Teradyne Users Group

Source-Synchronous Testing Using Paired Strobes

Keith Remlinger
Test Spectrum, Inc.
105 W. Riverside Dr, Suite 200, Austin TX 78704
keith.remlinger@testspectrum.com

Travis Volek
Freescale Semiconductor, Inc.
6501 William Cannon Drive West, Austin, TX 78735
travis.volek@freescale.com

Table of Contents

Abstract 1
Background 1
Source Synchronous 1
Jitter 1
Paired-Strobe Technique 3
Passing Criteria 4
Failing Criteria 5
Characterization 6
Status of Work 6
Tiger Implementation 6
Practical Application 9
Test Results 9
Significance and Highlights 9
Adaptable 9
Low Cost 9
Efficient 10
Supporting Data Results 10
Conclusions 13
Acknowledgements 13

Table of Figures

Figure 1: Hold Measurement with Synchronous Jitter 2
Figure 2: Edge Histogram for Hold Measurement 2
Figure 3: Setup Measurement, Example 1 3
Figure 4: Setup Measurement, Example 2 4
Figure 5: Failing Setup Measurement, Example 3 6
Figure 6: Tiger Paired-Strobe Process 8
Figure 7: DDR1 Setup at 333MHz Data Format Example 10
Figure 8: DDR1 Address Setup at 333MHz Verification 11
Figure 9: DDR1 Chip Select Setup at 333MHz Verification 12
Figure 10: DDR1 Data Setup at 333MHz Verification 12

Table of Tables

Table 1: Test and Reference Sweep Results, Example 1 4
Table 2: Test and Reference Sweep Results, Example 2 5
Table 3: Characterization with Paired-Strobes 6
Table 4: Specs and Pin-Pair Combinations for DDR1 9

Abstract –

Source Synchronous Testing Using Paired-Strobes provides an innovative approach for testing and characterizing tight AC design specifications using conventional tester hardware and an efficient post-processing algorithm. By using the combined results of the comparators of the source-synchronous clock and data pins, the effects of synchronous jitter can be removed. The technique has been implemented on the Tiger platform in Image and is currently used to characterize and test three different types of high-speed digital buses across five different product lines. Three of the products are using the technique for volume testing in final manufacturing.

Source-synchronous testing using paired-strobes provides an innovative approach for testing and characterizing tight AC design specifications using conventional tester hardware and an efficient post-processing algorithm. By using the combined results of the comparators of the source-synchronous clock and data pins, the effects of synchronous jitter can be removed. The technique has been implemented on the Tiger platform in IMAGE™ and is currently being used to characterize and test three different types of high-speed digital buses across five different product lines. Three of these products are using the technique for volume testing in final manufacturing.

Background

Source Synchronous

Source synchronous devices present a clock (strobe) to the connected application along with the data. The clock is used by the receiving application as a signal to strobe the corresponding data. In a typical system, the clock and data are routed together so that effects such as trace length and bus noise are common to both. This allows for a significant increase in data rates using standard IO device designs. Source synchronous designs present a challenge to testing because ATE functions primarily as the maintainer of synchronization. The ATE typically controls synchronization with the device. With the exception of specialized hardware, the ATE has no way of releasing this control and allowing the device to dictate when data should be compared.

Jitter

Synchronous jitter is defined as jitter that is present or common to both the clock and data. When measuring ac parameters such as setup or hold times using classical edge-find techniques, the effects of synchronous jitter may penalize the device and result in lower than expected measurements. With the edge-find technique, the comparators are swept through a range to locate the edges of interest for each pin *relative to the tester.* The difference between these locations is then calculated and expressed as setup or hold. For example, consider the waveform produced by a device in Figure 1. Synchronous jitter has caused both the clock and data to occur “earlier” in cycle 2 than in cycle 1. The actual hold measurement between the clock and data for both cycle 1 and 2 are the same (5ns).

To measure the worst-case hold time using an edge-find, one would search for the latest occurring clock edge and the earliest occurring data edge. In the Figure 1 example, the latest clock edge occurs in cycle 1 (4ns). The earliest data edge is found in cycle 2, at 6ns. Using this technique, the measured hold time is 2ns while in reality the actual hold time is 5ns.

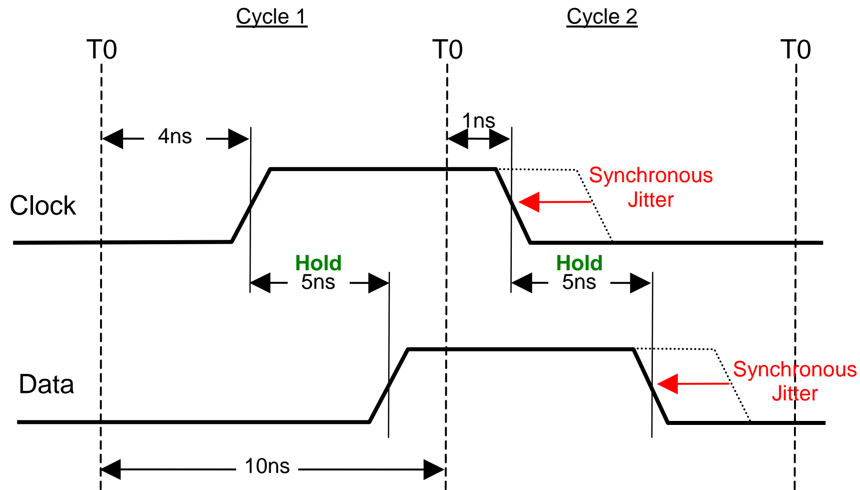


Figure 1: Hold Measurement with Synchronous Jitter

Figure 2 further illustrates this point. If all transitions of clock and data are viewed from the perspective of the tester cycle, then a histogram of the edge locations can be constructed relative to the tester cycle. Figure 2 shows the clock and data histogram. The worst, average, and best case hold measurements can be measured from the histogram. This worst-case analysis assumes the "latest" clock edge across all cycles occurred *in the same cycle* as the "earliest" data edge. Source synchronous test measurements require cycle-by-cycle analysis capabilities to remove the effects of synchronous jitter and effectively measure critical AC specifications.

