

Trends in manufacturing test methods and their implications

Sandip Kundu, T. M. Mak, Rajesh Galivanche

Design Technology, Intel Corporation

Contact: Sandip.Kundu@intel.com

Abstract

Driven by market applications in the areas of computing, networking, storage, optical, wireless, portable, and consumer electronics, semiconductor chips today are as diverse as ever. Confluence of multiple applications and rapid integration has also driven the heterogeneity of chips. Test methods have evolved with the products. However, the basic goals in testing remain the same: quality of product, recurring and non-recurring costs and time to market. In this paper we try to catalog some commonly used test methods, identify their associated DFT requirements and trends in terms of tester requirements. Given the diversity of semiconductor chips today such as various PLDs, volatile and non-volatile memories, analog, mixed signal, FPGA, ASIC, SOC, MEMs and processors, it is impossible for a paper of this nature to be fully comprehensive. So we limit our focus on processor, ASIC and SOCs.

1 Emerging Trends

In the following section we observe some noticeable trends in semiconductor industry that affects how chips will be tested.

1.1 Hyper-Integration

In every segment of the semiconductor business, form-factor, power and cost savings is driving significant level of integration. According to published reports 75% of SOCs in 2006 will be mixed signal [1].

Test can no longer take advantage of the traditional method of using optimized equipment for particular circuit types. For example today's CPUs are SRAM centric where more than 2/3rd of the transistors belong to large on-die caches [2]. Due to the signaling and bandwidth mismatch of the CPU IO (more on that later) with that of the tester, these products cannot take

advantage of algorithmic pattern generator on large VLSI tester. Testing on chip memory with memory testers also suffers from the same limitation. Also, to test a die with two or more different testers will require multiple socketing. This has dire cost implications in terms of resource requirements and material flow handling. In such a scenario, DFT must bridge the gap between target circuit type and the tester capability. This learning is easily extended to mixed signal chips as well.

In a mixed signal chip, analog complexity may vary from relatively low performance baseband up to and including multi-gigahertz radio frequency (RF) applications. In addition to logic and analog circuitry, a SOC design may even contain embedded volatile and/or non-volatile memories. Multiple test socketing is a feasible solution only if the alternative DFT solution proves too costly.

1.2 Polyolithic integration/System in Package (SiP)

In some applications, the process requirements of various blocks are so different that monolithic integration does not make sense. An example may be integration of high-speed digital logic with high density flash memory or high speed digital logic with RF receiver circuitry. Mixed technology multi-die packaging carries unique challenges of accessibility and debug-ability because many a times the dies may come from different suppliers. Polyolithic integration may hide accessibility of constituent individual dies. This complicates testing and may require both a DFT solution as well as alternative test approach for individual dies. Furthermore, testing the behavior of the die interfaces may involve more than just electrical continuity test as interface signal integrity problems may manifest during functional operation only.

