Meeting your DPPM and other test requirements demands more than just high test coverage—you’ll need techniques such as layout-aware diagnostics to improve yield and quality.

By Colin Renfrew, Freescale Semiconductor, and Bruce Swanson, Mentor Graphics

Suppose you were given the assignment to find a basketball in the Los Angeles area—just one basketball in no particular location, but in a location where it wasn’t supposed to be. How would you begin?

Being assigned the task of finding a defect in a modern IC device can be similar to having to find that errant basketball. Are you ready for this task? Your company’s reputation for quality may depend on it.

When it comes to products for the medical or automotive markets, the quality requirements are extremely high, targeting very low to zero DPPM (defective parts per million). In many cases, the customer sets these requirements, dictating, for instance, both the specific metrics for test-coverage goals and the fault models that you must use. Test requirements also can be associated with specific industry standards, such as those from the Automotive Electronics Council, to ensure that comparative metrics are used.

Many companies have entire organizations and teams dedicated to product quality and yield enhancement, and DFT (design for test) is a key element from product planning through production. Although high test coverage is commonly used as a defining metric, it is only one piece of providing high-quality IC products. In addition to ensuring adequate test coverage, you must generate the correct test patterns, ensure they run on the specified ATE (automatic test equipment), and also ensure they work on first silicon and in production test. And when material starts to fail tests, you must be able to identify the root cause so the problem can be avoided next time. Quality doesn’t happen by accident; it must be designed into the product and process.

Planning for quality
Some wise person once said that most people don’t plan to fail; they just fail to plan. A comprehensive test plan is imperative for achieving high-quality ICs. The first step in developing a plan is to determine what the quality requirements are for the design and what elements in the design need to be tested to ensure those requirements are met. For instance, is the design all digital logic, or does it also have some analog pieces? Does the design include embedded memories or PLLs (phase-locked loops) for on-chip clock generation? Are there high-speed I/O pins or any other special interface requirements? What are the requirements for the target market?

Here are some of the main items to consider when putting together a test plan:

• For large designs or when using multiple ATPG (automatic test-pattern generation) fault models, it is common to include some on-chip test-compression technique. Does this design need test compression? If so, how much?
• Which BIST (built-in self-test) algorithms will you use to test any on-chip memory?

(continued)
• Which type of ATE will you use? What are its capabilities and limitations? Will testing occur at the wafer level, at the packaged-part level, or both?
• Must tests run at system speed? How many clock domains are there, and how should testing be done between those domains?
• Will diagnostics and FA (failure analysis) be done when ICs fail during production test? How will that data be used to improve yield and quality?
• What standards need to be complied with, such as IEEE 1149.1 or 1149.6 for boundary scan?

Chip designs for use in critical medical applications, such as microcontrollers for blood-glucose monitors or medical-imaging equipment, have no room for error. Across Freescale Semiconductor’s product portfolio, the company’s DFT engineers use multiple different fault models in ATPG—including stuck-at, transition-delay, path-delay, bridging, and small-delay-defect models—to target the many potential fault sites in the device and exercise them in different ways.

This approach maximizes their ability to detect problems on silicon, including simple stuck-at defects, resistive nets that vary across PVT (process, voltage, and temperature) points, and crosstalk or power-induced effects. The DFT team chooses memory BIST solutions to provide as many different test methods and algorithms as possible.

When it comes to achieving high quality, a single high-test-coverage number on one fault model is not enough. Stuck-at and transition-delay high-test-coverage values are essential in today’s ICs, commonly at 99% and 90%, respectively.

For example, some of Freescale’s automotive products require in-field test to ensure that the product still functions correctly several years after it has been manufactured. It is not enough for the engineers to test a device after manufacturing and put it into a car. Automobiles have hundreds of ICs, and many of them are safety-critical, so it is essential that they can test themselves and flag any errors to the customer. Logic BIST can accomplish this task. Adding logic BIST capabilities to a device will enable it to perform these self-tests.

A basic cost-benefit analysis can help you choose which DFT methodologies you should adopt for a design and which tools you should use to execute these test methods. Thoroughly testing every possible physical net and logic gate on a chip would be ideal, but you also need to ship a product on time and within cost constraints, so you may need to make tradeoffs.

In addition to having the right tools and knowing how to apply them to the test methodology, you will need to understand the interaction between tools at
different stages of the design and test flow. For example, when diagnosing yield failures, can your diagnostic tool directly read the test patterns from ATPG, or will you need translation scripts to get from one tool to the other? During the planning phase, you’ll need to consider the entire tool flow when evaluating a certain DFT technique, especially if you plan to use a new technique, architecture, or tool. At this point, you will also need to determine who in the organization will own specific parts of the DFT flow.

