Manufacturing test and failure analysis remain at the forefront of determining why and how chips fail. Defective chips (i.e., those that fail the production test) offer a goldmine of information. If properly mined, they can provide valuable insight into the defects and failure mechanisms that are limiting yield. The industry standard for testing digital integrated circuits (ICs) after manufacturing is to place them into a piece of automated test equipment (ATE). Various logic test patterns are then run on them to find defects in the circuit. This process is scan test. Most of the test patterns are created by an automated-test-pattern-generation (ATPG) tool. This tool is based on fault models, which are designed to model the various failure behaviors that are found in the ICs when defects are present.

To be able to run scan tests through the IC, some amount of test logic is added to the design that facilitates ATPG. The biggest change is that all (or at least most) of the design's flip-flops are changed to scan cells. These scan cells are linked together into scan chains, which operate like shift registers when the circuit is put into test mode.

The scan chains are used by the tester to deliver the test-pattern data from its memory into the device. After the test pattern is loaded, the design is placed back into functional mode. The test response is captured in one or more cycles. The design is then put in test mode again to shift out the captured test response. At the same time, the next test pattern is loaded into the design. The tester then compares the captured test response with the expected response data that also is stored in its memory. Any test mis-comparisons are most likely defects. They are logged for further evaluation (see Figure 1).

A third fault model, which has been used for several years, is called path delay. This model, which also is dynamic, performs at-speed tests of identified timing-critical paths. The stuck-at and transition-fault models add faults for all of the design's nodes by testing for defects across the whole device. The path-delay model simply adds faults for the exact paths specified by the ATPG user. It also tests those paths.

Usually, the designer runs static timing analysis as part of the design and verification process. From this analysis, a list of the most critical paths is obtained. These paths are specified to the ATPG tool for creating the path-delay test patterns. The theory is that if the most critical timing paths can pass the tests, the other paths with longer slack times should have no timing problems. The path-delay test is used as more of a process check rather than just looking for manufacturing defects on those specified paths.

**TESTING FOR ADVANCED DIAGNOSIS**

As IC design and manufacturing technologies move to 90 nm and below, defects on a die are becoming more difficult to diagnose. At these nodes, achieving the highest levels of test quality and accuracy is critical to locating and analyzing defects. The basic fault models still capture the majority of defects. To increase accuracy and quality at these resolutions, however, two advanced fault models can be applied: bridging and small delay.

The feature sizes used in advanced IC designs are so small that more bridging types of defects are occurring. The manufacturing process is very complex and requires hundreds of steps. If they aren’t done perfectly, many of these steps can inadvertently produce unintended defects. Advanced scan test and diagnosis enhances yield learning. Discovering why chips fail is a shortcut to increasing yield.

**BASIC TEST FAULT MODELS**

The most widely used and basic fault model is called stuck-at. This model checks each node location in the design for either stuck-at-1 or stuck-at-0 logic behavior. For example, if a NAND gate in the design had an input pin shorted to ground (logic value 0) by a defect, the stuck-at-0 test for that node would catch it. The stuck-at model also can detect other defect types like bridges between two nets or nodes. The stuck-at is classified as a static type of model because it’s a slow-speed test as opposed to an at-speed test.

The main dynamic fault model being used today is called transition fault. Similar to the stuck-at model, it checks for two faults at every gate terminal in the design. These are classified as slow-to-rise and slow-to-fall faults. The transition-fault model uses a test pattern that creates a transition stimulus to change the logic value from either 0-to-1 or 1-to-0. The time allowed for the transition is fixed. If the transition doesn’t happen or occurs after the allotted time, a timing type of defect is presumed.

As IC design and manufacturing technologies move to 90 nm and below, defects on a die are becoming more difficult to diagnose. At these nodes, achieving the highest levels of test quality and accuracy is critical to locating and analyzing defects. The basic fault models still capture the majority of defects. To increase accuracy and quality at these resolutions, however, two advanced fault models can be applied: bridging and small delay.

The feature sizes used in advanced IC designs are so small that more bridging types of defects are occurring. The manufacturing process is very complex and requires hundreds of steps. If they aren’t done perfectly, many of these steps can inadvertently produce unintended defects.
bridges between nets in the design. These problems can be classified as systematic defects that affect many of the ICs on a wafer.

The other common way that bridges occur is when some type of impurity unintentionally connects two nets. These are classified as random defects. The physical spacing between nets in the layout is now very small compared to older technologies with wider spacing. As a result, these random defects are more likely to create bridging problems.

