Key Impediments to DFT-Focused Test and How to Overcome Them

Ken Posse, Geir Eide
Teseda, Corporation
812 S.W. Washington
Portland, Oregon 97205

Abstract

In a carefully structured study spanning several months, the authors visited numerous companies focused on Design For Test methodologies in SoC Test, Characterization, and Failure Analysis. In interviews with the leading engineers in these projects, the various DFT structures and test processes used were studied. The results of the study revealed a number of impediments to the adoption of these processes on low-cost, DFT-focused testers. This paper presents some of the more glaring difficulties together with suggestions as to how they might be overcome.

Introduction

Existing functional testers are expensive, and therefore, time on these machines is also expensive. The cost to test a part is the same, whether or not the cost-sensitive resources are used. For this reason, many companies have expressed an interest in using testers designed to focus on DFT test methods such as DC and AC Scan, Iddq, and BIST. Since these machines can cost as much as an order of magnitude (or more) less than a traditional tester, moving tests from the more expensive environment to the less expensive environment can save significant amounts of money.

This all makes a lot of sense, of course; but is it possible to move tests from the more expensive machines to the less expensive systems? The answer lies in the manner in which the DFT structures were implemented in the devices to be tested. In this document, the authors will look at Scan and BIST techniques and point out some traps that are easy to fall into.

What Costs So Much

Before going into the details of the designs, it is important to understand what drives up the cost of test on traditional test equipment. Basically there are three, more or less interrelated characteristics that represent the difference between expensive and low-cost test equipment. These characteristics are:

1. Speed – the rate at which vectors can be applied. In general, the cost of increasing the vector application rate goes up exponentially with the frequency.
2. Precision – both voltage and time. As with vector rates, an increase in the precision in both time and voltage typically will cause an exponential increase in the cost of the equipment.
3. Flexibility – this is a more abstract concept, but in this context it is meant to represent such things as the number of independent complex waveform generators available, the number of tester pins available, etc. While it is more difficult to define a "flexibility variable" that relates cost to system features, it is clear that the cost of test equipment increases significantly with flexibility.

So, when facing the need to reduce the cost of test equipment, DFT-focused testers are usually designed to be slower, less precise, and less flexible than their traditional cousins. This does not mean that they are less capable, however. Because DFT chip architectures focus on low-cost test techniques, the requirements for speed, precision, and flexibility have been largely removed. As a result, high-quality can be achieved without the need for expensive test equipment.

The Study

Over the last few months, the authors have studied Design For Test practices at a number of well-known companies[5][6]. This study revealed that many of today's DFT architectures are designed specifically for execution on traditional testers and as a result have grown dependent on some of the

0-7803-8106-8/03 $17.00 Copyright 2003 IEEE

high-cost resources that DFT-Focused machines seek to eliminate. While this may be somewhat expected (after all, DFT_Focused Test is fairly new), it is still instructive to examine some of these chip architectures, to understand the reasons for their dependence on high-cost equipment, and see what can be done to eliminate these dependencies.

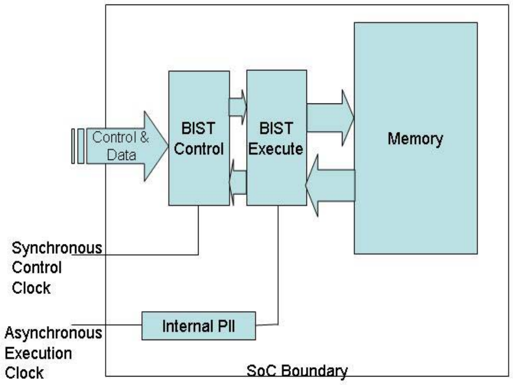
Memory BIST

The study revealed that memory BIST is pervasive. The number and size of memories in SoCs is increasing, and MBIST is used to test virtually all except the very small. The study also revealed that most MBIST is home-grown, though there is an increasing propensity toward purchasing the technology from third parties. Homegrown solutions were preferred, in most cases where flexibility with respect to choice of algorithms was desired, and also because MBIST controllers were considered simple to design. Third party solutions were usually preferred for more sophisticated debugging mechanisms.

