Trends in Testing Integrated Circuits

Bart Vermeulen Camelia Hora Bram Kruseman Erik Jan Marinissen Robert van Rijsinge*

Philips Research Laboratories
Prof. Holstlaan 4, 5656 AA Eindhoven
The Netherlands

*Philips Semiconductors – ATO
Gerstweg 2, 6534 AE Nijmegen
The Netherlands

{Bart.Vermeulen, Camelia.Hora, Bram.Kruseman, Erik.Jan.Marinissen, Robert.van.Rijsinge}@philips.com

Abstract

New process technologies, increased design complexity, and more stringent customer quality requirements drive the need for better test quality, improved test program development, and faster ramp-up at overall lower product cost. In this paper we describe the main industry test trends and recent innovations in testing integrated circuits as they are applied within Philips.

1 Introduction

The semiconductor industry has through persistent effort succeeded in following Moore's law for Integrated Circuit (IC) process technologies. Both feature size and the cost of manufacturing a single transistor have consistently been going down. This allows more functionality to be integrated per mm^2. With decreasing feature sizes and increasing complexity comes a strong demand to improve the manufacturing test. A modern IC contains tens of millions of transistors and wires, each of which can suffer from a manufacturing defect. To bring a new product to the market at a competitive price, it becomes increasingly important to drive the overall product cost down [1].

Test plays a key role in the overall process of bringing a product to the market. Without test it would be impossible to ship quality chips to the customer. Test therefore is for us an absolute requirement for each product. However test also adds to the overall product cost. In [2] an economic test cost model is presented that helps to make trade-offs between test benefits and costs. This model includes test preparation cost, test execution cost, test-related silicon cost, and imperfect-test-quality cost. It does not include the cost associated with the late introduction of a new product on the market, which test can help reduce. If an increase in test cost results in a larger decrease in other product costs, for example by improving the yield or shortening the time-to-market, this would also be very acceptable. That is why we generally refer to our requirements for test as *good*, *cheap*, and *fast*. A test has to be *good* (high defect coverage), *cheap* (Design-for-Test area and test execution time), and *fast* (fast test development, and market introduction). In practice, we need a different trade-off between these three aspects for each product and market segment. This paper describes trends in testing integrated circuits and the practical trade-offs we make to bring a wide range of chips to a number of different competitive markets.

The remainder of this paper is organized as follows. In Section 2 trends in test quality are described. Section 3 presents efforts in Design-for-Testability (DfT). Section 4 outlines the trends in Automated Test Equipment (ATE). In Section 5 trends in both single die diagnosis and rapid yield improvement are described. Section 6 addresses other (miscellaneous) trends in test. We conclude this paper with Section 7.

2 Trends in Test Quality

Process technologies have become progressively more complex. Each new technology node introduces new materials, more process steps and generally pushes the capabilities of manufacturing equipment closer to the limits of what is physically possible. It is not surprising that this trend gives rise to new test requirements. There are two main reasons why the industry requires better tests. These are (1) more stringent customer requirements, and (2) the integration of more functionality using smaller device structures.

- *More stringent customer requirements*.
  While in the 1970s a defect level of 1000 DPM (defective chips per million delivered) was perhaps still acceptable, automotive products nowadays require DPM levels below 10. Moreover initiatives are ongoing to push these requirements even further and go for a 'zero' DPM level.
- *Integration of more functionality using smaller device structures*.
  With the introduction of device structures with smaller feature sizes comes the possibility to integrate more device structures and consequently more functionality on a given silicon area. With each new technology node, the number of single structures (transistors, vias, line segments) that can be put in a certain area size grows with a factor of about two. To keep the same test quality for this area, the test quality level

 0-7803-8580-2/04 $20.00 Copyright 2004 IEEE

for a single structure needs to increase with at least the same factor. In addition, not only does the complexity of chips grow quadratically with the decrease in feature size, also the number of metal layers is slowly increasing. Combined with the smaller feature sizes, this requires that we detect defect mechanisms that could still be ignored in previous process technologies.

Table 1 shows a simplified example to highlight how these two items affect testing. Let us assume that we have a defect type (e.g. an open via) which causes a test escape in 1 out of 10^{12} manufactured structures. Over the decades and technology nodes, we have designed an IC with the same area, but with increasingly more functionality and device structures. As a result the impact of this defect on the defect level of the complete IC rises with a factor of 20 per decade. In addition, the required defect level has become smaller by for example a factor 5 per decade, due to more stringent customer requirements. If no improvements in the manufacturing process or in the defect detection methods are made, this defect type, which could initially be ignored, now has gradually become a quality issue and may end up making the quality level of the IC unacceptable for customers.

Table 1: Quality trends over the past few decades

	Technology	Structures	DPM Impact	Total DPM Allowed
1985	1.0 μm	5×10^4	0.05	100
1995	350 nm	1×10^6	1	20
2005	90 nm	2×10^7	20	4

As a result of these trends, improvements in test quality are clearly required. From a test point of view these improvements can be made by using more advanced fault models for the creation of test patterns and by applying additional test methods. Both options are described below.

2.1 Improved Fault Models

Improving the fault models used is one way to achieve higher test quality. In this subsection we describe the search for better fault models and investigate the acceptance of various fault models by the industry.

In the 1990s, the semiconductor industry successfully used the single stuck-at fault model to achieve high quality. At that point, achieving higher test quality could still be achieved by higher fault coverage given the stuck-at fault model. Presently a single stuck-at fault coverage of 98-99% is not uncommon, while a decade ago people still discussed its usefulness [3]. The single stuck-at model has been the main fault model for the last few decades and there is no indication that this will change in the future. Despite the known limitations to describe real defects such as intra-gate shorts, bridging faults, complex shorts and opens, the stuck-at model is still the most popular fault model in use today. This is due to its strength in ATPG and the fact that both practice and defect simulations show that it is very effective in catching a wide range of unmodeled defects. However this model does not catch *all* defects. As a result even if one would reach a 100% single stuck-at fault coverage this is not sufficient to reach a zero DPM customer requirement, because some unmodeled defects will surely be missed. So improvements in fault models are required and they have also been provided by academia and industry.

