Small-delay-defect testing

Roberto Mattiuzzo - June 01, 2009

WEB-EXCLUSIVE
SIDEBAR:
"Test effectiveness"

See other articles from our June 2009 issue.

Semiconductor companies have come to rely on delay testing to attain high defect coverage of manufactured digital ICs. Delay testing uses TD (transition delay) patterns created by ATPG (automatic test-pattern generation) tools to target subtle manufacturing defects in fabricated designs. Although TD ATPG improves defect coverage beyond the levels that stuck-at patterns alone can achieve, the methodology is limited in its ability to reach the test quality levels required for nanometer designs. As a result, STMicroelectronics is deploying a new delay test methodology called SDD (small delay defect) ATPG as a means to achieve even higher defect coverage than standard TD ATPG.

Why SDDs?

“Delay defect” refers to any type of physical defect, or an interaction of defects, that adds enough signal-propagation delay in a device to produce an invalid response when the device operates at the targeted frequency. Experimental data going back two decades (Ref. 1 and 2) have shown that the distribution of delay-related failures is skewed toward the smaller delays. That is, the majority of devices that fail due to delay defects fail because of “small delay defects” that contribute to delays much smaller than the clock cycle times associated with the process technology node. Targeting these SDDs during test improves defect coverage and lowers the test escape rate, measured as DPPM (defective parts per million).

On-chip process variations are more pronounced in today’s manufacturing processes because of the increased presence of systematic defects—stemming from complex interactions between layout, mask manufacturing, and wafer processing—compared with previous process technologies (Ref. 3). These process variations tend to further skew the delay-failure distribution toward smaller delays, adding enough incremental signal delay to adversely impact circuit timing in a higher percentage of devices. In essence, for a given die size, the product yield of a 45-nm design can decrease...
sufficiently over that of a 90-nm design that manufacturers must boost the coverage of SDDs just to maintain about the same DPPM levels observed for the 90-nm process.

**Limitation of TD testing**

So, why doesn’t standard TD testing cover SDDs? In fact, it does cover some of them, but not enough to achieve the quality levels required at STMicroelectronics, which is pursuing a strategic quality objective of zero DPPM.

The traditional goal of ATPG tools has been to minimize run time and pattern count, not cover SDDs. TD ATPG targets delay defects by generating one pattern to launch a transition through a delay fault site—which may activate either a slow-to-rise or a slow-to-fall defect—and by generating a second pattern to capture the response. During testing, if the signal doesn’t propagate to an end point (a primary output or scan flop) in the at-speed cycle time, then incorrect data is captured. In this scenario, the pattern sequence detects a delay defect through the activated path.

![Figure 1](image1.png)

**Figure 1.** Coverage of small delay defects depends on the fault’s path of detection and the amount of slack in that path (indicated by the green arrows). Here, path 1 exhibits the minimum slack.

To minimize run time and pattern count, TD ATPG uses a “low-hanging fruit” approach to targeting transition delay faults: It targets them along the easiest sensitization and detection paths it can find, which often are the shortest paths. To understand how this affects SDD coverage, consider the circuit in **Figure 1**, which shows three possible detection paths for a single delay fault. TD ATPG typically generates a pattern sequence that targets the fault along the path that has the largest timing slack, path 3. Notice this pattern sequence doesn’t cover smaller delay defects associated with path 1 and path 2 that would have been covered by targeting the path with smallest slack, path 1.

TD ATPG does manage, however, to detect some SDDs, either directly as targeted faults or indirectly as bonus faults when targeting other faults. Even so, TD ATPG rarely detects delay faults along the longest paths needed to detect defects of the smallest “size” (that is, delay).
In summary, TD ATPG is effective for detecting delay defects of nominal and large size, but because it doesn’t explicitly target delay faults along the minimum-slack paths, it’s not effective in detecting delay defects of relatively small size.

**SDD ATPG**

SDD ATPG, in its purest form, is similar to path-delay testing. It targets each undetected fault along the path with minimum timing slack. In Figure 1, path 1 has the minimum slack, so the pattern generator will target the fault through path 1. If the fault is detected, it is categorized as DS (detected by simulation). The fault simulator also classifies bonus faults as DS if they’re detected along their minimum-slack paths.