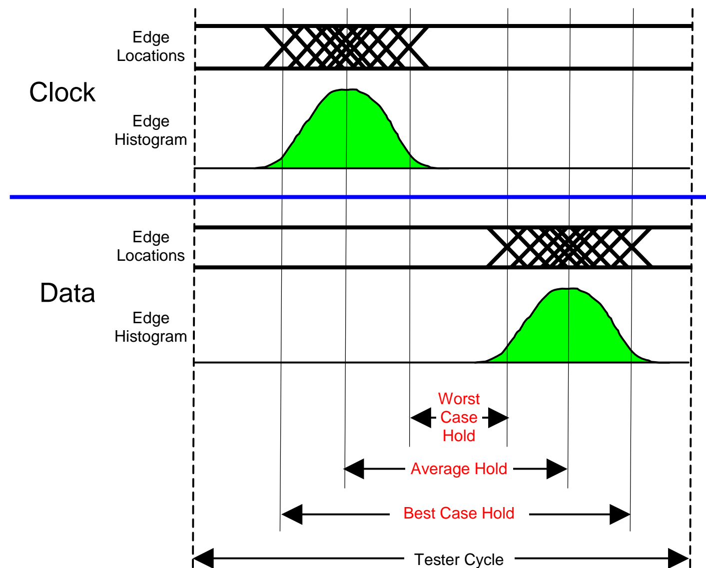


Figure 2: Edge Histogram for Hold Measurement

Paired-Strobe Technique

The paired-strobe technique allows cycle-by-cycle measurement of critical AC timing specifications. The technique uses the standard tester comparators to locate individual edges per cycle. This is achieved by pairing the clock (Reference) and data (Test) comparators together and fixing the distance between them by the desired measurement spec. Both comparators are then swept through the range of the jitter band of the clock and the results are analyzed. The combined states of both the Reference and Test pin are examined, cycle-by-cycle, to determine if the AC spec was met. Figure 3 illustrates this technique for measuring setup time. In this example, the Reference comparators start (position 1) in a passing region and the corresponding Test pins are fixed spec distance before the Reference. The patterns are executed and the results for cycles of interest are stored. Both the Reference and Test comparators are moved to the next sweep position (2) with the spec distance between them preserved. The process is repeated until all sweep iterations have been performed and all data has been collected. *If both the Reference and Test comparators pass in the same cycle, on the same sweep iteration, then "Spec" setup time must exist between them.*

