuration was just 2 steps above the mask. This is the worst case eye margin. If the goal is to find one configuration to use for both directions of testing in all stages of validation, then the Test Case 3 configuration is clearly the configuration to use for Platform 1.

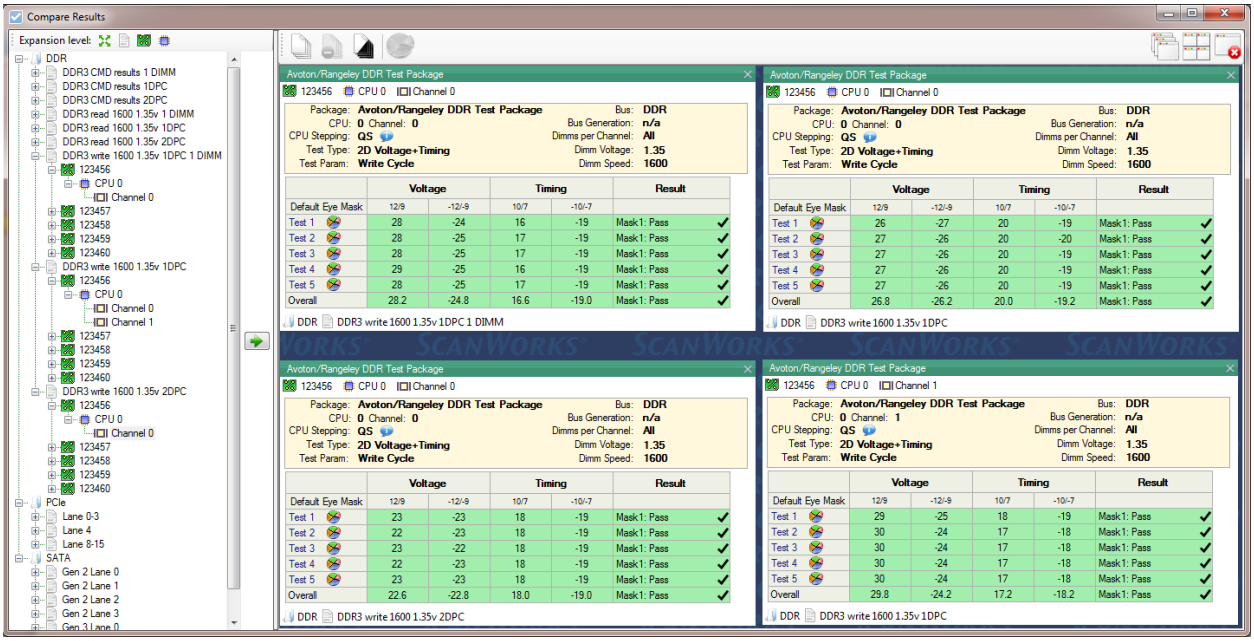


Figure 11: Platform 1 all test case results in the write direction

The results in Figure 12 are for Platform 2 in the write direction using the identical configurations used in the read direction. Notice that the ranges are tight in the write direction for the Test Case 2 configuration as opposed to the wide range seen in the read direction with the same configuration. If the engineer is only concerned with pass/fail, the results pass. But if the engineer is concerned with the risks associated with the size of the margin at the eye mask, then there is cause for some concern as the test only passes the –Voltage margin by one step. For different reasons, the Test Case 2 configuration is the best test configuration for this platform.

Fortunately, because the test results are very close, there is no need in the write direction to take extra steps to ensure the margins.