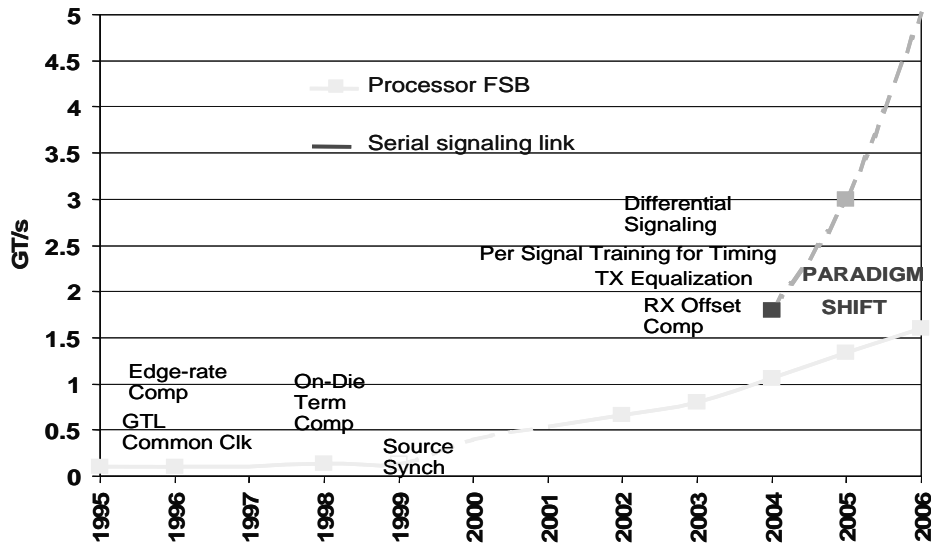


Figure 1 Device IO Interface and tester transfer capability Trend

1.3 High speed differential links

High-speed serial interfaces have transcended their traditional communications market and crossed over into the microprocessor, ASIC and System on Chip (SOC) markets. Application of high-speed multi-lane interconnects in processors, disk-drive controllers and memories have impacted how DFT in testing is conducted in these applications. We will expand this point in our next section.

1.4 Testing with modulation of test environment

With the advent of designs that contain environmental sensors to modulate the chip behavior [3], we are seeing the dawn of a significantly new set of test problems. These on-silicon sensors observe temperature, frequency and voltage problems, allowing additional circuitry to somehow compensate for some behavior of the die in a controlled fashion. Temperature sensors detect when the die is too hot or too cool and respond by changing a combination of voltage, clock frequency and, sometimes, even the die functionally. Similarly a voltage sensor may detect over or under-voltage situations that require change in clock frequency. Clock frequency detectors may sense the out-of-range frequencies that a customer might impose on a die via over-clocking that may cause stress-related reliability problems. There has been a significant growth in these types of the sensors in microprocessors and it may spread to other chips in portable applications as well.

We expect that, in the near future, these sensors will start controlling more local die regions instead of controlling a single global aspect across the whole die. *This requires more asynchronous testing between the tester and the die and prods us to think about alternative interfaces.*

1.5 Power Dissipation

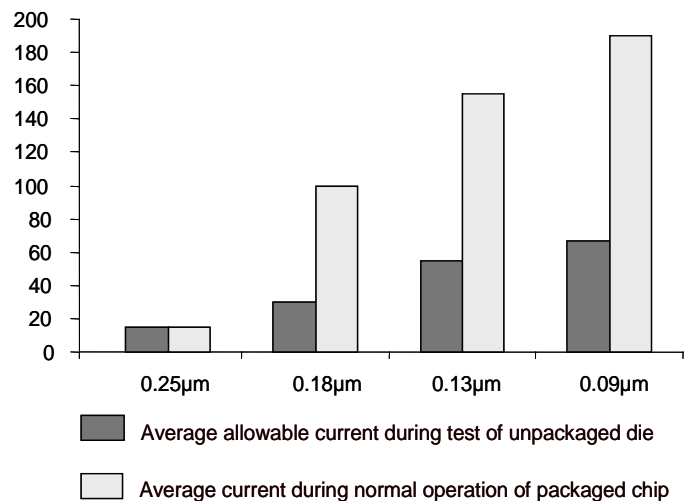


Figure 2 Power availability (shown as Amps in Y-axis) during wafer testing

The power dissipation is already a significant test problem. Exponential rise in transistor count and simultaneous increase in frequency, more than offset any benefit from reduction in Vdd. In fact, power consumption has grown exponentially for microprocessors throughout the last decade. A typical C4 contact may only be good for an average of 50mA of current delivery to the chip so a large array of C4 bumps (thousands of bumps) are needed to supply the necessary current needed for the chip to operate. This becomes a problem in wafer testing, where probe needles deliver the power to the chip. Many chips today already consume >50A of current during normal operation. Thus, it will be necessary to design probe cards with thousands of probe contacts. Given the fine contact pitch and the force required to create an ohmic contact with a C4 bump, this is not achievable from a mechanical point of view. Furthermore, these probe pins have significant inductance which naturally limits the in-rush current (di/dt) whenever there is a sudden change in activity levels. Therefore, there will be a limitation on available power. Power supply noise issues during wafer test and DFT strategy must comprehend this limitation thoroughly.