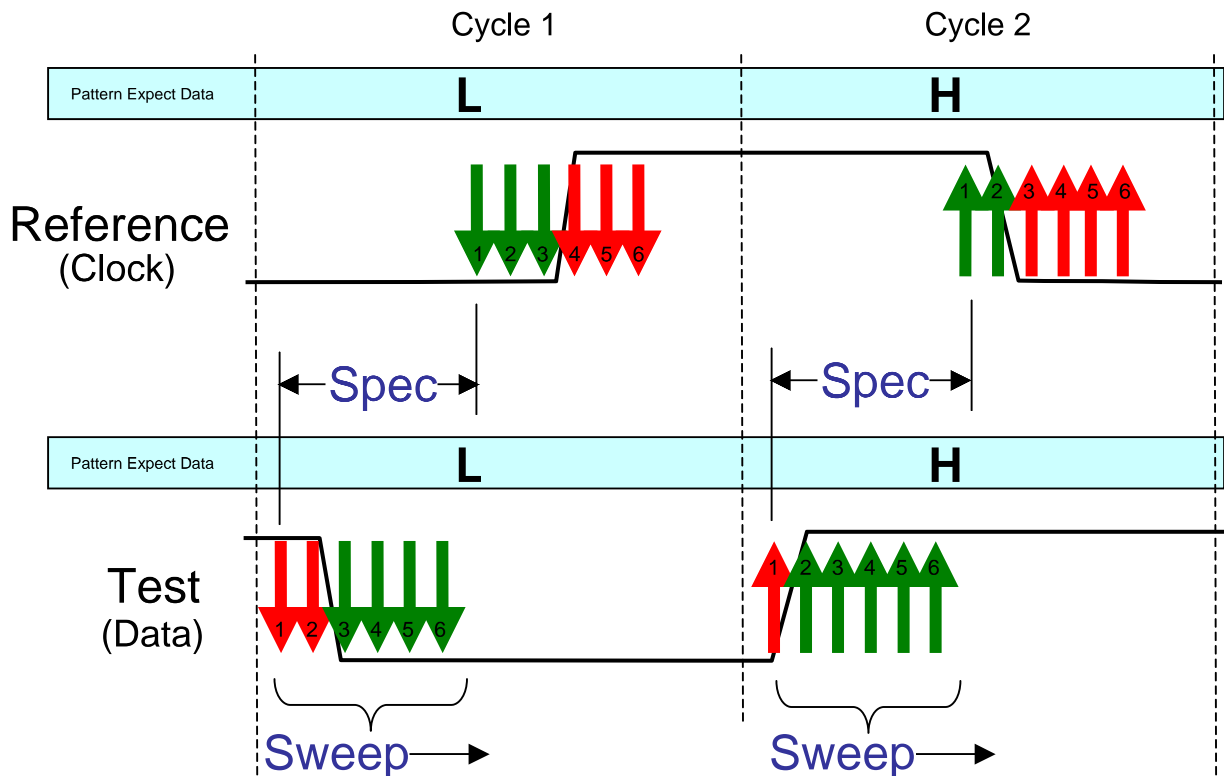


Figure 3: Setup Measurement, Example 1

Table 1 summarizes of the pass/fail status of each sweep location and cycle for the example in Figure 3. In Cycle 1, both the Reference and Test pin passed in Sweep 3. For Cycle 2, both passed in Sweep 2. Both cycles met the "Spec" setup requirement – they just happen to pass in different sweep iterations. Since the relative time between the Test and Reference pins are all that matter for the AC setup measurement, the example would be considered passing.

Sweep	Cycle 1 Test	Cycle 1 Reference	Cycle 2 Test	Cycle 2 Reference
1	Fail	Pass	Fail	Pass
2	Fail	Pass	Pass	Pass
3	Pass	Pass	Pass	Fail
4	Pass	Fail	Pass	Fail
5	Pass	Fail	Pass	Fail
6	Pass	Fail	Pass	Fail

Table 1: Test and Reference Sweep Results, Example 1

A traditional edge-find measurement uses the composite results of all cycles and would fail the example presented in Figure 3. The Reference pin stops passing in sweep position 2 (cycle 2) and the Test pin begins passing in sweep position 3 (cycle 1). *No single timing location (sweep position) can be found where all cycles pass both the Test and Reference pins*.

Passing Criteria

The example used in Figure 3 illustrates the PASS-PASS passing criterion. Both the Test and Reference pins are expected to pass in order for the spec to be considered passing. This criteria is dependent on the arrangement of the expect (compare) data found in the patterns being used to test the spec. Figure 4 demonstrates a similar situation where the waveforms from the device match those of Figure 3. The difference between these examples lies with the way the expect data is arranged in the pattern.

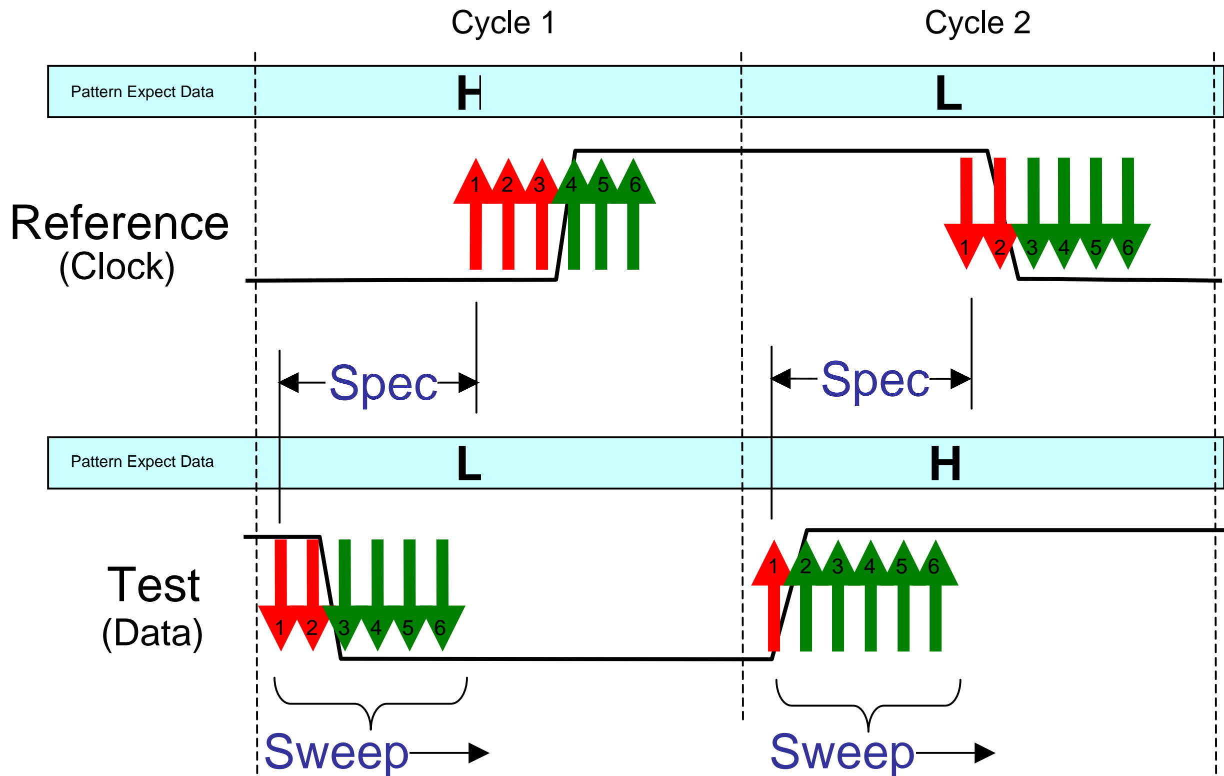


Figure 4: Setup Measurement, Example 2

The Reference pattern's expect data in Figure 4 has changed, resulting in a change to how the AC spec can be measured. In this example, the AC setup spec is present when the Test pin comparator passes and the Reference comparator is still failing. This represents a PASS-FAIL passing criterion. The passing sweep locations are the same in this example; the only difference is the arrangement of the pattern expect data and the resulting passing criteria. Table 2 contains the pass/fail status for each sweep and cycle.