About half of the MBIST systems that we looked at were controlled by the IEEE 1149.1 Test Access Port (TAP); the rest were controlled by a variety of mechanisms including multiplexed I/O and dedicated Internal Scan chains.[5,6,9,10] In all cases, the memories were tested at-speed; most through the use of the system clock, though a few were tested by a direct clock from the tester.

There are two purposes of MBIST architectures; the first is to determine whether or not a given memory is good or bad (pass/fail) and the second is to provide information as to the exact location of the failure and the test that was running when the failure occurred (i.e. a memory map). This data can be used for two purposes; repair and process control.

From this it can be seen that it is important to be able to both send data to the MBIST controller (for such things as selecting the test routine, initializing certain masks, etc.) and to receive data from the MBIST controller (for such things as pass/fail, the memory locations of failures, etc.)[9]. Many memory test engines are designed to synchronize to a single clock, and therefore they take data in and send data out, and run the test all at the same clock rate. Unfortunately, DFT-focused testers typically cannot send and receive data nearly as fast as the required clock rates of an MBIST engine (typically 133 Mhz or higher).



A two-clock MBIST Architecture. Communications to and from The tester are via a slow (<50Mhz) data path. Execution of the BIST, however, is at the rated speed of the memory.

Figure 1: Load Slow, Run Fast MBIST Architecture

Therefore, for MBIST designs, the authors recommend use of the IEEE 1149.1 interface as both a setup and data-extraction mechanism for MBIST. In this way, the test engine can execute synchronous with the system clock, and the tester can send and receive data at whatever clock rate is acceptable to the equipment. This technique is termed “Load Slow/Run Fast” (Figure 1). ”Load Slow”, for the fact that the MBIST control structures can be loaded with commands and data through TDI at the tester-supplied rate of TCK, and “Run Fast”, for the fact that once loaded, the BIST engine can execute at system clock rates supplied by the internal PLL.[1]

A second difficulty often encountered in MBIST engines is the termination condition. In Pass/Fail mode, most MBISTs run to completion, which is a fixed number of clock cycles. In this way, a tester need only set up the execution of a test and then count clock cycles to determine when the test has completed. Pass/Fail can then be read from a status register.

However, in diagnostic mode, some MBIST engines stop on the first failure and raise a flag, expecting to be queried within a certain number of cycles of the

---

[1] IEEE 1149.1 is recommended for this functionality because it is a common standard present in many SoC’s already and because it has proven to be an effective solution to this problem.

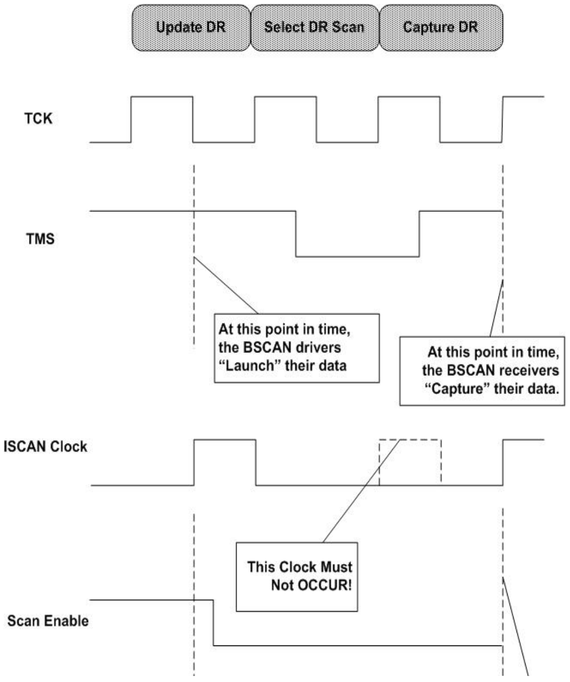


Figure 2: Synchronizing BSCAN to ISCAN

detection of the failure. Watching this flag requires what is known as "Matchloop Capability" on the tester, and the matchloop latency must be less than the number of cycles expected by the BIST controller. If these conditions are not met, data can be lost.