1.6 Reliability screening

Test process is not only responsible for screening of manufacturing defects that affect device functionality and performance but must also address other issues that result in customer perceived Defects per Million (DPM). A portion of the test flow has to be dedicated to the acceleration of latent defects that do not appear as test failures but would manifest as long-term reliability failures. Unfortunately, leakage current goes up with rising temperature (Figures 3, 4). Thus, when thermal acceleration is used during burn-in, increased leakage may cause rising temperature, which in turn causes increased leakage, setting off a positive feedback cycle known as *thermal runaway* problem. To eliminate any thermal runaway problem, burn-in temperature must be capped. In each technology generation, the capped temperature goes down, reducing effectiveness of thermal acceleration.

A similar situation occurs with gate oxide leakage as well. In geometries below 0.09µm, gate oxide leakage constitutes a rising percentage of overall leakage (reported to be as high as 50% of overall leakage in some 0.09 µm processes). When voltage acceleration is used, it causes leakage to rise exponentially, and also contributes to rising cases of gate oxide failure. In order to avoid these problems, maximum stress voltage has to be limited. Thus, present acceleration methods for reliability screening come up short for addressing future

needs. When effective reliability screens are not available, ability of the design to withstand latent defects has to become an integral part of DFT.

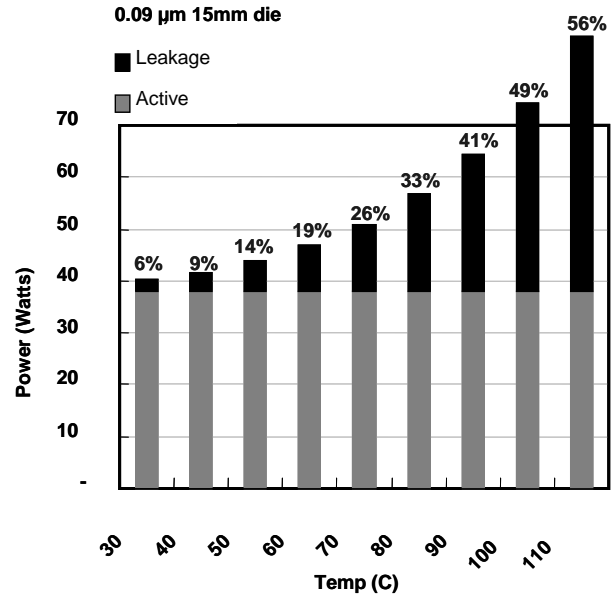


Figure 3 leakage and active power against temperature

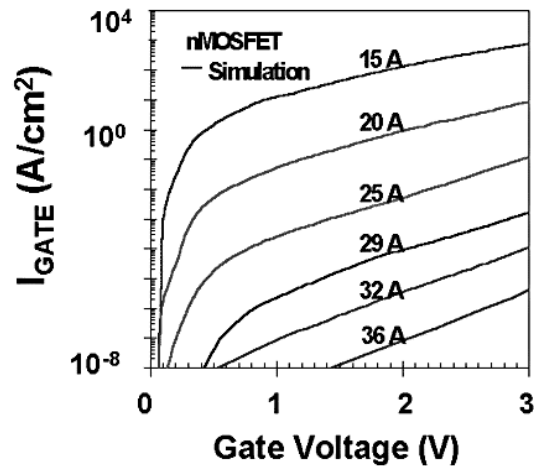


Figure 4 Gate leakage current against gate voltage (family of lines corresponds to gate oxide thickness in Angstrom)

1.7 Yield loss

At higher frequencies, overall tester timing accuracy (OTA) may cause significant yield loss from tester inaccuracy. Unless, test system timing accuracy improves in tandem with device speed, alternative test

methods become necessary, as has been the case with testing faster I/Os in PC platform chips.