	Cycle 1		Cycle 2	
Sweep	**Test**	**Reference**	**Test**	**Reference**
1	Fail	Fail	Fail	Fail
2	Fail	Fail	Pass	Fail
3	Pass	Fail	Pass	Pass
4	Pass	Pass	Pass	Pass
5	Pass	Pass	Pass	Pass
6	Pass	Pass	Pass	Pass

Table 2: Test and Reference Sweep Results, Example 2

Similar to the effects of the arrangement of pattern expect data, the type of measurement may require a different passing criterion. For example, measuring hold time for Example 1 would require a PASS-FAIL passing criterion; hold for Example 2 would require PASS-PASS.

Failing Criteria

Depending on the test requirements, it may be necessary or convenient to define a set of failing criteria. Using Example 1 again, imagine a waveform that does not meet the "Spec" AC setup requirement.

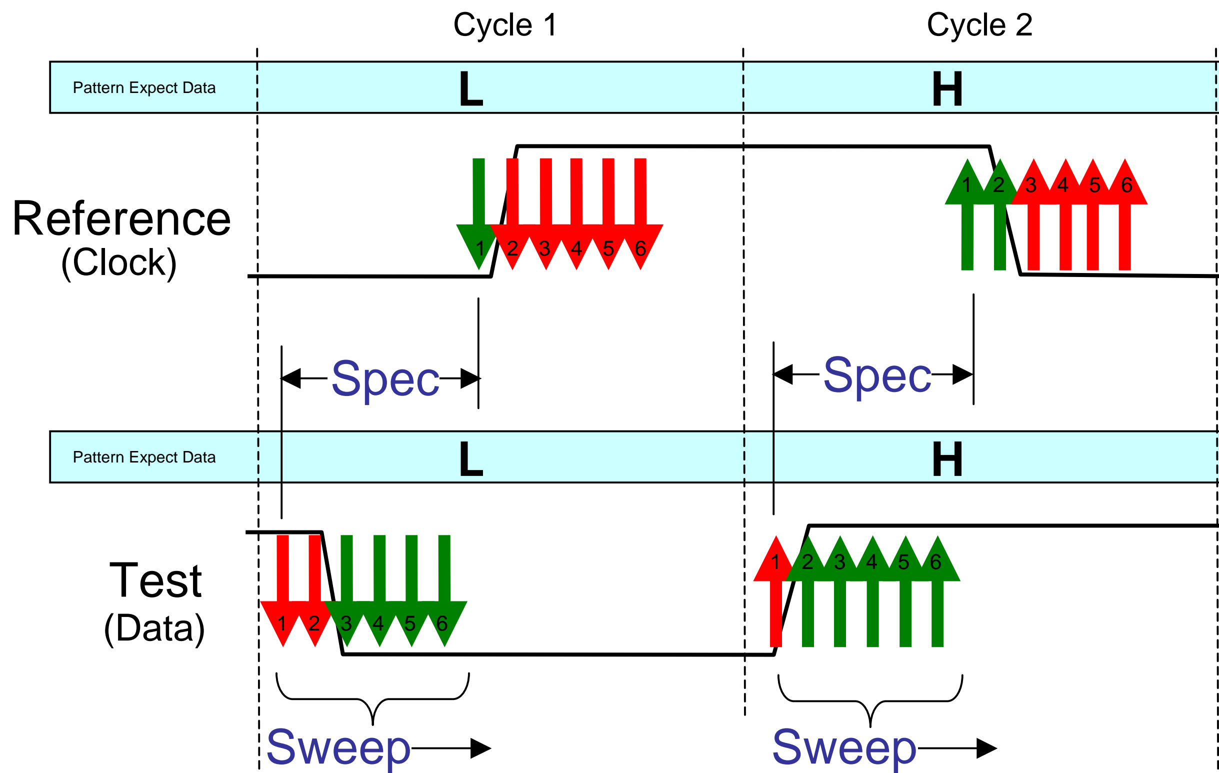


Figure 5: Failing Setup Measurement, Example 3

In Sweep iteration 2, both the Test and Reference pins fail in Cycle 1. If the failing criterion FAIL-FAIL is selected, this sweep and cycle would cause the setup spec measurement to fail. The selection of a failing criterion is not necessary but may provide more coverage or a quicker means to determine a failure. Example 3 would have also failed because the passing criteria PASS-PASS was not met.

Either a passing or failing criterion must be specified. *A failure is determined by the lack of the passing criterion or the presence of the failing criterion on any cycle.* Both criteria are optional but one is necessary for a measurement to be made. For flexibility with regards to measurement type and pattern expect data, all binary combinations of the passing and failing criteria are available for verifying measurements.

Characterization

In addition to verifying if an AC spec is met, the paired-strobe technique can be extended to make actual spec measurements or AC characterization. The process explained for the above examples can be repeated over and over, varying the actual spec value with each iteration. Table 3 demonstrates the characterization of setup time with the passing criterion of PASS-FAIL and the failing criterion of FAIL-PASS. For Spec1, the passing criterion was met on sweep 1 and 2 (indicated in light-blue). Spec2 also had one instance of the passing criterion on sweep 2. Spec3 however met neither the passing criterion nor the failing criterion. Spec4 and Spec5 both contained failing criterion measurements (indicated in yellow). The actual measurement reported using this approach would be Spec2. This iteration is the last instance where the passing criterion was met.