If any fault—a targeted fault or a bonus fault—is detected along a path shorter than its minimum-slack path, it is designated as TP (transition partially detected). A TP fault will continue to be fault-simulated each time ATPG generates a new pattern in hopes that a pattern will eventually be generated that detects, as a bonus fault, this same fault through its minimum-slack path. The main drawback with this method is that, in the process of targeting every undetected fault along its minimum-slack path, SDD ATPG wastes time and patterns working on faults that don’t contribute to detecting SDDs.

![Figure 2](image)

**Figure 2.** This sample histogram of transition delay faults in a design shows faults distributed according to minimum slacks.

To illustrate, Figure 2 represents the histogram of all TD faults in a hypothetical design, with the faults normally distributed according to their minimum slacks. For each discrete value of slack on the horizontal axis, a certain number of faults in the design have this value as their minimum slack. As the vertical dotted line indicates, there are nearly 15,000 faults in the design with minimum slack of 0.5 ns. Some 200,000 faults, or 20% of the total, have minimum slack less than or equal to 0.5 ns. Targeting faults with minimum slack in this lower range is an effective way to test for SDDs. On the other hand, targeting faults with much higher minimum slacks doesn’t detect SDDs; the slacks of these faults are large enough that standard TD ATPG covers them very efficiently.

A hybrid approach to pattern generation uses SDD ATPG to target faults having relatively small minimum slack along their minimum-slack paths and uses TD ATPG to target the remaining faults along their easiest-to-detect paths. Synopsys’ TetraMAX ATPG product (Ref. 4) uses a parameter called max tmgn (the maximum timing margin) to assign the cutoff slack level for targeting faults at their minimum slacks. Faults along paths with minimum slack less than or equal to max tmgn will be
targeted by SDD ATPG algorithms, while other faults will be targeted by TD ATPG algorithms.

When the minimum slack of a detected fault exceeds the maximum timing margin, the fault will be classified as TP even if the fault is detected along its minimum-slack path. Narrowing the scope of SDD ATPG to focus on what it does best—testing for the smallest delay defects—reduces both pattern count and run time significantly.

**Reducing pattern count**

In some situations, it may be beneficial to further reduce the pattern count of SDD ATPG. One approach is to “cut ATPG some slack” by letting it generate a pattern that can still detect small (though not always the smallest) delay defects related to a fault. TetraMAX ATPG uses a parameter, max_delta, to control this behavior. If the slack of the detection path for a fault exceeds the fault’s minimum slack by not more than the value of the max_delta parameter, then the fault simulator will classify the fault as DS.

Referring again to Figure 1, if ATPG cannot detect the targeted fault through its minimum-slack path, path 1, it will attempt to target the same fault through a path with larger slack. As indicated in the timing diagram, the slack of path 2 is less than the sum of the fault’s minimum slack and the value of max_delta, so detecting the fault along path 2 will cause the fault simulator to classify the fault as DS. The slack of path 3, however, exceeds the minimum slack of the fault plus the value of max_delta, so detecting the fault along this shorter path will cause the fault simulator to classify the fault as TP.

Increasing max_delta above its default value of zero reduces pattern count because more faults per pattern pass the criterion for DS classifications, and fewer faults need to be targeted by ATPG in each subsequent pattern. Moreover, with each succeeding pattern generated, there are fewer and fewer TP faults for the fault simulator to keep track of, so the simulation run time decreases. A non-zero value of max_delta, however, reduces delay effectiveness.

**Delay effectiveness**

Delay effectiveness is a coverage metric used to quantify how successfully patterns detect faults through their longest paths. Only faults that are detected along paths having slacks less than or equal to max_tmgn contribute to delay effectiveness. These can include both DS and TP faults. TP faults can contribute to delay effectiveness because a fault could be detected along a path with slack less than max_tmgn, even though the slack exceeds the minimum slack of the fault plus the value of max_delta.
Figure 3. A detected fault contributes to delay effectiveness depending on the ATPG parameters \texttt{max\_tmgn} and \texttt{max\_delta}, and the slack of the detection path. In this example, \texttt{max\_tmgn} = 0.5 ns and \texttt{max\_delta} = 0.3 ns. The fault is classified as a DS or TP fault depending on the fault’s minimum slack, the additional slack needed to detect the fault (delta), and the ATPG parameters. The red-shaded combinations contribute to delay effectiveness.