Stuck-at test patterns can catch the majority of bridging defects. The stuck-at test targets only one node at a time, however. As a result, it cannot catch all bridging defects. The leading ATPG tools now provide a four-way bridging fault model to address this growing problem. The standard bridging model between two nets is for one net to dominate the other. To add more coverage for each pair of nets, four faults are added:

- sig_A is dominant with a value of 0 (sig_A=0; sig_B=1/0)
- sig_A is dominant with a value of 1 (sig_A=1; sig_B=0/1)
- sig_B is dominant with a value of 0 (sig_B=0; sig_A=1/0)
- sig_B is dominant with a value of 1 (sig_B=1; sig_A=0/1)

An important capability is selecting nets to use with the bridging fault model. Because so many combinations of net pairs are possible within a design, it’s unrealistic to create test patterns for every potential pair. New techniques evaluate the design’s physical layout to look for areas or net properties that make them more likely to bridge. A set of extraction rules is used with the layout database to find the net pairs that match the targeted characteristics. Examples include properties like corner-to-corner spacing, long parallel runs at tight spacing, and end-to-end spacing.

Once the set of net pairs to target is ready, it’s loaded into the ATPG tool. Test patterns are then created. If stuck-at test patterns already exist for the design, it’s possible to fault-grade those patterns with the new model before creating the targeted bridging test patterns.

A method to statistically test for bridging defects across the whole design is the multiple detection pattern technique. The principle here is to test if each fault location is detected multiple times with randomly different control and observation points. This process raises the probability that a shorted/bridged net will be at the opposite value and detected. This method can be used for either the stuck-at or transition-fault models.

**SMALL-DELAY DEFECTS**

In addition to using smaller feature sizes, today’s designs are running at much higher speeds than in the past. With these high clock frequencies and tighter timing windows, there is a greater opportunity for the occurrence of small-delay defects. Like bridging problems, these defects can be either systematic or random.

One would assume that transition test patterns would find all of the timing defects in the design when they occur. Although this is true most of the time, some of the smallest delay defects can escape being caught with the basic transition test pattern. For example, suppose a gate output is driving a net that connects to five different loads with five different total delay values on those paths. For the transition-fault model, the ATPG tool gives credit for detection on that output pin if a successful test is created. That test can use any of the five paths for observation. One of the shorter paths (longest slack time), for example, could be used for the transition test pattern because the ATPG program uses random decisions to excite and propagate the transition.

The newer small-delay fault model was developed to address this transition-fault-model limitation. For the small-delay fault model, the actual timing information from the completed layout is used. The ATPG tool reads in that timing data and uses it to target fault detection on the longest possible paths. This type of pattern has a higher probability of catching small-delay defects if they’re present.

**GETTING THE MOST FROM SCAN TEST**

For input, scan logic diagnosis takes a flattened design description, test-pattern set, and fail log generated by a tester when applying the test-pattern set to the die. The diagnostic report can be used to determine a fault’s location, type, observed faulty behavior, and a confidence value (rank) of how accurately the reported location and type match the actual faulty behavior.

Accurate diagnosis involves identifying the defect’s logical and physical location as well as the defect mechanism whenever possible (see Figure 2). In a high-volume manufacturing flow, diagnostics must be performed directly on production data. Tens of thousands of devices per day may fail manufacturing test. To achieve statistically relevant results, effective yield learning must consider all failures.
With the advanced designs and manufacturing processes being used today, large volumes of data are required to identify yield trends and focus yield-improvement efforts. If diagnosis is going to play a key role in yield learning, the embedded test-compression solution being adopted must support diagnosis directly from the compressed responses and patterns. Volume diagnosis has to work in concert with test-pattern compression techniques. Running additional or bypass tests during manufacturing for the sole purpose of logging diagnostics data is both impractical and cost prohibitive.

Mainstream manufacturing test for large digital devices relies on scan cells and scan chains to access the design’s internal nodes. If the scan chains themselves are not working, the test patterns for testing the digital logic cannot be applied. The flip-flops, scan logic, clocks, and enable lines that comprise the scan-chain structures can account for a significant portion of the logic. As such, they also can account for a large portion of the failures coming out of manufacturing test. For this reason, the accurate diagnosis of scan-chain failures offers an important tool to help in understanding yield loss.