	Spec1		Spec2		Spec3		Spec4		Spec5	
Sweep	Test	Reference	Test	Reference	Test	Reference	Test	Reference	Test	Reference
1	Pass	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail	Fail
2	Pass	Fail	Pass	Fail	Fail	Fail	Fail	Fail	Fail	Fail
3	Pass	Pass	Pass	Pass	Pass	Pass	Fail	Pass	Fail	Pass
4	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Pass	Fail	Pass

Table 3: Characterization with Paired-Strobes

It is important to note that Table 3 illustrates only a single cycle for simplicity of explanation; numerous cycles can be characterized and measured. The same analysis would be performed on each cycle of interest.

Status of Work

The Paired-strobe measurement technique was initially developed and debugged for the Teradyne Tiger test platform in Image. A standard code library was developed that allows testing and characterizing multiple sets of Test and Reference pins across many cycles and patterns.

Tiger Implementation

The Tiger Implementation uses digital capture (dig_cap) to store the passing/failing status for each cycle of interest. Scripts are used for parsing potential patterns and auto-inserting the necessary dig_cap microcode. The patterns are then chained together to form large groups (bursts). Each iteration of the paired-strobe technique bursts the single large pattern group.

In the test program code, the Test and Reference pins are registered for each type of measurement to be made. The registration process gathers information about each pin-pair, such as the passing

and failing criteria. The expected passing and failing sequence for the Reference pin during a sweep is also indicated during registration. This information is needed to determine that cycles are behaving as expected.

With registration complete, training can begin. Since multiple specs may be measured on the same vector, determining which specs belong with which vectors is difficult. To alleviate this issue, a set of training timing must be created. The training timing forces the specs that are to be measured to fail. These failing vectors are identified and associated with the spec under test. The locations for each test and pin-pair are stored in a file. The creation of this file is done in an engineering (debug) mode and is only done once per spec and pattern. Once created, the test program reads the information from the file and configures each spec and registered pair with the vector information. See the "Training Routine" block in Figure 6 for more information.

Optimization of the spec test is achieved by performing a Reference search. The Reference search locates the jitter band of the Reference pins and updates the sweep start and stop ranges to span this band. The Reference search is done using the digital comparators and HRAM. For each iteration of the search, the number of failures is retrieved from HRAM. In the case of the Reference transitioning from pass to fail, the HRAM data is used to locate the last sweep iteration where the Reference passed and the first iteration where the number of failures stabilized. See the "Reference Search" block in Figure 6 details on how a search is performed.

Spec testing is performed by sweeping through the Reference jitter band. In each iteration, the data is captured and stored for processing. When the sweep range is complete, a separate analysis thread is started and capture begins on the next spec set. Each pin-pair and its associated criterion are verified while the next test's data is being captured. When all specs have been tested and all analysis threads have stopped, the results are used to verify whether or not a spec passes or fails. The steps involved in Spec testing are listed in the "Test Flow" block of Figure 6.

Spec characterization is performed by repeating the test process while varying the Spec value through a range. Typically, the initial Spec value would be set so that the test passes and then gradually increase the Spec until it no longer passes. The characterization result is the last Spec value that passed.