Because many DFT-Focused testers do not have matchloop capability, and those that do may not be able to meet the latency[2] specification, the authors recommend (once again) the use of the IEEE 1149.1 TAP. The memory test engine should halt on first fail, and set a failure bit in the status register; while the tester continues to count clock cycles to the end of the test and then queries for pass/fail. If a failure is detected, the test system can then set a bypass count in the test engine and restart the test. This will cause the test to execute, ignoring the first "N" failures. An alternative for some BIST engines is the recording of multiple failures that can then be read out without matchloop capability.

Many DFT-Focused testers provide a clock generator that is capable of providing clock rates up to several hundred megahertz. In many cases, however, these clocks are provided by a separate clock generator that is not synchronized to the vector rate of the tester. For this reason, the authors recommend that the MBIST engine be designed so that an asynchronous system clock may be used.

DC Scan (Stuck-At Scan)

To be truly effective, the scan chains must be architected to test 100% of the logic of the SoC. On a traditional (expensive) tester, with a high pin-count, this is not a problem. Traditional testers, with their plethora of test pins, can contact the full I/O of a SoC, and thus directly test the I/O shadow logic[3]. It must be kept in mind, however, that reducing the pin count is paramount in reducing the cost of test. Therefore, DFT designs must be capable of testing the I/O shadow logic in order to achieve the high defect coverage necessary to guarantee chip quality.

To account for this, the authors recommend the use of either the IEEE 1149.1 Boundary-Scan architecture or an I/O scan ring to provide coverage for the I/O shadow logic. If this is done, only the scan and other test control pins need be contacted during the test process (this is often referred to as "Reduced Pin Count Testing" or RPCT). For the purposes of DC Scan, it is possible to synchronize the Boundary-Scan boundary register "Update" and "Capture" cycles with the internal scan (ISCAN) Launch and Capture cycles (see Figure 2). Therefore, test data may be exchanged between the two scan architectures (BSCAN and ISCAN) resulting in a higher quality test for the device. While this is possible, it often not a trivial task to properly exercise the 1149.1 circuitry in an ATPG tool.

A second potential problem with the scan is the scan chain routing. Scan chains frequently are routed through multiple clock domains (in some cases, more than twenty have been encountered). Because of variations in insertion delays, and other things that affect clock skew, these designs often require an independently programmable shift clock for each of the domains.

Unfortunately, DFT-Focused test equipment often has a limited number of independently programmable clock generators that are synchronous with the scan cycle. This may make it impossible to

---

[2] "Latency" is defined as the number of vector cycles required to recognize the specified matchloop condition and act on it.

[3] "Shadow Logic" refers to regions of logic that cannot be tested by scan chains alone. Thus, logic that lies between the I/O pads and the first (last) group of scan cells is the "I/O Shadow Logic".

test SoCs that require a large number of simultaneous, but independent, clock domains without using expensive and cumbersome circuitry on the load board.

There are two potential solutions to this problem:

1. Do not route scan chains across more than one or two clock domains. This may increase the number of scan chains on the DUT, but this type of routing will allow multiple domains to share a single clock. If the number becomes too large, chain multiplexing technology can be employed (i.e. see the IEEE P1500 standard as a potential solution).
2. Use an LSSD-style scan flip-flop. There are designs of this type of flop that function as a two-clock LSSD type device during scan operations, but resort to a D-type edge-triggered flip-flop during normal functional operation. Since the two, non-overlapping clock model of the LSSD shift is "safe" (i.e. protected from the problems such as shoot-through, and other hold violations that plague the Mux-D type operation), it can be shown that a maximum of three independent clock generators can be made to service any number of clock domains.