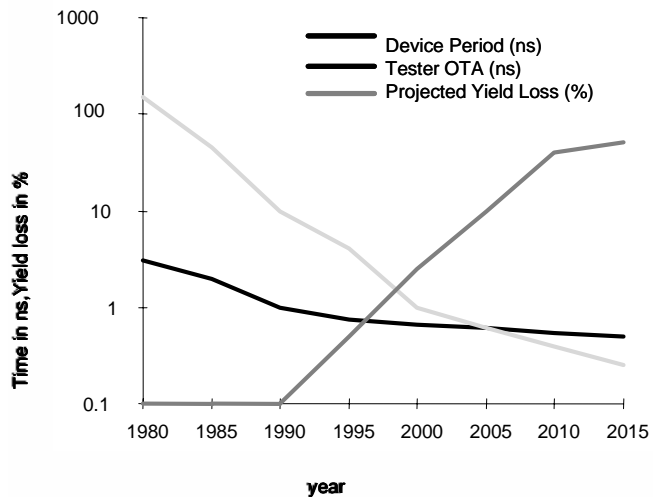


Figure 5 Yield loss projection due to OTA

This problem is somewhat mitigated by the move to DFT enabled IO testing (see later sections). For structural testing, the only accurate timing needed is that of the system clock. Most processors or large SoC use the tester supplied system clock to generate an internal core clock, which can be many multiples of the system clock. Hence, inaccuracies (or quality) in tester clock can be magnified to affect functionality or speed binning.

1.8 Manufacturing test cost

Test equipment capital cost has been a catalyst for change in some DFT practices [4], which in turn has lead to other innovations in the area of testers.

2 Test Techniques

In this section we discuss various test method that are in use today to address common test problems. Some terminologies used in this section assume familiarity with the subject. At a philosophical level, there are two common approaches to testing.

Functional testing mimics the system level behavior with inputs and outputs controlled and observed by a tester. *Structural testing* deploys divide and conquer approach and tests sub-circuits by using an alternate mode that vastly improves controllability and observability of the internal signals of the signals. Structural approach often requires special circuitry (Design for Test or DFT) but DFT also enables bridging performance gap between device under test

(DUT) and tester hardware as well as enable automation in test pattern development. Over the years, DFT techniques have been developed to do functional testing with less capable testers. Similarly, improved automation of test pattern generation, with supporting DFT hardware, allows for some speed test as well as cycle limited pseudo-functional testing. It is impossible to discuss all possible variants of structural and functional test methods in this paper and hence we will summarize the major test techniques that are in use today.

2.1 Functional test

Lack of automation in functional test pattern generation, large pattern development time, debug cost and need for full speed testers have been cited as reasons to move away from function testing to structural testing. However, functional tests are more effective in testing at speed, under realistic electrical noise (signal as well as power supply noise) and sometimes are the only way to test when problems with skew in clock distribution, non-functional states, clock gating, contention, initialization problems render stuck-at fault coverage ineffective or yield loss problems insurmountable.

Full speed, broad side functional test requires the tester to match the device behavior completely. In view of the latest product features (processor with large cache), new functional test methods that execute codes out of the large cache have been developed [9]. The test codes can either be generated offline and then loaded into the cache, or it can be generated on chip with the processor itself. This would require an access mechanism to load the cache, and that the processor also needs to be designed in such a way that it can execute out of the cache directly. Tester requirements have been reduced substantially to match that of the test port for loading the cache.

While the above approach allows cache based designs to apply functional tests from less capable tester, such test must avoid explicit data transaction with tester during functional execution and all data transactions must be cache bounded [9]. However, special on-die DFT and queues can allow pseudo-external transaction, further improving the quality of such functional tests.

Pattern synthesis technique must make sure that such tests are consistent and deterministic [9]. The test pattern generator can be embedded in the initial data that is loaded on to the cache allowing for self-test in functional mode. This effectively increases the test pattern volume without increasing tester storage.

2.2 Scan test

Scan testing is usually supported with special scan tester channels, where each channel is backed by scan memory that currently ranges from 64M to 1024M. Scan testing is also possible with broadside functional tester. However, it may be *limited* by the amount of vector memory behind the functional tester channel (typically, 4M to 32M) since this memory has to be shared with any functional patterns needed by the device under test. Since scan shifting is usually done at lower than core speed, tester performance may not be critical. Launch/capture clocks are usually generated on-chip, with control driven through the test port (JTAG TAP or otherwise).