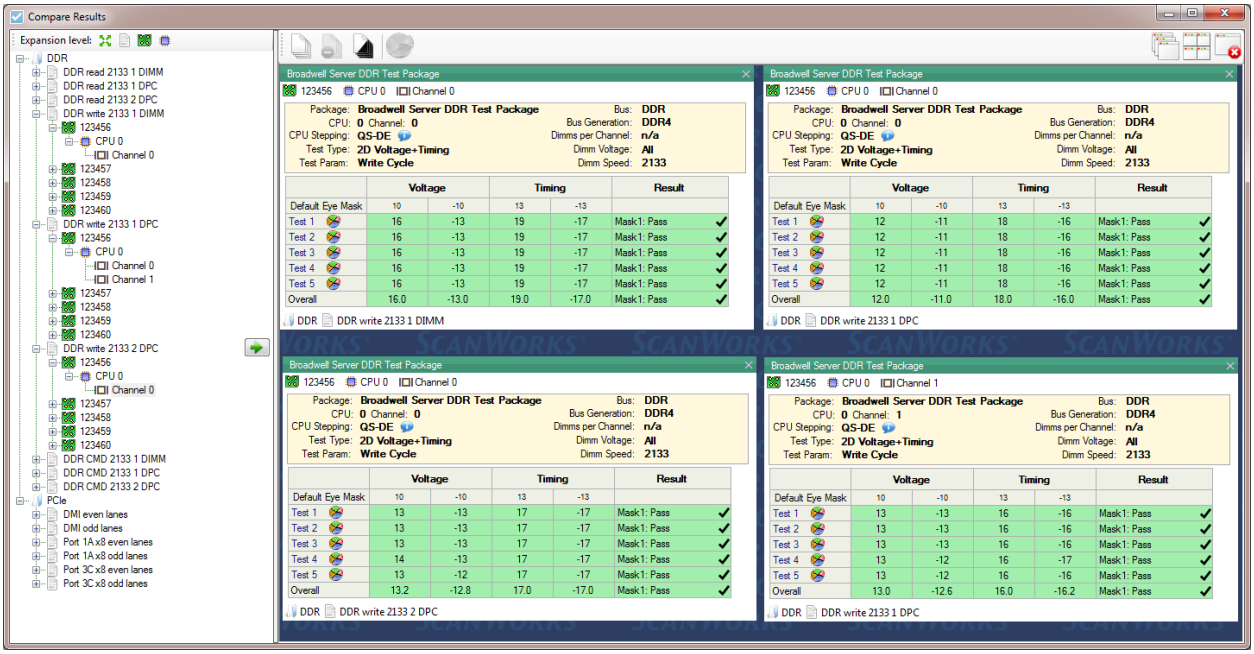


Figure 12: Platform 2 all test case results write direction

Data mining has shown that initially multiple test configurations are needed to investigate which configuration is the most stressful for each unique platform. By eliminating those configurations that do not need to be tested further because their margins are not worst case, significant time and money can be saved. This implies that the subsequent validation performed by I/O and memory validation engineers, as well as production engineers does not need to include configurations of the platform other than this configuration. This reduces total validation test time throughout the entire validation test cycle. Data mining allows validation engineers to make informed decisions that do not sacrifice product quality.

By eliminating those configurations that do not need to be tested further because their margins are not worst case, significant time and money can be saved.

Validation Must Move from Pass/Fail to Statistical Data Mining

By comparing the pass/fail and statistical data mining use cases summarized above, one can see how easy it is for engineers to miss critical aspects of HSIO performance when they rely on only

one pass/fail result. In contrast, the results of a validation methodology based on a statistical NxN approach will generate a rich database of information that, when mined with powerful analytic tools, can yield far reaching benefits to the system supplier.

Instead of thinking of validation as just pass/fail, validation should be thought of as a process. The process begins with establishing a baseline and documenting it so others can re-use or reproduce the process. Once this baseline has been established using the NxN method, any validation engineer can see not only whether buses passed or failed a test, but also the range of test results for buses, the amount of margin above the mask and the results on critical parameters. Once these data points are known, the engineer can establish a worst-case configuration based upon the platform. Other validation teams can now use this recipe for their validation testing. Since a baseline has been established, the variables for further testing have been minimized. The use cases described previously demonstrated that intermittent issues on a HSIO bus can only be identified through multiple tests. A single test has nothing to compare its results against. There is no baseline. While a board may pass a one-time eye mask test, running multiple tests on multiple systems could return a wide range of results which would show that other issues may be at work on the design, such as reflections, power fluctuations or routing problems.

A simple pass/fail method will not reveal the worst case margin parameters. And it won't produce a range of test results or determine the amount of operating margin on the bus. Only statistical validation using NxN testing will do this. Adopting an NxN methodology is critical in establishing a worst-case configuration for the platform. Some of the beneficial results will be less total time spent on validation testing while, at the same time, providing a more thorough and comprehensive validation process throughout the organization. Without identifying locations in the design with worst-case margins, engineers could be testing platform configurations with plenty of margin and concluding that further validation is not needed, when, in fact, the risk of failure or poor performance could be very great. Only by following the NxN testing methodology and statistically mining the resulting database will validation testing be focused where it should be.

One system configuration for validation testing is not necessarily effective for all platforms. Multiple configurations should be tested to identify which is the most stressful (i.e. least margin) in both the read and the write directions. All parameters should be examined to determine which

one causes the most stress. Ideally, one test configuration would be found for all validation testing because this would minimize the handling of the motherboard. Each parameter should be compared against the eye mask independently to determine the margins.

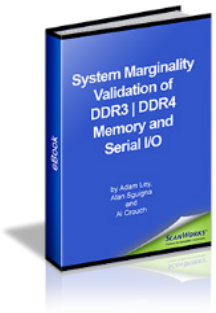
Conclusions

Most silicon providers of powerful and complex processors recommend an extensive suite of testing to statistically validate the HSIO buses on new system designs and motherboards. Unlike simple 1x1 pass/fail testing, statistically validating a design's buses with an NxN methodology is predictive insofar as it quantifies the risk of bus failures and performance degradations over time. The results from one pass/fail test only offers a snapshot of bus performance at the time the test was performed, typically in a research lab.

The repetitive testing inherent in an NxN statistical validation methodology generates a database of test data. Data mining this database with analytic tools can actually reduce the number of validation tests performed over the life cycle of the product, shorten the total time spent on validation and still deliver the benefits of statistical validation. With the right data mining analytic tools, signal integrity engineers, I/O and memory validation engineers, and production engineers are able to identify which high-speed buses are most vulnerable (least margin) to failures and/or performance degradations, and which particular validation tests place the most stress on the design. This can be found from a comparative analysis of test results to identify when the operating margins on a bus are not large enough to compensate for possible reductions in margins caused by variations in manufacturing processes, component performance, design flaws and other factors.

Learn More

For more validation eBooks, check these out: "System Marginality Validation on DDR Memory and Serial I/O" and "Signal Integrity Validation for Intel® Core™ Platforms"



Register Today!