AC Scan (Delay Fault Testing)

Perhaps because this is an area of DFT that has yet to reach maturity, it tends to have the largest number of potential problems as far as DFT-Focused Test is concerned. The key characteristics of concern are listed below, and will be discussed individually:

1. Precision requirements for Scan Enable when used with Last Shift Justification
2. Synchronization of the Primary Inputs (PIs) with the Launch signal.
3. Provisions for applying ISCAN synchronized data from/to the PIs and POs.
4. Clock Pulse requirements including the clock formats Late-Late-Early (LLE) and Early-Late-Early (ELE).
5. Synchronizing to a Phase Locked Loop (PLL) generated Launch/Capture.

Before going further with this discussion, the authors wish it known that they have no intention to enter into a "religious" argument about the relative value of Launch Off Last Shift (LOLS) versus Launch Off Capture (LOC or Functional Justification) as a means of performing AC Scan. There are, however, advantages to the use of Functional Justification on a DFT-Focused tester the reasons for which will be pointed out. Nevertheless, it is possible to successfully use LOLS on such a tester if the requirements and limitations are fully understood.

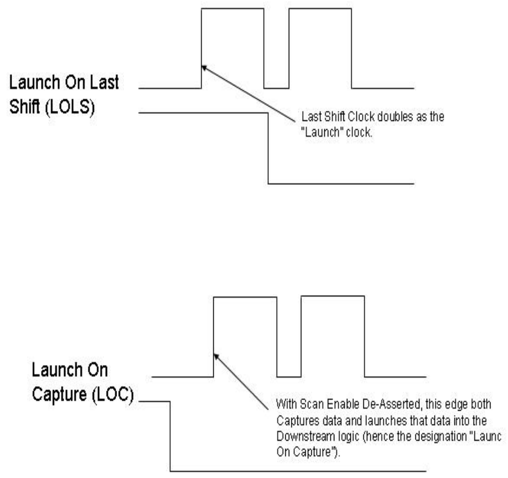


Figure 3: LOLS and LOC Scan Enable Timing

Precision Requirements for Scan Enable

In the case of Launch Off Last Shift (LOLS) in a Mux-D environment, Scan Enable must be de-asserted following the Launch clock (last shift clock) and before the corresponding Capture clock. Furthermore, setup and hold requirements must be met, further constraining the time at which SE can be de-asserted.

Just how narrow is this window? SE cannot be lowered before latest data output by either a PI or a scan cell, and it must be in place before the capture clock (see Figure 3). If, for instance, we want to check for 300 Mhz operation, the window during which SE must be lowered is 3.3ns minus any clock or data skew, current data hold time, and new data setup time. This might result in a window of 2.5ns or less. So the precision requirement for SE in this case might be as tight as plus or minus 1.25ns.[4] As the Launch-to-Capture timing decreases, so does the

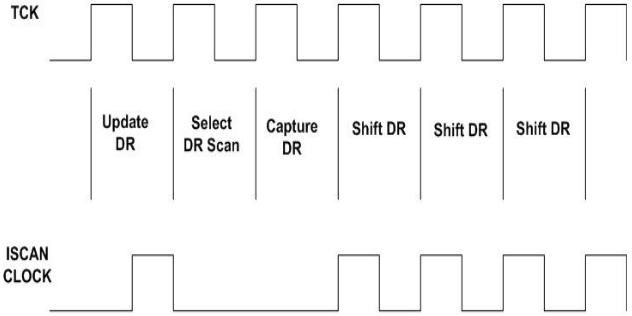
---

[4] Scan Enable may be independently programmed for each clock domain. This paper will not discuss the requirements for inter-domain testing.

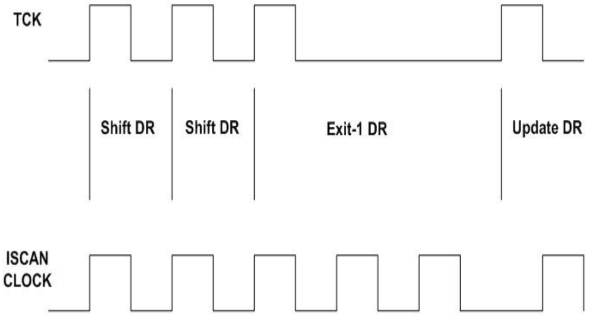
probability of being able to properly place the Scan Enable signal.