Figure 3 illustrates how a single detected fault can contribute to delay effectiveness depending on the slack of its detection path and the parameters \texttt{max\_tmgn} and \texttt{max\_delta}, which are assumed to be 0.5 ns and 0.3 ns, respectively, in this example. The slack of the detection path is simply the sum of the minimum slack of the fault (horizontal axis) and the additional slack needed to detect the fault, referred to as “delta” (vertical axis).

The entry for each combination of minimum slack and delta indicates whether the detected fault is of type DS or TP, with red-shaded entries representing those combinations that result in the fault contributing to delay effectiveness. The sidebar “Test effectiveness” describes how to calculate test coverage and delay effectiveness for SDD testing and explains how the ATPG parameters \texttt{max\_tmgn} and \texttt{max\_delta} affect these metrics.

Timing is everything
Figure 4. The flow for generating slack data and timing exception data for SDD ATPG involves a four-step process.

SDD testing is feasible only if the ATPG tool can make efficient decisions—often thousands per pattern—based on accurate timing information about the design. Yet, the need to dynamically perform timing calculations inside the ATPG tool (based on Standard Delay Format data, for example) compromises run-time performance and can produce results that don’t correlate well with sign-off timing analysis.

TetraMAX ATPG avoids these issues by accessing data generated by Synopsys’ PrimeTime static timing-analysis tool. PrimeTime models all the key process, physical, noise, and clock network effects required for accurate timing analysis of nanometer designs, so it helps TetraMAX ATPG target SDDs.

Two basic kinds of timing information are required for SDD ATPG: slack data and timing exception data. Figure 4 depicts the four-step flow for generating this information:

1. Standard Test Interface Language (STIL) procedure file (SPF) describes the test-mode protocols for launch and capture.
2. The TetraMAX ATPG DRC (design rule checker) interprets the launch and capture timing requirements and produces an SDC file that defines the launch- and capture-mode timing constraints.
3. (a) PrimeTime interprets the launch- and capture-mode timing constraints and produces an SDC file that defines the test-mode timing exceptions, and (b) PrimeTime generates a report containing the test-mode timing slacks for each fault site.
4. TetraMAX ATPG reads in the slack and timing exception information produced by PrimeTime to generate SDD patterns.

The correct processing of timing exceptions is essential to both SDD ATPG and standard TD ATPG; the combined flow for SDD testing differs only in the requirement to import slack data from PrimeTime to TetraMAX ATPG (step 3b). The flow ensures that the design’s timing constraints reflect the test-mode launch and capture timing, which differs fundamentally from mission-mode timing.
Silicon testing results

The Synopsys flow for SDD pattern generation is now supported in the sign-off methodology at STMicroelectronics, where SDD testing is considered key to reducing DPPM levels across the company’s product lines. The company has found that SDD testing offers improved product quality over standard TD testing (or any other type of test in use today), as evidenced by failure statistics collected for an automotive IC that STMicroelectronics designed and manufactured. The design, consisting of approximately 1 million equivalent gates, was manufactured in a 90-nm CMOS process.

Figure 5

![Venn diagram of test types](image)

Figure 5. This Venn diagram shows the percentage of failing parts covered by each type of test.

displays a Venn diagram of data collected from testing hundreds of thousands of parts using four fundamental types of tests: stuck-at, static bridging, dynamic bridging, and delay tests. The delay tests were composed of standard TD patterns and SDD patterns. The data indicate that about 94.5% of all failing parts were covered by the delay tests, and that 20% of failing parts were covered only by the delay tests.