**Designing for test, designing for quality**

Once you develop a test plan, how do you execute it? Figure 1 illustrates the main pieces of the test and quality-improvement flow. The first step after developing the test plan involves generating and inserting logic BIST, memory BIST, and on-chip compression engines, and then stitching the scan chains. Control logic for PLLs and chip-test-mode configuration are also important to consider at this stage. Any required test logic should be generated and integrated into the rest of the design carefully. This logic can be either generated as separate IP (intellectual property), such as for memory BIST logic, or generated automatically by electronic-design-automation tools.

Before the pattern-generation stage, you should learn the capabilities and limitations of the ATE system so you can create the optimal test-pattern set. You must also validate the test patterns before applying them to silicon. Just because the ATPG tool was able to generate the pattern doesn’t mean it will actually work—the models used by the ATPG tool could have errors or simply be the wrong version of the design.

Freescale engineers validate all test patterns in simulation with back-annotated SDF (standard-delay-format) timing to ensure correctness. This validation is especially important for at-speed patterns, where timing constraints, false paths, or timing exceptions can be missed, causing false fails on silicon. For the majority of products, the engineers validate at least some patterns in a virtual test environment on a simulated tester before first silicon arrives. Once you per-
form your validation simulation, you will need to convert the test patterns to run on the specific tester platform you are using. You will also need to characterize and validate both the test patterns and the test program before starting production test. The sheer volume of test patterns together with the time-to-market push means that there is little time to spare on debug once silicon arrives—the test patterns should, as often as possible, work on day one.

**Improving yield and quality**

Depending on the product, hundreds to tens of thousands of Freescale ICs may be run through production test every month, generating a wealth of data that is constantly being analyzed and reviewed to ensure maximum yield.

Often, ATE test patterns are executed over a range of voltage and frequency. A typical shmoo plot (Figure 2) shows the passing and failing ranges for a tested device. When failing devices are caught on the tester, the ATE system creates a failure-log file to capture the details of what failed. With a process called diagnosis-driven yield analysis (Ref. 1), engineers can use the data from the failure logs to make improvements at all phases of design and test and thus improve overall quality and yield.

The main goal of using diagnosis for failure analysis is to localize the defect or defects causing failures on a specific die, such as the voided (or open) via shown in Figure 3. A diagnosis tool can take the design netlist, the test patterns, and the failure file from the tester to figure out the most probable defect suspects, both type and location, within the die. Even better results are obtained if the design layout information also is included in the process, using a technique called layout-aware diagnosis.

Layout-aware diagnosis provides comprehensive results that include more than 50 attributes for each failure. For applications such as FA or test bring-up, diagnosis is typically done on a small number of devices. But to most effectively leverage diagnosis results for yield analysis, it is better to run diagnosis as part of the manufacturing test flow. This process provides a wealth of diagnosis data that is analyzed for yield impacts.

One of the main challenges of having so much diagnosis data is finding a way to separate valuable information from noise so you can distinguish between random defects and ones that indicate a systematic defect issue that could be fixed to improve yield. A yield-analysis tool can perform statistical analysis on the diagnosis data to find the failure signatures that have the largest impact on yield. Then, the tool further determines which failing die are the best candidates to send to FA for physical analysis.

**Figure 4** shows a suspected defect area highlighted in the layout viewer. After device selection, FA is used to confirm the presence of the suspected cause. The results from the yield-analysis and FA processes must be fed back to the design teams so they can take corrective actions or measures such as process changes, design changes, or library changes to fix the problems and drive up yield levels. At Freescale, this feedback is given to more than just the SOC team. It is also provided further upstream to the IP designers, library teams, process and technology teams, and design-for-manufacturing teams to ensure they can make appropriate corrections for the next product or product revision. This approach ultimately leads to a higher overall quality by either improving the inherent yield or, through DFT, by screening out any defective parts before they reach the customer.

Quality does not come easily and requires careful planning across the entire design flow, from concept to feedback from silicon analysis. Improving yield improves your gross margin, and reducing the number of defective parts found by the customer improves profitability and keeps the customer happy. Both of these are directly impacted by and reliant upon the DFT techniques and methodologies that enable high quality to be achieved. So now that you’ve done the hard work to ensure high product quality in your ICs, relax a little and go shoot some hoops—if you can find a ball!

**Improving yield improves your gross margin.**

REFERENCE


Colin Renfrew is a DFT manager in the Networking and Multimedia Group at Freescale Semiconductor in Austin, TX. He received an MS in system-level integration from the Institute for System Level Integration at the University of Edinburgh, Scotland, and a BEng in computer and electronic systems from Strathclyde University, Scotland. Renfrew has been with Freescale (previously Motorola SPS) since 2003, working in the areas of DFT, design, and verification both in Europe and the US.

Bruce Swanson is a technical marketing engineer in Silicon Test Solutions at Mentor Graphics. He received an MS in applied information management from the University of Oregon and a BS in computer engineering from North Dakota State University. Bruce has more than 20 years of experience in EDA and computer hardware design.