The key challenges in scan testing are managing power during wafer testing, figuring out how to improve speed related failures with ATPG patterns without causing significant yield loss due to non-functional paths, managing test data volume, and keeping down the test application time. Some of these issues may be addressed by layering logic built-in self-test (LBIST) on top of scan. In Figure 6, we have captured the issue with growing test pattern count problem that invariably translates to test application time issues.

To reduce test time, one trend is to increase the number of chains and shorten the length of each scan chain. This certainly increases the number of needed tester channels.

2.3 LBIST

LBIST tests the logic portion of a die with pseudo-random patterns generated by a Linear Feedback Shift Register (LFSR) and compresses response into a signature using a corresponding Multiple Input Shift Register (MISR). This also involves setting up of LBIST with test mode (usually through JTAG TAP port) and signature unloading afterward (through the same port). Other than the setup and unloading of the signatures, a clock is all is needed. The tester requirements are similar to ordinary scan test.

LBIST is intrusive on design. Unlike ATPG patterns, pseudo-random patterns are unconstrained. This may lead to tristate contention. All designs are not suited for pseudo-random pattern testing. Additional control and observe point insertion becomes necessary to improve fault coverage when fault coverage from pseudo-random patterns is inadequate. Memories need to be well isolated. For multicyle patterns, even memory isolation is not adequate. Additional DFT is needed to have predictable multicycle memory reads. Clock gating, un-controlled inputs, presence of non-scan latches require additional DFT/isolation for producing deterministic predictable signature. These problems are typical, especially in custom designs and need to be addressed through the last stage of design phase. DFT intrusion at this stage adds to design time and cost. Thus, while appealing for its apparent value, adoption has not been overwhelming in the majority of designs.