The multiple stuck-at fault model is a logic choice to try to further increase test quality. It is relatively easy to generate multiple stuck-at test patterns using an existing single stuck-at fault ATPG core. These patterns are also successful in detecting additional defects. The success of these patterns to detect additional defects can however not be explained solely on the fact that there are defects in silicon that exhibit multiple stuck-at behavior [4] (see Figure 1).

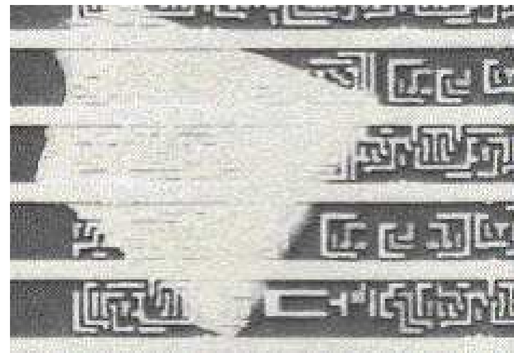

Figure 1: Defect with multiple stuck-at fault behavior.

After all, most of these defects are also detected using single stuck-at test patterns. The main reason why multiple stuck-at test patterns are successful in detecting additional defective chips is because, in order to detect these defects, for each targeted fault, the local environment around that fault is set to several different states while testing for that fault. Testing for a fault under different local circumstances decreases the chance of fault masking. Fault masking can occur if a few neighboring structures, which interact with each other, are affected by the same defect. For the situation in Figure 1 fault masking is however very unlikely to occur, but this example does show that we can improve detectability, if we put the local environment during test in all states that can occur during its functional use. If for all these states the result is fault-free behavior then the IC is free of defects that behave as single or multiple stuck-at faults. Methods

which comes close to this approach are N-detect [5] and Do-Re-Me [6]. These methods are based on the single stuck-at model but now potential faults are targeted multiple times by the ATPG tool with an additional requirement that the stimulation and/or observation conditions are different. For resistive shorts [7] and opens it is easy to show that some defects missed with single stuck-at testing can be caught using this method. Given the pragmatic approach with respect to fault models in the last decades, N-detect seems to be the natural extension of single stuck-at testing for the coming years.

Most defects however are actually not really hard defects (i.e. hard opens or hard shorts), which prevents their behavior from being completely modeled by the stuck-at fault model. In addition, the stuck-at model assumes that a defect only affects a circuit's static fault behavior, which also is not always the case. Stuck-at patterns are usually applied via scan chains at a speed well below the application speed of the IC. As a result, different fault models are needed that have defect coverages that are complementary to the defect coverage of stuck-at faults.

The most commonly used fault model next to the stuck-at fault model is the delay-fault model. Delay-fault testing (also referred to as AC-scan) fills a large gap in defect detectability, as it checks the dynamic behavior of the circuit-under-test, which the stuck-at test does not. In its simplest form a delay-fault ATPG tool creates test patterns with transitions from zero to one and from one to zero for all gates. These patterns can detect defects that have a negative impact on the speed of the circuit-under-test. This includes other defects than stuck-at detects, such as opens, bridges, resistive wires, and resistive vias. Often however these pattern generators are still based on the stuck-at ATPG core and as transitions are harder to generate, the resulting test pattern count for delay-fault testing is typically a factor four higher than for stuck-at testing. Also, typically the fault coverage figures are still well below those of stuck-at fault testing, either due to pattern generation problems, pattern count restrictions on the ATE, or test time restrictions. This is definitely an area in which much progress can still be expected. Especially the introduction of pattern compression removes some of the restrictions in pattern count.

Next to resolving these test generation and ATE restrictions, delay-fault testing itself can still be improved. At present a defect is only detected if the combined delay of the tested path and the defect is longer than that of the critical path. Besides these gross delays, we also need to start detecting smaller delays, e.g. to detect resistive opens, to further improve test quality. This capability to detect small delays is needed, because statistical process anomalies have started to emerge in ICs that are not designed on the basis of the worse process conditions. Practical methods to detect these delays are just emerging [8].

Similar capabilities for resistive shorts exist already for decades in the form of current-based testing [9] or very low-voltage testing [10]. One would also like to have this capability for opens to catch reliability risks or defects that only cause a failure under extreme conditions.

2.2 Additional Test Methods

A second option to improve the test quality level is to develop additional test methods that can supplement the conventional 'stuck-at' test method of measuring single voltages for test responses and comparing them to the expected binary responses. The voltages used are often the nominal voltage for the circuit and ground, with some margin as specified in the functional specifications.

Well-known complementary methods are current-based testing (Iddq) from the 1980s and very low voltage testing from the 1990s. However, process scaling does not favor these methods. Iddq testing is hampered by the increase in the transistor off-state current, which manifests itself as a larger static current for the entire IC due to the increased number of transistors. This increased static current obscures the additional currents due to defects. Very-low voltage testing becomes less effective due to the unequal scaling of the supply voltage and threshold voltage. Therefore most of the attention is focused at finding ways to keep these methods effective. This has led for example to the introduction of ΔIddq testing, which can handle static currents that are 100 times larger than the defect current.