Upon examining the data for the 20% of defective parts covered only by the delay tests, we observed that 63% of these parts were covered only by the SDD patterns. The remaining 37% covered by TD patterns were also covered by SDD patterns. Test engineers at STMicroelectronics have observed results similar to these for other designs; SDD tests consistently screen more failures than any other type of production test in use at STMicroelectronics, reducing DPPM levels relative to rates achievable using standard TD tests.

Small delay predictions

Recent advances in design-automation technologies have made it possible for semiconductor companies to efficiently target SDDs during manufacturing test. This development has ensured that, in spite of Moore’s Law, very high defect coverage will be achievable on a consistent basis in the years ahead: We expect designers will increasingly adopt SDD testing as the primary means to maintain low DPPM levels as they take advantage of ever-smaller geometries to squeeze more functionality on a chip.
SDD (small delay defect) ATPG produces two distinct coverage metrics to reflect the test effectiveness of the pattern set. One is test coverage, the other delay effectiveness. The test coverage of transition delay faults is the percentage of the total testable delay faults that are detected:

\[ \text{Test coverage} = \frac{\text{DT}}{\text{all faults} - \text{UD} - \text{AU}} \]

where UD and AU are the undetectable and ATPG-untestable faults, respectively, that are subtracted from the total number of faults. DT is the number of detected faults. These include faults in the special SDD ATPG subcategory of TP (transition partially detected), in addition to DS faults (faults detected by simulation) along their longest paths (or paths with slacks satisfying the extended criterion for detection discussed in the section on reducing pattern count).

Narrowing the scope of SDD ATPG by using max_tmgn to specify a relatively small maximum timing margin doesn’t affect the delay fault test coverage in Eq. 1. This is because the total number of TP faults tends to decrease by the same amount as the decrease in DS faults. All other factors being equal, a standard TD ATPG run and an SDD ATPG run should produce the same test coverage.

The second SDD ATPG coverage metric, delay effectiveness, reflects how successful the patterns are in detecting faults through their longest paths. The blue curve in Figure a, labeled $F_T(t)$, plots the same total fault slack data included in Figure 2 as a cumulative distribution up to 0.5 ns. For each discrete value of slack $t$, there are a certain number of faults in the design that have minimum slacks less than or equal to this slack. For example, there are about 200,000 faults with minimum slack less than or equal to 0.5 ns.

The red curve, $F_D(t)$, represents the cumulative distribution of detected faults assuming the maximum timing margin for minimum slacks is 0.5 ns—that is, max_tmgn = 0.5 ns. The faults are distributed according to slacks of their detection paths. For example, out of 200,000 faults with minimum slacks of 0.5 ns or less, ATPG detects about 155,000 faults along paths that have slack that is less than or equal to 0.5 ns, or $F_D/F_T = 77\%$ of the total.

The detected faults in $F_D(t)$ include both DS and TP faults; for both types, only faults that are detected along paths having slacks less than or equal to max_tmgn are included in the distribution. Unless SDD ATPG is successful in detecting every fault along its minimum-slack path, the cumulative distribution of detected faults $F_D(t)$ will always be stretched right with respect to the cumulative distribution of total faults $F_T(t)$. We could have calculated the $F_D/F_T$ ratio at any other slack in the domain to determine the effectiveness of the pattern set. For instance, at $t = 0.2$ ns, the ratio is 72%. Instead of having as many ratios for delay effectiveness as the number of slack intervals, we used a single metric that accounts for the entire range of slacks. Delay effectiveness is typically defined as the ratio of the integrals (Ref. 2):

\[
\text{Delay effectiveness} = \frac{\int F_D(t) \, dt}{\int F_T(t) \, dt}
\]

Referring to Figure a, delay effectiveness is equivalent to the area under the red curve divided by the area under the blue curve, approximately 75% in this example. Narrowing the scope of SDD ATPG by decreasing the maximum timing margin may or may not compromise the delay effectiveness depending on the shape of the cumulative distribution of detected faults $F_D(t)$ in the slack domain [0, max_tmgn]. Increasing max_delta, however, decreases the number of delay faults detected at their minimum slack, stretching $F_D(t)$ to the right and reducing delay effectiveness.

REFERENCES