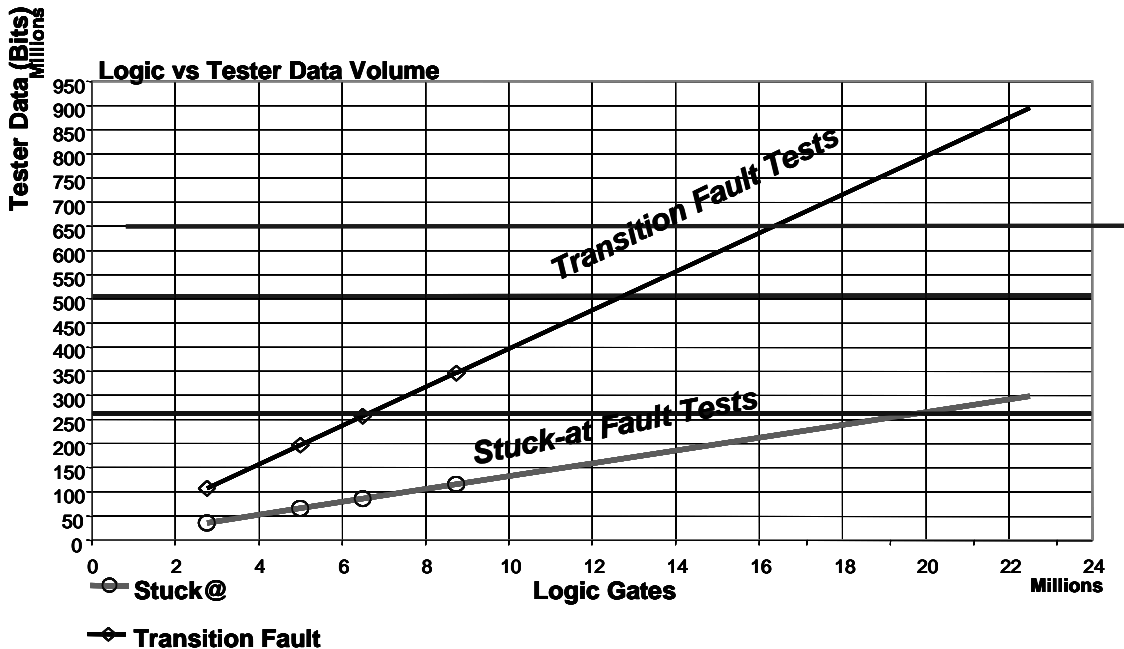


Figure 6 Tester data volume against logic gate count

2.4 On-Chip Test Data Compression

Another technique that is recently gaining widespread exposure is the use of on-chip hardware that does partial generation of the stimulus and signature generation of the response [13][14]. Unlike Logic BIST where the entire pattern is generated using the on-chip LFSR, test data compression hardware on-chip works in conjunction with ATPG tool. It exploits the fact that for a vast majority of the faults, only a small number (<1%) of the scan chain elements need to be loaded with specific states while the rest of the elements can be loaded with any random state. These random states are then generated by the on-chip hardware, thus, reducing the test data volume that needs to be driven from an ATE. The ATPG tool works in a special mode where it determines the specific scan chain elements and the associated logic states (a.k.a care bits) that are necessary for the detection of a fault and also avoid or minimize bus contention and other illegal states in the design. The advantage of this scheme is that it lets efficient usage of ATE bandwidth for necessary data that must come from tester memory and allows generation of majority of the scan test data using high speed on-chip hardware. The result is that we not only alleviate ATE memory requirements but also reduce the test application time. However, the design must ensure that the number of care bits is small and the chip does not generate unspecified logic values (due to uninitialized memories etc) which makes test data compaction or signature generation unwieldy.

2.5 Cache or embedded memory Test

Large memory arrays are very common in today's processors and SoC. Due to the sensitivities of the RAM cells and the arrayed nature of the design; large embedded memory arrays usually require a large set of test patterns to detect subtle faults that may affect the functionality of the array. Also, due to the embedded nature of these RAM, external testing with the tester may or may not be possible. External testing with a tester requires the tester to have a programmable APG (algorithmic test pattern generator) and to be able to run at the same internal frequency of the memory. That may not be possible if the chip runs faster than the tester. To work around this limitation, DFT has been used to pipeline the test patterns into the chip and apply them at speed, after an adequate number of patterns have been buffered. This method of testing is not very efficient but may work for a small array.

A more sophisticated approach of this DFT scheme incorporates a microprogramming machine (MBIST: memory BIST, PBIST: programmable BIST) that can

generate different patterns needed for testing the array. In PBIST/MBIST schemes, instead of the full test patterns, only the programming codes are loaded from the tester onto the microprogramming machine. The actual test patterns are generated by this internal machine at full device speed. To make this more effective, modern MBIST or PBIST engines also have provision for hooks to allow rastering [7] (showing the failure locations) so that repair algorithms can be used and the appropriate fuses can be programmed to replace the faulty rows, columns or blocks. To facilitate repair, computing controller on the tester must be capable of running the repair algorithms in real time without slowing down testing. Alternatively, offline fusing is also possible, but this requires additional socketing of the device under test and adds significant cost.

2.6 Parametric Test

Chip level parametric test includes power consumption test (or Idd test) and standby current test. Most other parametric tests are tied to chip's IO and will be detailed below.

Power consumption tests are conducted with functional patterns that are intended to maximize chip level activities. Since this current is usually large and fluctuates with time, a power supply with internal current monitor is preferred over an external PMU (Precision Measurement Unit). In contrast, the standby current is much lower and it makes sense to have the measurement done with an external PMU. To measure standby current, a functional pattern that conditions the internal circuits (to put the chip in a low drain mode) is applied before the measurement unit is turned on. Since the dynamic current and standby current can vary over several orders of magnitudes, splitting the PMU range is necessary so that the precision of measurement is not skewed by the range. The system clock may or may not be stopped for this test (depending on how the devices work during the various standby modes). For products that require Iddq test, this will be similar to the standby test. The clock will be stopped at specific vectors with a given Iddq pattern for Iddq test. Iddq current needs to stabilize before the measurement is taken. This slows down testing. A balance needs to be struck between the precision of Iddq test and test throughput. In order to maintain test throughput, capture defects and avoid killing potentially good devices, appropriate design techniques for both the chip as well as the Test Interface Unit (TIU) are essential to dampen the current waveforms.

Today's processors have standby current in the order of 25Amps and dynamic current in the range of 100

Amps. Iddq defect detection capability diminishes with higher background current.