An interesting aspect that has changed in recent years is that advanced test methods for digital ICs have become more and more analog. In current-based testing one does not only want to know if the current is above a certain threshold but also what the actual value is. Likewise in very low voltage testing one wants to know the fail voltage and in delay fault testing the maximum speed. Based on these values and additional information about the chip or its neighborhood on the wafer [11] the pass/fail limits are determined. Moreover, information of two measurements can be combined to allow tighter limit settings. These methods allow one to detect outliers. The values for these outliers are still within the test limits, but are outside the normal distribution of the wafer [12] and can therefore be considered a reliability risk. It is expected that these methods become a standard part of test and could replace expensive alternatives such as burn-in.

3 Trends in Design-for-Testability

A typical modern IC contains tens of millions of transistors and wires and can contain defects that manifesting themselves as opens in and shorts between these structures. In order to test for this, ATE access is provided exclusively via the I/Os of the circuit-under-test, of which we typically only have a few tens or hundreds. There is a gap between the very large number of on-chip components to be tested and the (relatively) small number of I/Os through which test access can be provided. This gap necessitates design modifications to improve the accessibility from the I/Os to the internals of the IC and vice versa. The traditional definition of *Design-for-Testability* (DfT) has been exactly that: additional on-chip hardware, added to the IC design in order to improve its internal accessibility for testing purposes. For this, we distinguish between the ability to drive arbitrary stimuli combinations (*'controllability'*) into and the ability to read responses (*'observability'*) out of specific on-chip circuitry. As the ratio of on-chip transistors over circuit I/Os keeps growing [1], so does the need for DfT.

We distinguish three main roles for DfT. The first role is to enable high-quality testing, achieved by the DfT inside the on-chip modules. The second role is the test access from I/Os to (deeply) embedded modules, in the form of test wrappers and Test Access Mechanisms (TAMs). The third role is the on-chip generation of test stimuli (sources) and evaluation of test responses (sinks). Figure 2 shows these three roles of DfT for a SoC. We discuss these three roles in more detail in the following three subsections.

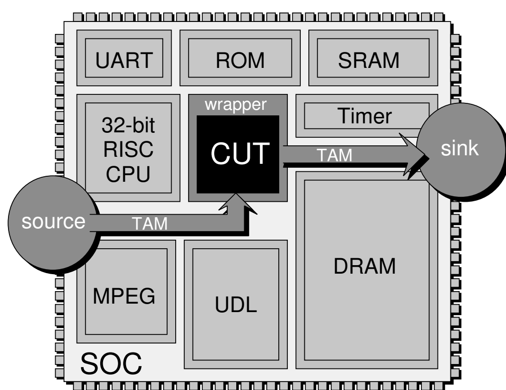

Figure 2: Generic test access architecture for embedded cores, consisting of source and sink, TAMs, and wrapper.

3.1 Module-Internal DfT

The most well-known example of DfT for individual modules is of course *scan design*, in which in a special 'scan mode' all of the module's flip-flops ('full scan') or a subset thereof ('partial scan') are connected into one or more shift registers. Pioneered by NEC as early as 1968 [13, 14], today scan design has become a mainstream design approach for most companies. The success of scan design is related to the fact that it provides controllability and observability, in addition to cutting sequential feedback loops in the circuits. Full scan design enables the usage of Combinational Automatic Test Pattern Generation (C-ATPG) tools, which offer a fast and reliable route to high-quality test patterns.

3.2 Access to Embedded Modules

The larger the IC, the more attractive a modular test approach [15]. Non-logic modules (e.g., memories, analog, RF) as well as black-boxed third-party cores require stand-alone testing. For the remainder of the IC, modular testing can reduce the test generation time, as it presents more digestible chunks of circuitry to the ATPG tools and enables concurrent engineering. At the same time, modular testing facilitates test reuse, which especially pays off in the case of a family of chip derivatives [16]. Modular testing requires an on-chip test infrastructure in the form of test wrappers and Test Access Mechanisms (TAMs) [17]. The test wrappers enable separation of stand-alone testable modules, while the TAMs transport data from IC pins to test wrapper and vice versa.

IEEE Std. 1500 (a.k.a. SECT) [18, 19] describes a standardized, yet parametrizable test wrapper. Some of its features are based on IEEE Std. 1149.1 (a.k.a. JTAG) [20], which is in essence a chip-level wrapper to facilitate board-level testing. Nevertheless, there are also important differences between JTAG and SECT.

- JTAG has a fixed one-bit test data interface, while SECT also supports a scalable multi-bit test data port.
- JTAG mandates two-bit wrapper cells, while SECT also allows both single-bit as well as multi-bit wrapper cells.
- JTAG enforces a fixed operation protocol through its standardized finite-state machine implementation of the TAP controller, while SECT does not have such a fixed protocol.

At the time of writing, the endorsement of IEEE Std. 1500 as a full standard is forthcoming. It is to be expected that IEEE Std. 1500 becomes a widely used standard. This will be the basis of a new range of EDA tools, amongst others for wrapper design, compliance checking, TAM architecture design, test expansion from core to chip level, and test scheduling [21, 22].

Another example of 'wrapper-type' DfT is *Reduced-Pin Count Testing* (RPCT) [23]. The RPCT wrapper, which is

typically shared with the JTAG wrapper, is used to reduce the number of IC pins involved in testing. RPCT enables testing of high-pin-count parts on low-pin-count ATE, or, alternatively, enables testing of multiple ICs in parallel on the same ATE ('multi-site testing'). Multi-site testing is discussed in Section 4.2.

3.3 Stimulus Generation and Response Evaluation

Traditionally, the generation of test stimuli and evaluation of test responses was done exclusively on the ATE, and hence, no DfT was provided for this task. However, with the growth in test data volume, DfT has started to invade this arena as well. We refer to this as *Built-In Self Test* (BIST). With BIST, the need for external test equipment is reduced or even eliminated, as test stimuli and responses can be generated respectively evaluated by the IC itself. BIST also resolves the test access problem to embedded modules, as the stimulus generator and response evaluator can be placed right next to the module-under-test.