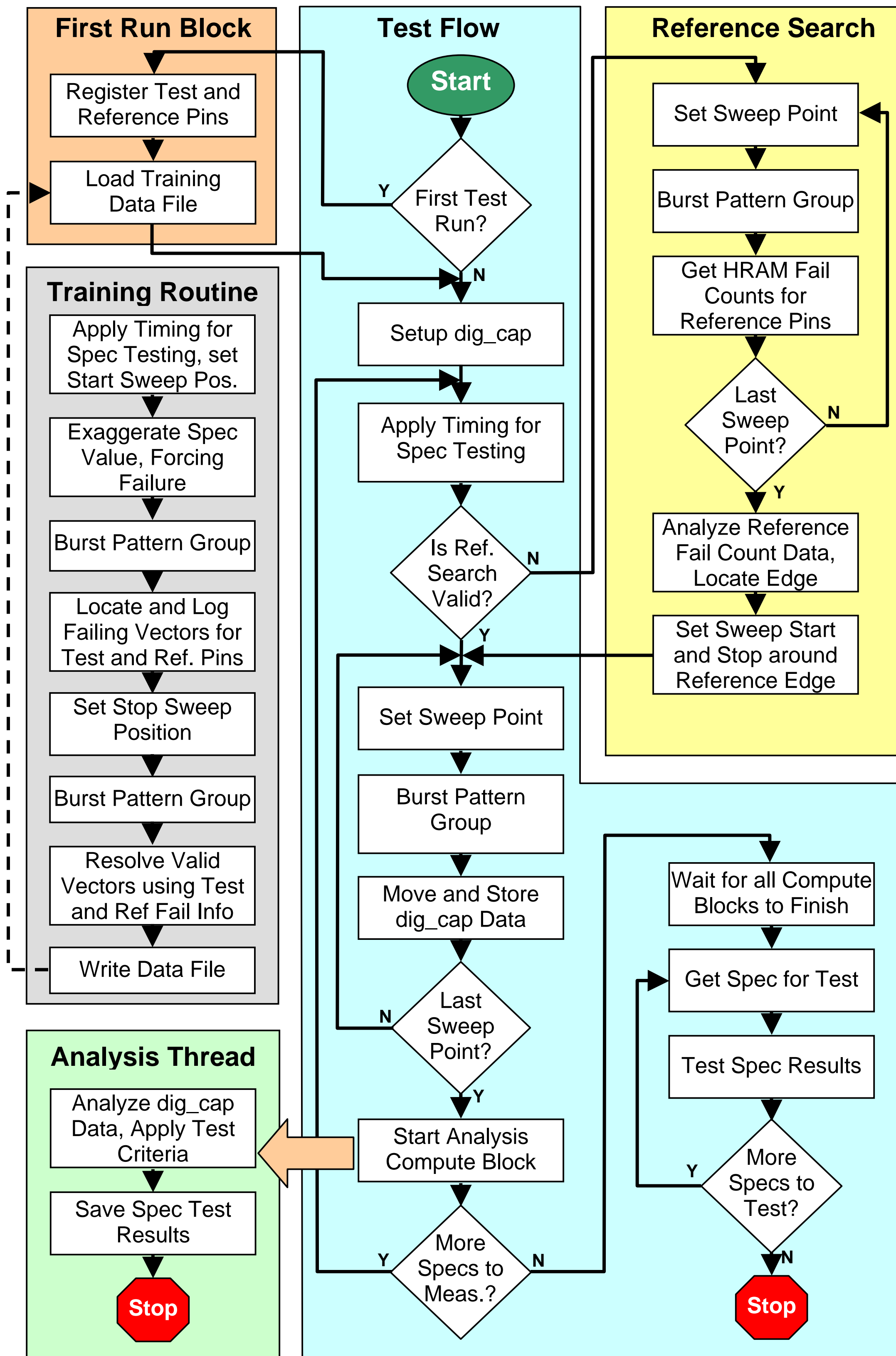


Figure 6: Tiger Paired-Strobe Process

Practical Application

Four source-synchronous bus types have been tested and characterized using the paired-strobe technique: DDR1, DDR2, Local Bus, and three-speed Ethernet. In each case, multiple Test and Reference pairs were analyzed in order to measure numerous AC specs across thousands of cycles. As an example, the DDR1 bus used for initial development contains 12 clocks, 9 data strobes, 15 address bits, 12 control bits, 64 data bits, 8 error bits, and 8 mask bits. Approximately 250,000 test vectors were used to measure the functional and AC performance of this design block. Out of the 250,000 vectors, 3,800 were selected and marked as candidates for AC spec measurement by identifying data and clock activity. The number and type of specs tested on this bus are listed in Table 4. Each spec listed in Table 4 was tested at three speeds (200MHz, 266MHz, and 333MHz) and two voltages (minimum and maximum).

Spec Name	Test Pins	Reference Pins	No. of Pairs	No. of Vectors per Spec	No. of Specs Tested across speed/voltage
Data Setup	64 data, 8 errors, 8 masks	9 strobes	720	3,300	19,800
Data Hold	64 data, 8 errors, 8 masks	9 strobes	720	3,300	19,800
Address/Control Setup	15 addresses, 12 control	12 clocks	324	2,800	16,800
Address/Control Hold	15 addresses, 12 control	12 clocks	324	2,800	16,800
Strobe Minimum Skew	9 strobes	12 clocks	108	800	2,400
Strobe Maximum Skew	9 strobes	12 clocks	108	800	2,400
		Totals	**2,304**	**13,800**	**78,000**

Table 4: Specs and Pin-Pair Combinations for DDR1

Test Results

In the case of the AC spec testing, the result is a pass/fail value indicating the configured specs were met or not. The main design constraint for production AC spec testing is efficiency of time. Where possible, AC specs are combined to minimize test time. Combining specs results in a loss of visibility as to which spec is causing the failure but test coverage and efficiency are achieved. For characterization, the results can be customized. For quick characterization, the results can be set to measure the worst value for each spec along with the pin-pair and cycle. For more information, the characterization routine can be set to output the worst value for every pin-pair along with the failing cycle. This level of information can be used to evaluate substrate and application layout designs. Likewise the characterization can be setup to report results for every pin-pair and cycle.

Significance and Highlights

Adaptable

The use of verification criteria allows different types of spec testing, regardless of pattern and timing format constraints. The technique can be used to measure many types of AC specs on many types of buses.

The technique can be configured to achieve fine test resolution during characterization or coarser resolution to conserve time during production testing.

Use of standard test hardware and modifiable code library provide flexibility for future bus architectures and new tester platforms. The implementation used on the Tiger can easily be transferred to other platforms.

Low Cost

Tight AC specs can be tested without the use of specialize hardware.

Efficient

The DDR1 example shows the level of complexity associated with measuring ac specs across all pins, vectors, and test conditions. The pin-pair technique allows for testing and characterizing many (in the case of the DDR1 prototype, 78,000) AC specs in a reasonable amount of test time. It is also useful for quickly identifying device issues and deterministic jitter (data dependencies) across numerous combinations of pins and cycles. Identification of these issues using an oscilloscope alone would be cost prohibitive given the number of possibilities. The identified pins/cycles of interest can then be examined further using an oscilloscope or in an application environment.

Supporting Data Results

Figure 7 shows a portion of a representative datalog output from the DDR1 application used in the previous examples. The spec tested in this datalog is data setup at 333MHz. The listing indicates the worst-case value measured across multiple patterns for each pin pair along with the pattern and vector that resulted in this measurement. The output is listed from worst performing pair to best.