The authors recommend two possible options for handling this problem:

1. Use Functional Justification – since no particular precision is required of Scan Enable in this case, it is much easier to properly use.
2. Use an Edge-Triggered/LSSD hybrid flip-flop as a scan register. Since this type of flop does not utilize a Scan Enable, no precision timing is required. (NOTE: in this case, the "Launch" occurs coincident with the "B" clock, and the Capture is normally coincident with the system "C" clock. As with Scan Enable, these may be generated by different clock generators in each clock domain.



Synchronizing ISCAN and BSCAN Launch/Capture Cycles and Shift Data Out/ In:Note that ISCAN launches on the rising edge of the clock while TCK launches on fhe falling edge of the clock.



Synchronizing ISCAN and BSCAN Shift Clocks to the next Launch Cycle: The Assumption is that the BSCAN TCK will have to be paused after it has shifted out all of the Boundary Register data until all of the ISCAN data is shifted out and caught up. Note the Launch synchronization with the falling edge of TCK in the Update-DR state.

Figure 4: AC Scan Synchronization of BSCAN to ISCAN

Synchronizing PIs with the Launch Signal

In order to be effective, the Primary Inputs (PIs) must apply their data synchronized to the Launch clock edge. However, a DFT-Focused tester can only change the data on its primary drive/receive logic at the beginning of a vector cycle. In both Launch Off Last Shift and Launch Off Capture, however, the "Launch" occurs at some point during the vector cycle, but not at the beginning. Therefore, it is not possible to synchronize the PIs with the data launch from the internal scan registers. For tests with an excess of Launch-to-Capture timing margin, this may not prove to be a problem. However, in many cases, this can result in incompletely tested areas of logic.

The authors recommend two possible solutions to this problem:

1. Provide either an I/O scan ring or a boundary-register that is synchronizeable to the Internal Scan clocking as the source of the PI data.[5] Note that this solution works only with the Functional Justification test method. The reason for this is that in LOLS, Scan Enable is still asserted when the "Launch" occurs. An asserted SE blocks the signals coming from the I/O logic because the I/O multiplexers are still routing the signal to the scan chains, and not the functional logic.
2. Provide a second test vector set that is used to verify only the I/O shadow logic. In this case, Scan Enable can be held low, and the PIs will be directly connected to the functional logic.

Note that the Primary Outputs (POs) do not suffer from this problem. This is due to the fact that they must be synchronized to the Capture clock, which occurs with SE low. Therefore, the POs are always connected to the functional (I.O shadow) logic when the Capture occurs.

PI and PO Synchronization with ISCAN

As has already been pointed out, it may be difficult to synchronize the PIs and POs with the Launch and Capture of data in the internal scan chains. The authors recommend the use of a boundary-register with a set of specially modified BC_1 registers. The modifications allow for the Update and Capture clocks to be synchronized to the Launch and Capture clocks supplied for the ISCAN chains.

---

[5] There are special boundary-register cells designed to scan in using TCK and to Launch and Capture using the internal system clock (see ref. [7]).

There are several such modifications suggested in the literature; however, the one referenced by the authors works best when used with the Functional Justification test technique[7].

Clock Pulse Requirements

In order to provide the necessary Launch to Capture timing, the Launch clock pulse must not exceed (approximately) 50% of the duration between Launch and Capture. So, if the Capture event is 3.3ns following the Launch, the Launch clock should have a pulse width of no more than 1.65ns. No such restriction exists for the Capture clock, however.