BIST was first applied on embedded memories [24]. Memories require large test pattern sets, and hence there is a pressing need to generate them on chip. Fortunately, memory test patterns are very regular in nature, and hence can be generated relatively straightforward by means of an on-chip finite state machine. For medium-sized and large embedded memories, BIST has become a mainstream approach.

An important trend for the future is the growing size of on-chip memory [1]. Both the number of embedded memories, as well as the sizes of these memories is growing. As memories are quite susceptible to manufacturing defects, this trend threatens the yield of these products. Currently, large embedded memories (say, >2Mbit) are equipped with redundancy in the form of some spare columns and rows, such that a limited number of faulty cells can be repaired, in order to improve the product yield. To determine which repairs to perform, the traditional memory BIST has to be extended with the capability to either make the repair decisions itself on-chip, or transport the fault bitmap of the memory off-chip. The latter is required also for failure analysis and yield ramp-up. As a result, the area cost of the BIST for embedded memories increases and with many embedded memories, it is becoming increasingly important to evaluate whether we can share test and repair resources between memories, i.e. to determine which test and repair resources are local to the individual memory versus global.

Test patterns for random logic are quite irregular, but definitely not random in nature. Nevertheless, most logic BIST stimulus generators are based on pseudo-random generators (e.g., Linear Feedback Shift Registers or Cellular Automata [25]), simply because they can be implemented area-efficiently. These get augmented by modifications of the module-under-test (such as Test Point Insertion [26]) or of the stimulus generator (such as Bit-Flipping [27]) to increase the obtained fault coverage to an acceptable level. Test responses are typically based on a Multiple-Input Shift Register in some form or another. BIST for random logic is not (yet) widely applied. Especially for circuits with many random-resistant faults, the silicon area costs of BIST for logic are relatively high. These costs can often only be justified if BIST is a customer requirement for in-field testing, when no ATE is available and BIST is the only feasible solution.

In recent years, the research field of *test resource partitioning* (TRP) has come into existence. It explores the spectrum between the two extremes for implementing source and sink: fully off-chip (ATE) and fully on-chip (BIST). With test data compression (TDC) techniques, test data is stored in compressed format onto the ATE, and decompressed only when loaded into the IC [28]. TDC is effective, as it exploits the fact that in an ATPG test set for logic, only 5% or less of the bits are actual 'care' bits, while the remaining bits are 'don't care'. As the test data decompression hardware is relatively cheap in silicon area cost (especially when compared to full BIST), this approach has quickly gained a lot of momentum. In other TRP approaches, the sequencer-per-pin on certain ATE systems is exploited to compress the data stored onto the ATE by means of runlength coding [29].

4 Trends in Automated Test Equipment

Today, requirements for a system chip may consist of a large set of digital, analog, and high-end RF functions. A combination of high-end RF, digital and analog performance in *one* semiconductor process is still far away, but via the System-in-Package (SiP) technology, this integration has become a reality much earlier than anticipated. This development will potentially lead to high-end test requirements in all of these domains at the same test stage.

To test the specification of such a system chip on any available ATE at a reasonable cost is a problem. It is clear that the sum of all these requirements cannot be met by a single hardware ATE solution. Even if a high-end-high-speed-RF-analog-digital-fast-data-managing-flexible system did exist, it would likely be too expensive. In addition, its delivery times would certainly not be in-line with the dynamics of the semiconductor industry. As such, the industry has to search for alternatives.

4.1 DfT-Based Testers

The price of a digital ATE depends on its channel count, vector memory depth, and accuracy. While large ICs seemed to

require large, and expensive ATEs, companies have started to realize that for many of their (structural) tests, at most 10% of the capabilities of their expensive ATEs is really used [30]. This led to a wave of interest into low-cost *'DfT testers'*. Such testers do work, provided that one is willing to rely on DfT-based structural testing only, and one is willing to add some on-chip DfT hardware to compensate for the reduced capabilities of the low-cost ATE.

As already indicated in the section on DfT, a main change in production test solutions is to find solutions via advanced DfT techniques. For example loop back techniques for high speed I/O testing [1] reduce the need for special high end capabilities on the ATE. Not only are these capabilities expensive, especially since they are in general active less than 1% of the total test cycle, they also have the risk of being outdated in a very short time. Using the proper DfT techniques reduces the needs for high-end performance, enabling real low-cost production ATE. A prerequisite for this is that a thorough evaluation and characterization of the device is still possible with a clear correlation to the DfT technique used.

4.2 Multi-Site Testing

Multi-site testing refers to testing multiple products on one ATE at the same time. When testing low-pin count devices in parallel on high-pin count testers, the infrastructure cost of a test cell is divided by the multi-site factor and the utilization of available testers is increased. The ability to perform high multi-site testing (up to 128× for wafer test and 32× for final test) is further enabled by the availability of advanced probe technologies and strip testing.

The application of large multi-site can in practice be restricted by several factors. Because the tests of the devices in one set run in parallel, the test of a new set cannot start until all devices in the current set have completed their test. In addition, multi-site may be restricted due to insufficient independent resources within the ATE, such as channels, analog instruments, data processing units, and/or power supplies. For example: To allow 32× multi-site testing of an IC which requires two independent power supplies, an ATE with 64 independent supplies is required. To be able to offer better multi-site support, ATE should be able to perform highly-parallel data processing and offer the possibility to use a scalable number of independent instruments.