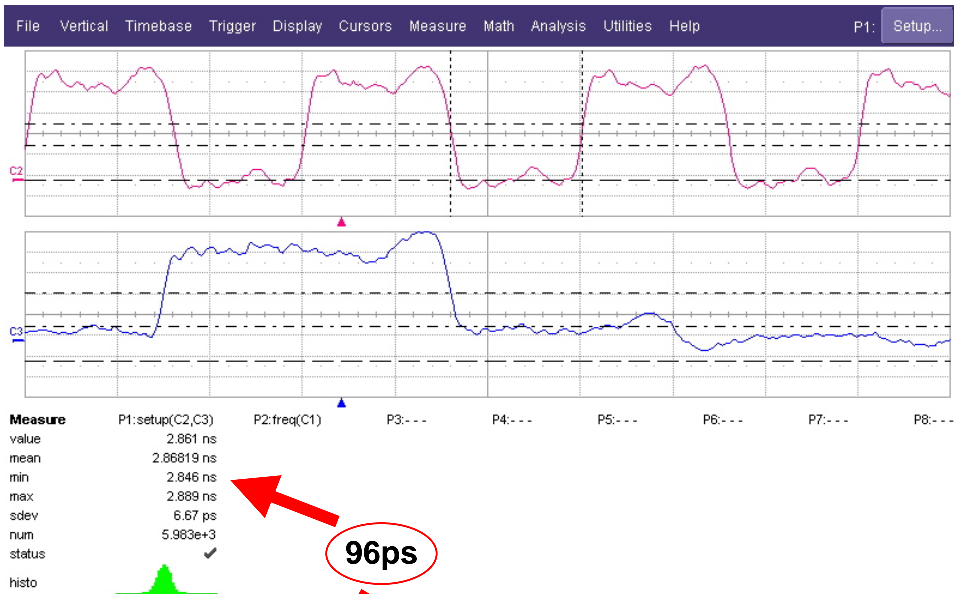
```
**** Spec: tddkhds, Res: 3e-11, Levels: min, Speed: 333mhz, Temp: 25

Pair: mdq1/mdqs0, Spec: 9.6e-10, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq2/mdqs0, Spec: 9.6e-10, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq6/mdqs0, Spec: 9.6e-10, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq8/mdqs1, Spec: 9.6e-10, Store: 12035, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14219
Pair: mdq10/mdqs1, Spec: 9.6e-10, Store: 12035, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14219
Pair: mdq17/mdqs2, Spec: 9.9e-10, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq30/mdqs3, Spec: 9.9e-10, Store: 12035, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14219
Pair: mdq31/mdqs3, Spec: 9.9e-10, Store: 12391, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 15091
Pair: mdq50/mdqs6, Spec: 9.9e-10, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq11/mdqs1, Spec: 1.02e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq14/mdqs1, Spec: 1.02e-09, Store: 12487, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 15331
Pair: mdm1/mdqs1, Spec: 1.02e-09, Store: 12009, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14161
Pair: mdq20/mdqs2, Spec: 1.02e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq22/mdqs2, Spec: 1.02e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq24/mdqs3, Spec: 1.02e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq25/mdqs3, Spec: 1.02e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq26/mdqs3, Spec: 1.02e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq43/mdqs5, Spec: 1.02e-09, Store: 718, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 15778
Pair: mdq47/mdqs5, Spec: 1.02e-09, Store: 9058, Pattern: ddr_prod_test1_cs0_11a_m4_dsio, Vector: 17242
Pair: mdq54/mdqs6, Spec: 1.02e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq55/mdqs6, Spec: 1.02e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq58/mdqs7, Spec: 1.02e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq62/mdqs7, Spec: 1.02e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq18/mdqs2, Spec: 1.05e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq23/mdqs2, Spec: 1.05e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq28/mdqs3, Spec: 1.05e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq35/mdqs4, Spec: 1.05e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq51/mdqs6, Spec: 1.05e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq63/mdqs7, Spec: 1.05e-09, Store: 718, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 15778
Pair: mdm7/mdqs7, Spec: 1.05e-09, Store: 777, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 15945
Pair: mdq21/mdqs2, Spec: 1.08e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq27/mdqs3, Spec: 1.08e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdq29/mdqs3, Spec: 1.08e-09, Store: 12035, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14219
Pair: mdq39/mdqs4, Spec: 1.08e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mdq42/mdqs5, Spec: 1.08e-09, Store: 3933, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32137
Pair: mecc4/mdqs8, Spec: 1.08e-09, Store: 3949, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 32169
Pair: mdm0/mdqs0, Spec: 1.11e-09, Store: 4177, Pattern: ddr_prod_msrcid_mdval_11a_m4_dsio, Vector: 10461
Pair: mdq19/mdqs2, Spec: 1.11e-09, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
Pair: mdm2/mdqs2, Spec: 1.11e-09, Store: 12009, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14161
Pair: mdq46/mdqs5, Spec: 1.11e-09, Store: 3750, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 31766
Pair: mdm5/mdqs5, Spec: 1.11e-09, Store: 737, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 15833
Pair: mdq53/mdqs6, Spec: 1.11e-09, Store: 758, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 15890
Pair: mdm6/mdqs6, Spec: 1.11e-09, Store: 737, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 15833
Pair: mdm4/mdqs4, Spec: 1.14e-09, Store: 777, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 15945
…
```

Figure 7: DDR1 Setup at 333MHz Data Format Example

Verification of the measurement capability was done using specially designed upside-down load boards and a Lecroy Wavepro SDA6000 (20Gs/s, 6Ghz) Oscilloscope. Characterization measurements on the Tiger using the upside-down boards were done for each AC spec of interest.

The selection of the pin-pairs, patterns, and vectors identified by the Tiger were then measured using the scope. Triggers were put in the pattern so that the vector of interest could be identified and measured. Figures 8, 9 and 10 are scope captures of different specs measured using this technique. In Figure 8, the AC spec address setup to clock was measured using the scope. The minimum reading out of nearly 6000 measurement is 2.864ns. The Tiger measurement made using the pin-pair strobe technique measured 2.750ns. The difference between the two instruments is 96ps. Similarly, Figure 9 shows the results of Chip Select setup to clock. The Tiger measurement was 2.750ns and the scope measured 2.794ns. Figure 10 contains data setup to strobe measurements at 333MHz. The Tiger measurement was 885ps; the scope measured 828ps.