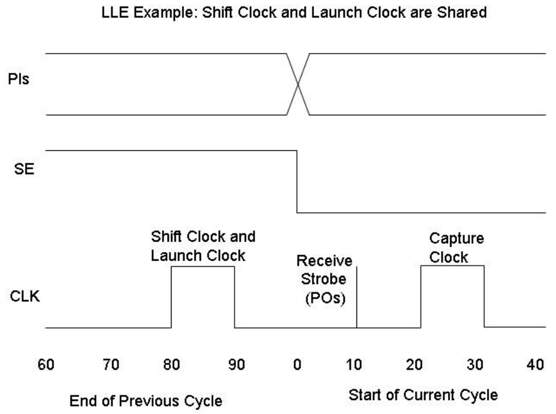


Figure 5: Typical LLE Launch On Last Shift Timing

Now, id DFT-Focused test, the assumption is that there are two clocks involved; one Launch clock and one Capture clock. Since these are the only two clocks available, one must do double-duty as the shift clock as well.

Since commercial ATPG software places the Launch clock late in one period, and the Capture clock early in the next period (see Figure 5), and since the shift clock must be associated with one of these, there are two clock formats that may be applied to shift, launch, and capture. They are:

1. Late, Late, Early (LLE) and
2. Early, Late, Early (ELE).

The clocking scheme defined as:

<shift clock>, <launch clock>, <capture clock> so that Late, Late, Early tells us two things. First, the Shift clock and the Launch clock are both the same and that they appear late in the vector cycle. The Capture clock stands alone and appears early in the vector cycle.

ELE, then, tells us that the Shift and Capture clocks are the same, and that the Launch clock stands alone and is late in the clock cycle.

Why is this important? In the LLE format, the shift clock pulse width will be the same as the pulse width of the Launch clock. This may be very narrow for high-speed tests. If timing closure and simulations are not performed to verify that the scan chains will function properly with a narrow clock, then there may be setup and hold violations discovered during test debug.[6]

The ELE format does not suffer from this problem, but if Launch Off Last Shift is chosen as the test method, the last shift pulse (the Launch pulse) will still be narrow and therefore a potential problem source. Be that as it may, the ELE approach begins to lose effectiveness as the Tpd decreases and therefore for frequencies above a few hundred Mhz, it may prove impractical. Thus, there may be no tester-driven adequate solution for high-frequency (i.e. low Tpd) delay fault testing. Therefore, for this type of test, the authors recommend utilization of Phase Locked Loop Synchronization.

Phase Locked Loop Synchronization

Our investigation shows there to be a definite movement in the industry toward the use of programmable PLLs to provide the Launch and Capture clocking[6,11]. This is being done for a number of reasons, the primary of which seems to be the need to be sure of the quality of the delay test when fast system clocks are involved.

A programmable PLL not only provides the double pulse timing, but should also allow for the programming of an arbitrary number of clock pulses (to support a variety of other test methods, among them functional verification using ISCAN – a technique not discussed here). The PLL should also provide a means for programmatically varying the Launch to Capture timing. Varying this timing is necessary for schmoo and certain other tests. The traditional method for accomplishing this is the varying of the input clock frequency to the PLL (as

---

[6] This can occur, for instance, if both edges of the clock are used or if the hold time of lockup latches becomes so narrow that they are unable to perform their function.

provided by the tester). However, the range and accuracies required for a global tester-provided solution of this sort might prove expensive. Therefore, a fully programmable PLL is highly recommended.

A programmable PLL normally outputs the double pulse when signaled by the lowering of Scan Enable (though other trigger signals can be used as well). Programmable PLLs designed for use in an LOLS environment must also provide for the control over the Scan Enable signal as well. This is due to the fact that the tester is unaware of when the double pulse will occur and thus it must give up the control over Scan Enable in to the PLL. Thus, if LOLS is being used, the PLL will provide a synchronous Scan Enable signal that must be precision routed to the scan flops.[7]

Also in conjunction with the programmable PLL, a synchronous scan ring (I/O scan ring consisting of either a synchronizable boundary-register or a separate scan chain circumnavigating the I/O logic) should be used. Since the placement of the Launch and Capture signals are unknown when a programmable PLL is in use, the I/O shadow logic may be untested unless this is provided for.

In the case of LOC, no precision timing requirement exists. Therefore, in this case, the PLL need not concern itself with Scan Enable.