4.3 Improved Utilization of the ATE Fleet

As the semiconductor market is characterized by steep up- and down-turns, matching test capacity to demand is an important issue. Subcontracting to dedicated test houses provides limited flexibility only, as in periods of scarcity, all market players are looking for extra capacity at the same time. Hence, proper management of the in-house ATE fleet remains important. Traditionally, ATE platforms have not been compatible; a test cannot be easily ported from one platform to another. That is certainly true for ATEs from different vendors, but even between the various platforms of a single ATE brand. This might lead to situations in which there is overcapacity on one platform, while we cannot secure sufficient capacity on another. Standardizing on one ATE platform prevents such situations and makes the test floor capacity management simpler and more effective. Currently, this can only be done by standardizing on one ATE vendor, which brings risks related to relying on one supplier only. An Open Architecture ATE platform [31] should allow multiple vendors to contribute components to a standardized ATE machine at competitive prices.

Standardized ATE platforms are something of the future; if only, because new ATEs are expensive, and we still have a working installed base, consisting of many different, incompatible platforms. In order to allow for a more balanced loading of all platforms in our current ATE fleet, we have engaged in porting the test program of many products to two or more platforms. IEEE Std. 1450 (Standard Test Interface Language or STIL) [32] is a crucial help to reduce the porting effort.

4.4 Product Lifecycle Reduction

Lifecycles for products are getting shorter. This leads to three different requirements. First, to be able to have a short first-time-right test development, the need to simulate and predict the device behavior during testing will increase. Second, since production ramp-up will be sharp, quick extension of production capacity is a must, giving pressure to lead times to be in the order of weeks rather than months. Third, during the quick ramp-up of the production, very fast and specific feedback is needed from the test operation to the wafer fab. A fast feedback will serve to find and correct failures that otherwise can have a negative impact on succeeding batches from the same product. Fast feedback will increase the yield learning, advancing both the process and the product quickly to maturity.

4.5 Enhanced Test Data Management

Collecting, storing, and retrieving of test data becomes increasingly important. For test quality, rejects are increasingly based not only on fixed pass/fail thresholds, but also on outlier behavior in a population of ICs. This requires that we collect and store the measurements of the population, before taking the actual pass/fail decisions. In order to remove the operational step of inking faulty dies in a wafer, we need

to store wafer maps, and make sure they remain linked to the corresponding physical wafers. Unique IDs per chip allow the identification of a single device all the way from wafer fab to customer, provided corresponding data is kept and accessible. And finally, test data management obviously plays a crucial role in the yield learning process between test floor and manufacturing fab.

5 Trends in Fault Diagnosis

The capability of test methods to provide feedback to the process engineers marks a shift in the position of testing. If one looks at the chain: from fab through test and product to customer, manufacturing test always has had the purpose to minimize the costs down-stream. Testing was only about determining if a chip is good or bad and we were willing to spend this money on testing because we knew that replacing bad ICs in PCBs or products is far more costly. Nowadays test should also provide information back to the process engineers to reduce the time-to-volume/yield of a process technology.

Process engineers have to find a balance between feature sizes and manufacturability. Large feature sizes ensure a high manufacturability however they result in larger and therefore more costly ICs, smaller feature sizes reduce the cost of making a single IC but with a higher probability that it is bad. To find this balance is an increasingly hard task, which is expressed in the growth of the number of design rules. An increasing fraction of the 'defects' is caused by design-process interaction or marginalities in the process instead of contaminations by particles. Most of these marginalities are detectable with test structures but certainly not all. Feedback from manufacturing test therefore becomes a necessity to economically make ICs in advanced process technologies.

As a result failure analysis and diagnosis are becoming key activities in today's IC manufacturing process. Successful silicon debug, yield improvement, and reliability enhancement heavily rely on the ability to quickly and precisely identify the root cause of the defects. Once the defect is detected and characterized, correction steps can be taken to on the one hand improve its detectability and on the other hand improve the manufacturing process to reduce the future chance that this defect occurs.

Fault diagnosis is the process of locating failures in ICs that have been identified as defective during test or customer application. There are two main activities in which fault diagnosis plays an important role: single die analysis and yield learning. The objective of these two activities is different but the end result is the same, improving the manufacturing process and achieving higher yields. The following two subsections try to give an overview of the challenges and trends in diagnosis seen from the perspective of these two application fields.

5.1 Single Die Analysis

Single die analysis is actually the traditional failure analysis process that consists of three steps: fault localization, deprocessing and defect characterization/visualization. Smaller feature sizes, higher integration densities, new materials and new defect mechanism introduced at each new technology node challenge the traditional physical fault localization methods. Some of these challenges as they are mentioned in the SIA roadmap [1] are the following: it becomes increasingly difficult to: probe a defective transistor for defect behavior characterization and localization without altering its functionality, to penetrate six or more metal layers with a laser beam (e.g., Optical Beam Induced Resistance CHange or OBIRCH) to activate the defect in order to locate it, and to use frontside and backside waveform acquisition techniques (e.g., Picasecond Imaging Circuit Analysis or PICA) because transitions emit less light. Therefore, precise and accurate software-based fault diagnosis methods are needed to supplement and/or substitute some of the physical localization methods.

The main trend in fault diagnosis is to develop methods whose callout should contain only one suspect location, and the spatial resolution to be limited to a single transistor or a metal line not longer than 10 μm [1]. These methods have to be able to analyze fail data from all test methods (scan for stuck-at, Iddq, and delay faults, and BIST.), cover new defect mechanisms introduced by the new technologies, and maybe even combine several test results for one single die if available to meet the precision and accuracy goals previously mentioned.

Software-based fault diagnosis techniques can be classified in two main groups [33]: cause-effect and effect-cause analysis. Cause-effect diagnosis techniques (see Figure 3) use fault simulation to determine the possible response of a circuit in the presence of faults when a test set is applied. This information is then matched with the response obtained from the tester in order to obtain the fault location. The other group of diagnosis techniques, effect-cause analysis, combine backtracking techniques, which justify all the values from the primary outputs as observed on the tester, with fault models to locate a fault within an equivalence class.