2.7 IO Test

IO testing involves qualifying a large number of specified parameters. These specifications are roughly separated into DC parametric and AC parametric tests. Traditionally, IOs are tested with testers that can connect to every pin with a corresponding tester channel.

DC test consists of leakage test and level tests (Vin/Vout). For DC level tests, either functional patterns (and a correct functional behavior is observed with the appropriate test drive and compare levels) or DFT (like boundary scan, where digital data are captured and compared) can be used. Leakage tests are carried out by forcing the tester pin electronics while the pins are pre-conditioned to remove pull-ups and pull-downs (if any). AC parametric tests are usually done with application of functional patterns at speed with the tester driving the worst case specification timings and comparing outputs at appropriate intervals specified by delay specifications.

With device performance increase and signaling method going from common clock to source synchronous, new test methods have been developed [5] that utilized the loopback nature of the IO pins (or connecting up input pins to output pins). With AC IO loopback [5], outgoing data is sent back to the input and they are compared with the appropriate adjustment in latency. The timing stress is accomplished with changing the bus clock frequency or with a delay inserted into the loop. Most of the adjustments are usually accomplished with the boundary scan test port along with some form of pre-test or calibration scheme for the on-chip testability circuits such as delay generators. Without calibration or self compensated circuits, on chip delay generators can vary so much that the test results are not sufficient to replace conventional testing with the ATE.

As shown in the trend earlier, the quest of increased system performance has resulted in demand for improved interface data transfer rate. This has prompted the change over from large swing, source synchronous bus to point-to-point, small swing, differential, clock embedded signals. This is essentially similar to the high speed serial signaling technology used in the communication sector. While it may be possible to use conventional testing for a few high speed serial channels on a communication chip, the economics change when the serial link technology is applied for the computing chips. The number of such channels on a

given computing chip will explode and it is not economically feasible to have ATE that can support a large number of such high speed serial channels. The AC IO loopback methodology (or enhancement of it) is a possible replacement solution to conventional ATE. The technology has been extended for many types of IO, including high speed serial, simultaneous bidirectional signaling, etc.

Loopback test can also handle DC parametric test with the control of IO sense amp threshold while the signal loops around to the input. Instead of running at high frequency, one simply has to lower the frequency while setting the threshold to the level desired.

The next set of challenges emerges from the fact that there are more features to be tested beyond the physical layer. As with all communication architectures, link layer, routing layer and protocol layers are all part of the communication stack that also need to be tested. Since these layers run on different clock domains, the application of scan testing may leave out testing of critical timing behavior. If these circuits cannot be tested thoroughly with scan, then functional testing becomes the solution by default. It is impossible to run functional tests without the high speed tester channels and signaling protocol that can match with that of the DUT. Moreover, the handshaking protocol of these interfaces creates non-deterministic behavior for the DUT [8]. This may inhibit any stored stimulus/response testing such as on an ATE. Thus, the transition from common clock bus interface testing to point to point interface testing is more revolutionary than evolutionary. Some form of DFT is needed to enable testing of these highly complex interfaces with low cost test hardware without compromising on testing for these timing characteristics of these signals that is at the core of these IOs.

2.8 Fuses

Fuses are used to configure the chip during and after manufacturing. Fuses are used for redundancy repair, clock tuning or IO compensation and many other applications. A typical fuse may consist of a strip of polysilicon material, which will change its resistance by several orders of magnitude with a strong programming current [6]. A higher level of voltage (and sufficient current capability) may be needed for programming.

With the number of fuses going up in every generation of chips (for compensation, reconfiguration, etc.), test time and area is becoming a significant concern.

2.9 Analog testing

Communication, audio/video interfaces and an array of sensors from automotive, medical and environmental applications are pushing a significant amount of analog content onto predominantly digital chips. Examples include, high-speed networking (10/100 and Gigabit Ethernet), digital subscriber line (DSL), cable Modems, Home Phone Networking (HPNA), Digital Video Interfaces (DVI) and wireless LAN (Bluetooth and 802.11). These functions were previously implemented by using separate chips altogether.