Once again, the authors recommend that the Programmable PLLs be programmed using the IEEE 1149.1 Test Access Port. This satisfies the “Load Slow” requirement without putting any significant burden on the designer. Up until recently, ATPG tools had not been designed to support programmable PLLs for AC scan. However this situation, along with the availability of Programmable PLLs has been rectified.

LBIST

While DC scan and memory BIST are widely used by the industry, as well as by the participants of this study, Logic BIST is still in an early phase. The main three arguments reported for using logic BIST are:

- It provides the ability to do in-system self test (on the PC Board)
- it helps reduce test data volume
- it facilitates reduced pin count testing.

While some home-grown solutions exist, these seem to be abandoned especially where LBIST is used for manufacturing test. Within organizations where LBIST is used, it is not used for all devices. For manufacturing test, LBIST is often (but not always) found together with DC scan. LBIST is in general well suited for DFT focused testers. Other than what has already been discussed with respect to clocking for scan and Memory BIST, no additional impediments were found with respect to how LBIST has been implemented.

One issue to keep in mind, is diagnostics using LBIST. For debugging and failure analysis purposes, various LBIST diagnosis routines typically use similar mechanisms. Typically, the LBIST is broken into small segments and a signature is collected for each segment. The test is run and scan chain data is then collected for each failing segment. While efforts have been made to reduce the time it takes to isolating failing patterns [1], such a process typically requires either large amounts of data, or multiple iterations of test. In many cases, this can all be greatly simplified by having a direct interface between the test equipment and the EDA equipment.

Scan compression techniques

Over the last couple of years, several techniques have been introduced that promise test coverage equivalent to ATPG, with significantly reduced data volume and time. These techniques have similarities to scan and/or BIST and commercially introduced solutions include Swarthiest [2], Embedded Deterministic Test [2], and Deterministic BIST [4]. They all target replacing DC and/or AC scan tests with a more compact test set. These techniques are aimed directly at reducing test cost by reducing test data volume and time. Therefore, the most significant test cost reduction can be achieved when these techniques are used in conjunction with a lower cost, DFT-focused tester.

While one might consider DFT-focused testers and scan compression DFT techniques competitors, these two technologies are a good match. Some DFT-focused test systems are designed to support

---

[7] Basic requirements of a Programmable PLL are: It must provide a glitchless clock, and it must remain locked and running during a test. Since Scan Enable is asynchronous to the clock itself, synchronization must be provided internally by the SoC to insure proper test execution.

large amount of scan test data, which might no longer exist with these compression techniques. But if Moore's Law continues to be true, a combination of both a low-cost tester and scan compression might be just what is needed.

All of the three commercial scan compression techniques are recently introduced, and none of these was used at any of the organizations visited. Based on published material, however, there is nothing that prevents using these DFT methods on a DFT-focused tester. The requirements are similar to those for scan and LBIST.

Summary of I/O Architectural Guidelines

In summary, the Nine Key Learnings from this experience are:

1. Include the IEEE 1149.1 interface if possible. It is valuable for a variety of reasons beyond those mentioned in this paper.
2. MBIST should be controlled by the 1149.1 TAP in a Load Slow/Run Fast environment. The TAP clocking via TCK provides the separate clock environment often needed by DFT-Focused systems.
3. MBIST failure information should be scanned out either via TDI as a data register or accessed through a dedicated scan chain. It is possible to provide multiple memory-map data package through this technique, thus eliminating the need for repeated executions for every failure detected.
4. If scan design requires the crossing of multiple clock domains by a single chain, use hybrid LSSD-Edge Logic flip flops as scan registers if possible. The safe clocking of such registers is invaluable in this case.
5. For those cases where Reduced Pin Count Testing is to be performed, the use of the IEEE1149.1 Boundary-Scan utilizing a synchronous Boundary-Register (a boundary-register cell that can be synchronized to both the boundary-scan chain and the internal scan Launch/Capture clocking) is recommended. This will require the design of a special BC_1 cell.
6. As Launch/Capture timing decreases, ELE clocking may be preferred over LLE clocking, though it implies the use of Functional Justification as an AC scan methodology.
7. When using a programmable PLL, the Scan Enable signal must be controlled by the PLL like a normal clock signal during Launch off Last Shift tests. Similarly, without the use of a synchronous I/O scan ring (i.e. boundary-register), the I/O shadow logic cannot be tested in this situation.
8. The authors recommend that all programmable tests be controlled by the IEEE 1149.1 Test Access Port (TAP). This includes MBIST, LBIST and any tests involving the loading and/or unloading of parameters. Once again, the load-slow/run-fast capability of the TAP is invaluable in these cases.
9. When designing devices with DFT, the designers should consider the characteristics of the DFT-Focused test equipment that might be used to test the SoC.