A fault model and a matching (comparison) algorithm are the primary elements involved in almost all software-based fault diagnosis techniques. On one side more complex fault models such as resistive bridging, resistive opens, path delay, and fault tuples [34] are being used for diagnosis purposes to try to cover every defect's complete behavior for

better localization with the cost of increasing the compute time. On the other side, the use of composite signatures and/or more complex matching algorithms [35, 36] tries to overcome the disadvantages related with a simple fault model to achieve the same goal. Adding layout information and the electrical characteristic of the neighborhood of the suspect fault location [37] have become a must if a good spatial resolution is desired.

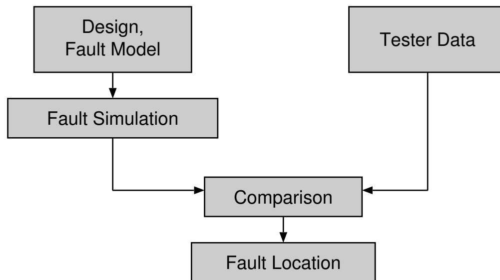

Figure 3: Cause-effect fault diagnosis flow.

5.2 Yield Improvement

The ability to get to high yields in a short time has become vital for every semiconductor company. Due to the increasing complexity and level of integration demanded by the new designs, the traditional methods based on using special test structures, embedded memories and in-line inspection data are no longer able to bring the yield fast enough to the desired high levels. Therefore, new techniques based on analyzing the failure that occur in the logic part of the design are needed. There are three main requirements imposed to any technique to be used for yield improvement:

- Production worthiness: the data processing algorithms have to be able to run in real-time and with little tester overhead.
- Volume: the data from several failing lots has to be analyzed because unique and rare defects are for yield improvement of lesser interest as our attention is focused on finding systematically recurring defects.
- Accuracy and precision: the technique has to pinpoint correctly to the processing step that is the main cause for the yield loss.

In recent years, software-based fault diagnosis techniques have started to fill this need with success: dedicated fault diagnosis algorithms have been developed to answer the first requirement [38]; statistical methods for post-processing of primary diagnosis callout and correlations with in-line inspection data have been implemented for highlighting systematic failure mechanisms [39, 40]. Fault diagnosis also starts to play an important role in yield improvement activities due to its ability to distinguish between the nuisance and real defects detected by the optical in-line inspection tools. The optical in-line inspection tools are loosing resolution due to the smaller feature size and increasingly detect also nuisance defects.

As mentioned earlier, the research in this field is just beginning, the trend is to build integrated systems which are able to analyze multiple data sources such as: diagnosis results of fail data from several types of test (e.g. scan based and parametric) performed at wafer level, data from yield monitors which may be present on the product wafers, defectivity information from in-line inspection systems, and design characteristics (e.g. critical area); in order to quickly and correctly pinpoint to the main cause of the yield loss.

6 Other Trends in Test

So far we have discussed trends that need to be solved largely in the test domain. In this section we discuss several other important trends, visible in the semiconductor industry, that will have their impact on the test community.

One of these trends is the move towards a disaggregated semiconductor industry. Where IP design, IP integration, design tools development, manufacturing, testing, and product marketing previously used to be done within the same company, these days we see companies specialized in one activity from this semiconductor supply-chain. An example is the manufacturing fab that is dedicated to offering technology processes to other companies. This allows semiconductor companies for instance to go fabless and outsource the manufacturing of their designs. To maintain competitiveness, whilst outsourcing some of your previously in-house activities, requires careful alignment between companies to allow efficient and effective exchange of information. Integrated Design Manufacturers (IDMs) have also started to use these additional services offered, for example by buying IP cores from third parties. The test techniques required for these cores have to be in line with the cores still developed internally, which is why standardization of test and diagnostic support is especially important for success in a disaggregated industry. IDMs also see the cost of their investments increase and try to align and standardize with other companies, for example in the domain of process technology development.

Today's chip designs, especially those with embedded microprocessors, start to include special functional modes to reduce the power consumption depending on the operational requirements. As these modes are by default not re-used during test, the power consumption during test is becoming an issue. A trade-off in both the design as well as the test domain is required to address this issue. The designer and

test engineer are left with two choices; either the power grid is designed to tolerate the (increased) power consumption during test, or the test patterns are modified to reduce the power consumption during test. The difficulty of this trade-off is that for the former solution, the design ends up being larger in silicon area and hence more expensive. For the latter solution, there is at this point little support in commercial tools to support power-constrained ATPG, which increases the test development time, and makes this solution unattractive to use.

A trend in today's system-on-chip designs is the use of multiple clock domains and, in many cases, multiple internally generated clocks. Internally there may not be one global clock controlling all communication. Instead, the communication between parts of the IC may be decoupled and handled asynchronously. Synchronizing and comparing the responses of these individual parts with test responses stored on the ATE may be extremely difficult due to non-deterministic behavior. In addition, the frequencies used inside the chip may be higher than the ATE's, resulting in the need to put part of the ATE functionality on the chip.

Improved security of consumer electronics devices, such that malicious persons cannot tampered with the operation of these devices, is becoming increasingly important. This applies both to mobile and domestic consumer appliances. The need for tamper-proof security appears to conflict with the requirements for test and diagnosis. The test community is used to adding 'back doors' to the design, for example DfT, that increase the internal observability to efficiently test for defects and provide feedback for the manufacturing process. Care has to be taken not to expose secret data, such as decryption keys, to the outside world. BIST offers a solution to this problem, as it keeps all information generated during test inside the device. Of course, this in turn makes failure analysis and fault diagnosis more difficult to impossible. Hybrid solutions, for example through providing authorized test and diagnosis operations, are still in a research and development phase.