Scope Measurement:

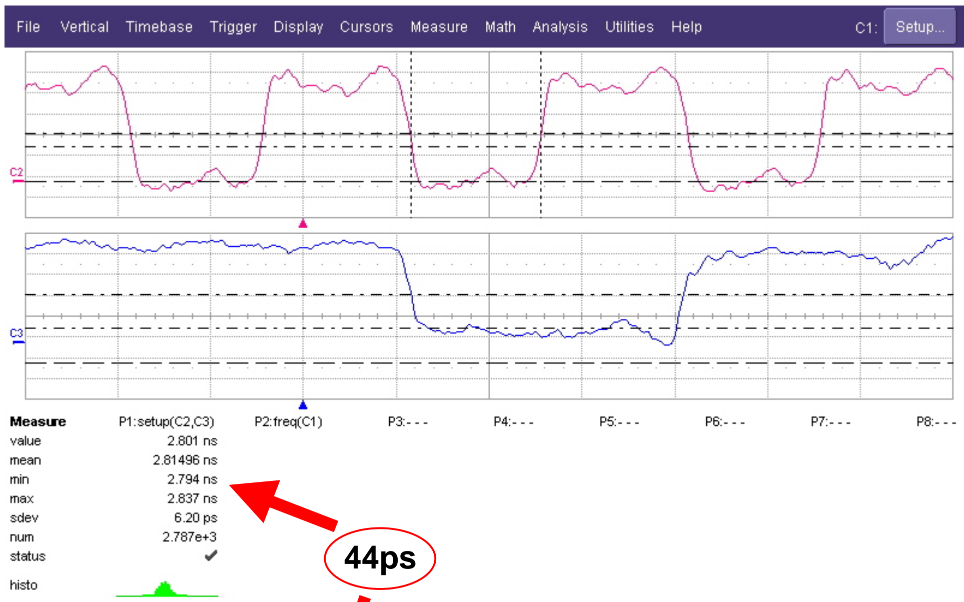


Tiger Measurement:

```
Pair: ma5/mck5, Spec: 2.75e-09, Store: 1936, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 19448
```

Figure 8: DDR1 Address Setup at 333MHz Verification

Scope Measurement:

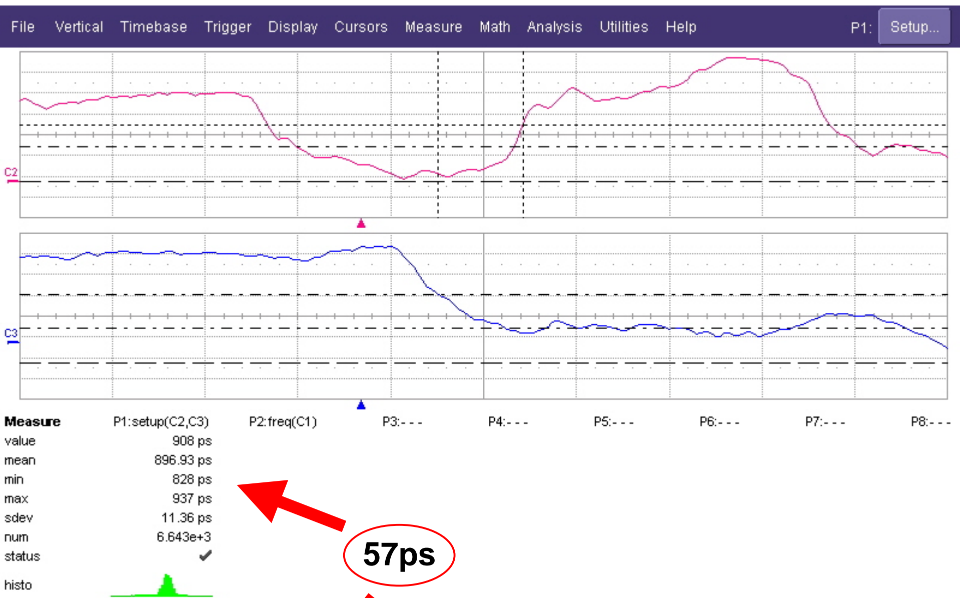


Tiger Measurement:

```
Pair: mcs_b1/mck5, Spec: 2.75e-09, Store: 444, Pattern: ddr_prod_ecc_cs1_11a_m4_dsio, Vector: 7296
```

Figure 9: DDR1 Chip Select Setup at 333MHz Verification

Scope Measurement:



Tiger Measurement:

```
Pair: mdq0/mdqs0, Spec: 8.85e-10, Store: 11987, Pattern: ddr_prod_x32_cs2_11a_m4_dsio, Vector: 14107
```

Figure 10: DDR1 Data Setup at 333MHz Verification

Conclusions

The paired-strobe technique provides an efficient test technique for measuring and testing AC specs across large numbers of cycles and pins. The resulting data is invaluable for quickly identifying design issues and data dependency problems. The use of passing and failing criteria provide flexibility for testing most AC specs regardless of the comparator arrangement in the patterns. Since standard test hardware is used, the implementation is low-cost and can be used to test very large (wide) buses. The use of standard test hardware makes the technique adaptable to other test platforms as well as other types of devices.

Acknowledgements

Ken Mockler, Freescale – Ken conceived the paired-strobe test process and was invaluable in allocating resources and time for the development and implementation. Many thanks for all his support and time.

Raoul Belleau, Teradyne – Raoul created the basic architecture used for the Tiger implementation and assisted throughout the project with ideas, improvements, and testing. Raoul has also developed training materials and curriculum for the Tiger implementation and is maintaining the code library and associated tools. The project would not have happened without his experience and assistance.