Where Do We Go From Here

The authors are now just beginning to look into more detail the implications to the design that follows from the use of a Low Cost DFT Tester. Experiments are currently being planned that will look into the fine aspects of AC-Scan (both transition fault and delay fault) testing, direct linkages to failure analysis, and early system Characterization. The readers are invited to get in touch with the authors as described below to discuss current research, exchange ideas, and in general converse on the DFT test techniques that ultimately will significantly reduce test costs.

The authors may be reached at:

ken_posse@teseda.com

geir_eide@teseda.com

References:

[1] "**Directed-binary search in logic BIST diagnostics",**
Kapur, R.; Williams, T.W.; Mercer, M.R.; Design, Automation and Test in Europe Conference and Exhibition, 2002. Proceedings , 2002 Page(s): 1121 -1121

[2] "**A SmartBIST variant with guaranteed encoding"**
Koenemann, B.; Barnhart, C.; Keller, B.; Snethen, T.; Farnsworth, O.; Wheater, D.; Asian Test Symposium, 2001. Proceedings. 10th , 2001 Page(s): 325 -330

[3] "**Embedded deterministic test for low cost manufacturing test"**
Rajski, J.; Tyszer, J.; Kassab, M.; Mukherjee, N.; Thompson, R.; Kun-Han Tsai; Hertwig, A.; Tamarapalli, N.; Mrugalski, G.; Eidel, G.; Jun Qian;
Test Conference, 2002. Proceedings. International , 2002 Page(s): 301 -310

[4] **DFT Compiler SoCBIST Deterministic Logic BIST Technology Backgrounder**. Available on http://www.synopsys.com/products/test/dft_socbist_techbgr.pdf

[5] "**The EDA Report, Sanitized Version**", an internal report for Agilent Technologies authored by Ken Posse, Ken Parker, John Algiere, Lynn Schmidt, Dick Toftness, 1998

[6] **Trip Reports of the DFT Interviews of Various Leaders in the SoC Industry**, Interviews conducted by Ken Posse, Geir Edie, Andrew Levy, Pete Decher, and Ken Klebart, Internal Teseda Corporation Documentation, 2003.

[7] "**The Use of the IEEE 1149.1 Boundary-Scan Standard in Design-For-Test (DFT) Based IC Testing**", Ken Posse, Test Resource Partitioning Workshop 2002, pp. 3.2-1 to 3.2-12

[8] **CTL For Test Information of Digital ICs**, Rohit Kapur, Kluwer Academic Publishers, 2003

[9] **"On programmable memory built-in self test architectures"**
Upadhyaya, S.J.,Zarrineh, K., Proceedings of the Design Automated Test Conference Europe, 1999, Page(s): 708 -713

[10] **"An effective distributed BIST architecture for RAMs" ,**
Prinetto, P.,Di Natale, G.,Di Carlo, S.,Chiusano, S.,Benso, A.,Lobetti Bodoni, M., Proceedings of the European Test Workshop, 2000, Page(s): 119 -124

[11] **"At-speed testing of delay faults for Motorola's MPC7400, a PowerPC ™ Microprocessor",**
By Raina,R.,Pyron,C.,Molyneaux, R.,Tendolkar,N., Proceedings of the VLSI Test Symposium 2000, Page(s): 3 -8