Another future problem is the need to address the impact of soft errors, i.e. temporal glitches caused by alpha particles and neutrons. The first serious impacts of soft errors have already been reported. Soft errors cause intermittent faults, and hence cannot be tested for in the factory. The only known protection technology is by means of masking: functional redundancy or redundant logic. Traditionally, testing and redundant logic do not go together very well, simply because detection and masking are contradictive. The increasing impact of soft errors will lead to an increasing role for redundancy, not only in memories, but also in digital logic [41]. Fine-grained redundancy might relieve hard-to-meet test quality constraints, although masking of hard defects will deteriorate the protection offered against soft errors [42]. Test technology and fault tolerance will have to learn to live together in a new equilibrium.

7 Conclusion

The continued development of new process technologies with smaller feature sizes and ICs with increased complexity is the driver behind many of the test challenges. This paper provides an overview of current trends in test.

We see a continuation of the research to find better fault models, and better test methods to detect more and weaker defects in our ICs. Statistical analysis, e.g. using the measured behavior of neighboring dies to help decide on what is still normal device behavior and what is defective behavior, starts to play an increasingly important role. Design-for-Testability continues to play its role as the enabler of structured test methods. In addition, it will increasingly be used to help drive ATE requirements down, by implementing part of the ATE's functionality on-chip. The need to efficiently utilize available ATE, through the use of standard and open architectures, or multi-site testing, will reduce the test cost per device. Improved fault diagnosis is required to analyze both individual parts, and help determine process and design marginalities. In production it should provide accurate and fast diagnosis information which can drive yield learning when fed back from test to the fab. This is essential to enable rapid yield ramping of new technology nodes. In the future test will continue to be the gate keeper to ensure that quality parts are shipped to the customer, and given the undeniable trends in the semiconductor industry, this role will be more and more important.

Acknowledgements

The authors thank Frank Bouwman and Bill Price for their valuable comments on the draft version of this paper.

References

[1] Semiconductor Industry Association. *International Technology Roadmap for Semiconductors (ITRS)*. SIA, December 2003. (Available via http://public.itrs.net/).

[2] P.K. Nag, A. Gattiker, S. Wei, R.D. Blanton, and W. Maly. Modeling the Economics of Testing: A DFT Perspective. *IEEE Design & Test of Computers*, 10(1):29–41, January 2002.

[3] E.J. McCluskey. Quality and single-stuck faults. In *Proceedings IEEE International Test Conference (ITC)*, page 597, October 1993.

[4] W. Maly, A. Gattiker, T. Zanon, T. Vogels, R.D. Blanton, and T. Storey. Deformations of IC Structure in Test and Yield Learning. In *Proceedings IEEE International Test Conference (ITC)*, pages 856–865, October 2003.

[5] S. Ma, P. Franco, and E.J. McCluskey. An Experimental Chip to Evaluate Test Techniques Experiments Results. In *Proceedings IEEE International Test Conference (ITC)*, pages 663–672, October 1995.

[6] M. Grimaila, S. Lee, J. Dworak, K. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, J. Park, L. Wang, and M. Mercer. REDO - Random Excitation and Deterministic Observation First Commercial Experiment. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 268–275, April 1999.

[7] R. Aitken. A comparison of Defect Models for Fault Localisation with Iddq measurements, Very-low Voltage Testing for Weak CMOS Logic ICs. In *Proceedings IEEE International Test Conference (ITC)*, pages 778–787, September 1992.

[8] B. Kruseman, A. Majhi, G. Gronthoud, and S. Eichenberger. On Hazard-Free Patterns For Fine-Delay Fault Testing. In *Proceedings IEEE International Test Conference (ITC)*, October 2004.

[9] H.T. Vierhaus, W. Meyer, and U. Gloser. CMOS Bridges and Resistive Transistor Faults: IDDQ versus Delay Effects. In *Proceedings IEEE International Test Conference (ITC)*, pages 83–91, October 1993.

[10] H. Hao and E.J. McCluskey. Very-low Voltage Testing for Weak CMOS Logic ICs. In *Proceedings IEEE International Test Conference (ITC)*, pages 275–284, October 1993.

[11] W.R Daasch, K. Cota, J. McNames, and R. Madge. Neighbor selection for variance reduction in Iddq and other parametric data. In *Proceedings IEEE International Test Conference (ITC)*, pages 92–99, October 2001.

[12] R. Madge, B.H. Goh, V. Rajagopalan, C. Macchietto, R. Daasch, C. Schuermyer, C. Taylor, and D. Turner. Screening MinVDD Outliers Using Feed-Forward Voltage Testing. In *Proceedings IEEE International Test Conference (ITC)*, pages 673–682, Baltimore, MD, October 2002.

[13] A. Kobayashi, S. Matsue, and H. Shiba. Flip-Flop Circuit with Fault Location Test Capability. In *Proceedings IECEO Conference*, page 962, 1968. (In Japanese).

[14] S. Funatsu, M. Kawai, and A. Yamada. Scan Design at NEC. *IEEE Design & Test of Computers*, 6(3):50–57, June 1989.

[15] S.K. Goel and E.J. Marinissen. SOC Test Architecture Design for Efficient Utilization of Test Bandwidth. *ACM Transactions on Design Automation of Electronic Systems*, 8(4):399–429, October 2003.

[16] S.K. Goel, K. Chiu, E.J. Marinissen, T. Nguyen, and S. Oostdijk. Test Infrastructure Design for the Nexperia^{TM} Home Platform PNX8550 System Chip. In *Proceedings Design, Automation, and Test in Europe (DATE) Designers Forum*, pages 108–113, Paris, France, February 2004.

[17] Y. Zorian, E.J. Marinissen, and S. Dey. Testing Embedded-Core Based System Chips. In *Proceedings IEEE International Test Conference (ITC)*, pages 130–143, Washington, DC, October 1998.