One design characteristics of these mixed signal blocks is that the “authoring” of these functional blocks often takes place outside the “integrator” team. The key test challenge is to test the analog content of the die (often at input/output data converter, receiver/transmitter end of a die) without resorting to multiple test socketing against multiple tester platforms.

Traditionally, analog testing is specification based; hence require specific instruments for testing analog specifications [12]. However, when analog functions are fully integrated into a die, it is uneconomical to use analog test methods that have been optimized for discrete components such as ops amps, comparators, various ADC, DAC, filters, etc. Thus alternative techniques are necessary for testing integrated analog components.

Two most commonly touted approaches are analog built-in self-test [10] and analog feature testing [11]. Each method has its pros and cons. For example, analog BIST requires large area/power overhead while feature testing does not provide a direct path for “trimming” that is often used to tune analog circuits during testing.

These techniques will continue to evolve based on percentage of analog content, required performance levels, stability of the underlying manufacturing process etc. However, a general prognostication is that due to the “availability” of more digital transistors and increasing “variation” of process parameters, analog functions will become increasingly digital pushing analog to the edge of basic data conversion and or transmission/reception. This in a way may take the pressure away from full analog specification testing.

3 Summary and Conclusions

The ever decreasing circuit geometries are enabling us to pack more transistors and consequently more

functionality in a given silicon area. It also enables massive heterogeneity involving circuits and processes of different types. Below is a list of most difficult test challenges that need to be addressed if we are to be successful in future silicon designs.

1. Bridging DFT gap between tester and target circuit types (such as testing embedded flash design using logic tester).
2. Testing for adaptive designs that regulate its ambience with on-chip sensors.
3. AC and parametric tests of embedded modules that are not easily accessible from the tester or isolated from the design.
4. Incorporating end-user reliability enhancement directly into design.
5. Testing of high-speed point to point serial IOs through various layers (physical, link, routing, protocol etc.) of the communication stack.

4 References

- [1] S. Ohr, “Mixed-signal IC layout tools support 'touch and feel””, EE times, July 09 2003, <http://www.eedesign.com/news/showArticle.jhtml?articleId=17408533>
- [2] <http://www.intel.com/pressroom/archive/releases/20040202comp.htm>
- [3] ftp://download.intel.com/pressroom/kits/centrino/enhanced_speedstep_animation.zip
- [4] M. Mayberry, et. al., “Realizing the Benefits of Structural Test for Intel Microprocessors”, Proc. Int. Test Conf. 2002
- [5] M. Tripp et al., “Elimination of functional testing of interface timing at Intel”, Proc. Int. Test Conf. 2003
- [6] M. Alavi et. al., “A PROM Element Based on Salicide Agglomeration of Poly Fuses in a CMOS Logic Process”, IEDM 97
- [7] C. Hampson, “Redundancy and High-Volume Manufacturing Methods”, Intel Technology Journal, Q4, 1997
http://www.intel.com/technology/itj/q41997/articles/art_4.htm
- [8] T. Mak, “How do we test for Adaptive Computing”, Test resource partitioning workshop, 2004
- [9] P. Parvathala et.al, “FRITS - a microprocessor functional BIST method”, Proc. Int. Test Conf. 2002

- [10] G. W. Roberts, A.K. Lu, "Analog Signal Generation For Built-In Self-Test Of Mixed-Signal Integrated Circuits", Kluwer Academic Publishers, Norwell, MA, USA, 1995
- [11] S. S. Akbay, A. Chatterjee, "Feature Extraction Based Built-In Alternate Test of RF Components Using a Noise Reference", 22nd VLSI Test Symposium, pp. 273-290
- [12] L. Milor., "A tutorial introduction to research on analog and mixed-signal circuit testing"; IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, Volume: 45 , Issue: 10 , Oct. 1998
- [13] S. Mitra and K.S. Kim, "X-Compact: An Efficient Response Compaction Technique," IEEE Trans. CAD, Vol. 23, Issue 3, pp. 421-432, March 2004.
- [14] S. Mitra, and K.S. Kim, "XMAX: X-Tolerant Architecture for Maximal Test Compression," Proc. Intl. Conf. Computer Design, pp. 326-330, 2003.