Now, assume that the device passes the scan-chain tests, but fails one of the many scan-test patterns meant to test the functional logic within the design. Here, diagnosis provides valuable insight into failure and yield loss mechanisms. Based on the test passing and failing information, logic diagnostics identifies potential defect locations as well as suspected defect types like those shown in Figure 3.

Once the defect suspects and types are determined, the next step is to link the failures in the logical domain with the features and locations in the physical domain. In both the first silicon debug and yield ramping stages, it’s not unusual to find physical features that are more prone to defects than others. A robust test set may contain a mix of test patterns including stuck-at, transition, path delay, and bridging. Because of observed defect rates and defects-per-million requirements, it’s often necessary to improve the effectiveness of a given pattern set. This task is accomplished by targeting additional tests at either specific areas or features within the design or at certain types of defects. Based on diagnostic results and links with the physical-layout information, specific areas (nets or layers) or features can be targeted for additional testing. With the knowledge of where and how things are failing, additional ATPG patterns that target suspect features can be created to improve test quality.

The full potential of layout information is only achieved when the scan-logic diagnosis can access the layout in its core algorithm. Layout information enables the scan-logic diagnosis tool to determine new fault candidates and drop physically impossible candidates from consideration. In doing so, it improves the resolution of the diagnosis result. That result doesn’t just report, for example, the nets of a bridge failure candidate and the X/Y coordinates of the nets. It also offers a list of all potential defect locations including their X/Y/layer coordinates and size (length and width). Such a list further reduces the failure identification time while improving ease of use. Most importantly, the reported failure candidates are certain to be both physically and logically possible. As a result, this process eliminates candidates that could waste investigators’ time.

**Applying DFM Recommendations**

Another change happening in the IC design and manufacturing flow is the use of design-for-manufacturing (DFM) rules or recommendations. Such recommendations go beyond the traditional design rule checks (DRCs). Essentially, these guidelines indicate what can be done in the physical design to help improve yield during manufacturing. From a test perspective, they also indicate the more likely defect sites within a design.

DFM-oriented test is a methodology that uses DFM recommendations and guidelines to extract from layout the most likely defect locations and types. It then specifically targets tests for those types of defects as well as specific locations. Using this methodology, specific tests for bridging or other potential physical defects can be targeted at the features within the design that are most likely to produce faults. Yield learning ensures that tests are targeted at the most likely defect types and locations. In doing so, they add to the quality of DFM-oriented test.

**Scan Diagnosis for Fabless Companies**

For a fabless company, getting access to foundry manufacturing process data can be difficult. This is especially true when production test and failure analysis also are outsourced. Similarly, the owner of the design is usually very reluctant to give away the layout information—let alone the design. One possible solution that fabless companies are investigating is a flow in which the design owner executes scan test and diagnosis with or without layout information. Sensitive design information in the diagnosis report is encrypted. For example, the net or instance names for which a fault was diagnosed can be rendered in a cryptic identifier. For pointing failure analysis at the right nets and polygons, a layout overlay file can easily be generated by the diagnosis tool from layout data.
From the point of view of most design owners, this file and a non-annotated GDS file of the design contain no sensitive information. They can therefore be given to external failure-analysis personnel to be applied to the yield learning process.

As the semiconductor industry moves to technology nodes smaller than 90 nm, scan test and diagnosis are an important part of the design-for-manufacturing flow. At 45 nm and below, scan test, compression, and diagnosis are absolutely critical for a fast yield ramp and meeting cost and time-to-volume targets. As IC technology evolves to meet new market demands, identifying defects and their causes, performing verification and debug at each stage of design and manufacture, and closing the loop between physical design and manufacturing are no longer optional.

**FURTHER INFORMATION:**

“Scan Diagnostics in the Nanometer Design Era,”

“Yield Improvement with Compressed Pattern Diagnosis,”
SDD 06, Imtiaz Ahmed, Jeffrey Bartsch, and Sanjay Sharma, Skyworks; Yu Huang, Wu-Tung Cheng, and Wu Yang, Mentor Graphics.


“Improving Transition Fault Test Pattern Quality through At-Speed,” ITC2006, Nandu Tendolkar, et. al., Freescale; Wu-Tung Cheng, et. al., Mentor Graphics; Akshay Gupta, University of Texas, Austin.


Bruce Swanson is a technical marketing engineer in the Design-for-Test division of Mentor Graphics Corp.