[18] E.J. Marinissen et al. On IEEE P1500's Standard for Embedded Core Test. *Journal of Electronic Testing: Theory and Applications*, 18(4/5):365–383, August 2002.

[19] F. DaSilva, Y. Zorian, L. Whetsel, K. Arabi, and R. Kapur. Overview of the IEEE P1500 Standard. In *Proceedings IEEE International Test Conference (ITC)*, pages 988–997, Charlotte, NC, September 2003.

[20] IEEE Computer Society. *IEEE Standard Test Access Port and Boundary-Scan Architecture - IEEE Std. 1149.1-2001*. IEEE, New York, July 2001.

[21] Y. Zorian and E.J. Marinissen. System Chip Test: How Will It Impact Your Design? In *Proceedings ACM/IEEE Design Automation Conference (DAC)*, pages 136–141, Los Angeles, June 2000.

[22] V. Iyengar, K. Chakrabarty, and E.J. Marinissen. Recent Advances in Test Planning for Modular Testing of Core-Based SOCs. In *Proceedings IEEE Asian Test Symposium (ATS)*, pages 320–325, Tamuning, Guam, USA, November 2002.

[23] H. Vranken et al. Enhanced Reduced Pin-Count Test for Full-Scan Design. In *Proceedings IEEE International Test Conference (ITC)*, pages 738–747, Baltimore, MD, October 2001.

[24] R. Dekker, F. Beenker, and L. Thijssen. Realistic Built-In Self-Test for Static RAMs. *IEEE Design & Test of Computers*, 6(1):26–34, January 1989.

[25] J. Rajski, G. Mrugalski, and J. Tyszer. Comparative Study of CA-Based PRPGs and LFSRs with Phase Shifters. In *Proceedings IEEE VLSI Test Symposium (VTS)*, pages 236–245, Dana Point, CA, USA, April 1999.

[26] N. Tamarapalli and J. Rajski. Constructive Multi-Phase Test Point Insertion for Scan-Based BIST. In *Proceedings IEEE International Test Conference (ITC)*, pages 649–658, Washington, DC, USA, October 1996.

[27] G. Kiefer, H. Vranken, E.J. Marinissen, and H.-J. Wunderlich. Application of Deterministic Logic BIST on Industrial Circuits. In *Proceedings IEEE International Test Conference (ITC)*, pages 105–114, Atlantic City, NJ, USA, October 2000.

[28] F. Poehl et al. Industrial Experience with Adoption of EDT for Low-Cost Test without Concessions. In *Proceedings IEEE International Test Conference (ITC)*, pages 1211–1220, Charlotte, NC, September 2003.

[29] H. Vranken, F. Hapke, S. Rogge, D. Chindamo, and E. Volkerink. ATPG Padding and ATE Vector Repeat Per Port for Reducing Test Data Volume. In *Proceedings IEEE International Test Conference (ITC)*, pages 1069–1078, Charlotte, NC, USA, September 2003.

[30] S. Comen. DfT-Focused Chip Testers: What Can They Really Do? In *Proceedings IEEE International Test Conference (ITC)*, page 1120, Atlantic City, NJ, October 2000. (Panel position statement).

[31] S.M. Perez and Y. Furukawa. Open Architecture Test System: The New Frontier. In *IEEE Int. Electronics Manufacturing Technology Symposium*, pages 211–214, July 2003.

[32] IEEE Computer Society. *IEEE Standard Test Interface Language (STIL) for Digital Test Vector Data - Language Manual - IEEE Std. 1450.0-1999*. IEEE, New York, September 1999.

[33] M. Abramovici, M.A. Breuer, and A.D. Friedman. *Digital Systems Testing and Testable Design*. IEEE Press, 1990.

[34] R.D. Blanton, J.T. Chen, D. Desineni, K.N. Dwarakanath, W. Maly, and T.J. Vogels. Fault Tuples in Diagnosis of Deep-Submicron Circuits. In *Proceedings International Symposium for Testing and Failure Analysis (ISTFA)*, pages 233–241, October 2002.

[35] L.M. Huisman. Diagnosing arbitrary defects in logic designs using single location at a time (SLAT). *IEEE Transaction on Computer-Aided Design of Integrated Circuits and Systems*, 23(1):91–101, January 2004.

[36] S. Venkataraman and S.B. Drummonds. POIROT: A logic fault diagnosis tool and its applications. In *Proceedings IEEE International Test Conference (ITC)*, pages 253–262, October 2000.

[37] Y. Sato, I. Yamazaki, H. Yamakada, T. Ikeda, and M. Takakura. A persistent diagnostic technique for unstable defects. In *Proceedings IEEE International Test Conference (ITC)*, pages 242–249, October 2002.

[38] C. Hora, R. Segers, S. Eichenberger, and M. Lousberg. An Effective Diagnosis Method to Support Yield Improvement. In *Proceedings IEEE International Test Conference (ITC)*, pages 260–269, October 2002.

[39] A. Kinra, H. Balachandran, R. Thomas, and J. Carulli. Logic Mapping on a Microprocessor. In *Proceedings IEEE International Test Conference (ITC)*, pages 701–710, October 2000.

[40] B. Benware. Logic Mapping on ASIC Products. In *Proceedings International Symposium for Testing and Failure Analysis (ISTFA)*, pages 579–586, October 2002.

[41] P. Shavikumar et al. Modelling the Effect of Technology Trends on Soft Error Rate of Combinational Logic. In *Proceedings International Conference on Dependable Systems and Networks*, pages 389–398, San Francisco, CA, June 2002.

[42] A. Nieuwland and R. Kleihorst. The Positive Effect on IC Yield of Embedded Fault Tolerance for SEUs. In *Proceedings IEEE International On-Line Testing Symposium (IOLTS)*, pages 75–79, Kos, Greece, July 2003.