Diagnosing the 'Undiagnosable' Faults

Phil Burlison
Verigy

Abstract

The manufacturing of complex IC devices is becoming more dependent on determining and solving factors that contribute to excessive yield loss. Analyzing the root cause results of electrical failures during test is now recognized as an emerging industry need. Fortunately, structural test using DfT in the IC design can facilitate this. The general perception has been that this is accomplished by just capturing all of the scan failures and logging the raw data to a file for subsequent analysis by an off-line diagnostic tool. However, an increasingly large portion of faults, specifically those occurring in the "state logic" sections of an IC, are extremely difficult, if not impossible, to diagnose this way. This article describes how new methods have been developed to diagnose these "undiagnosable faults."

Introduction

Understanding failure mechanisms has always been a driving consideration during fabrication of modern semiconductors – both during initial ramp and later during production. In the past, this understanding depended primarily on optical in-line inspection and process parameter measurements. However, with the newer technology nodes, this approach is no longer adequate on its own.

The "systematic" types of defects that occur in the newer technology nodes are increasingly important. A systematic defect is a lithographic design feature which may be too sensitive to the normal variability in manufacturing process to provide consistent results. While DfM (design for manufacturing) techniques provide the technology to manufacture these features, new product designs still tend to introduce new systematic problems. In addition to systematic defects, some types of parametric defects are also more difficult to detect during normal in-line inspection during fabrication.

Since test is the final arbiter about the presence of product-killing defects, its role should be expanded to contribute to identifying what circuit elements are contributing to the failures. Structural test enables this role.

Using Scan for Structural Test of Logic

Figure 1 is a simplified view of the logic sections of an IC. A single scan chain (real-world designs have several scan chains) is used to apply and capture the test patterns created by the ATPG (automatic test

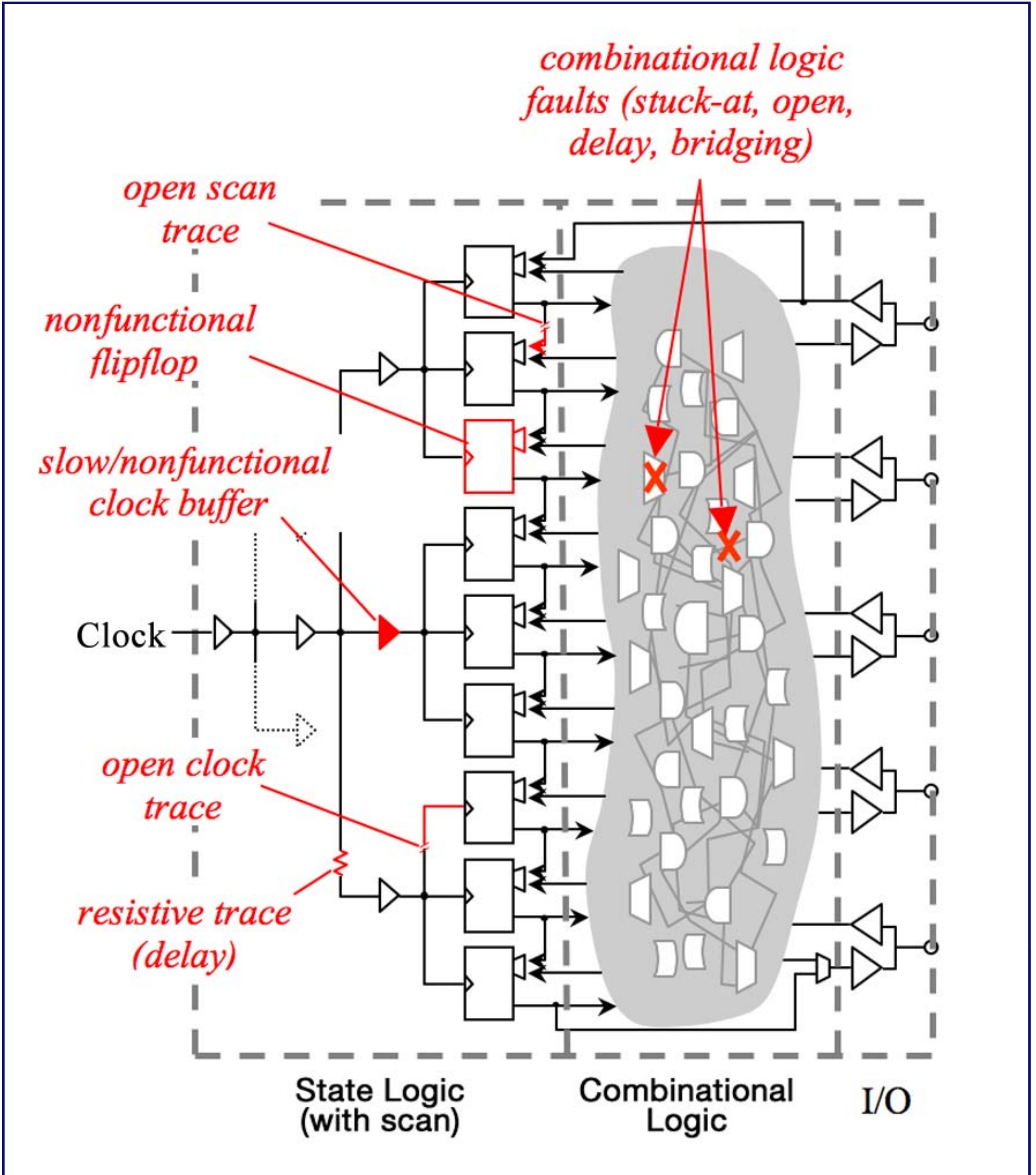

Figure 1. Sections in an IC's Logic

pattern generator) for the combinational logic. A scan pattern is a sequence that shifts the "stimulus" pattern into the flipflops, performs a "capture" cycle where the combinational logic results are clocked into the flipflops, scans the resulting data out and compares against expected results. Delay defects in the combinational logic can also be detected by applying two or more sequential "capture" cycles of test pattern data which cause data transitions in the targeted combinational logic nets/paths. Targeted test patterns can also be used to detect bridging between combinational logic nodes.

These failures are captured by ATE (automated test equipment) and logged to a file. An off-line diagnosis tool (generally the same ATPG tool that generated the test patterns) can subsequently perform fault simulation to identify the candidate combinational logic elements that caused the data results captured by the scan cells on the failing patterns. However, in high-volume manufacturing where large quantities of ICs need to be evaluated, it is much more efficient to statistically analyze just the failing flipflops to determine which have a higher incidence of failures. Then only those failures need to undergo the more prolonged process of identifying the combinational component/net candidates using fault simulation.

State Logic Faults

A fault in either the combinational or state logic of the IC will cause scan test patterns to fail. However, in a sequential logic failure, the results are much more catastrophic. A sequential logic fault will corrupt the test pattern scanned in and applied to the combinational logic, and also be unable to scan out the captured test results. Because of this, chances are there is no relevance between the data shifted out of a scan chain and the expected data in the test pattern. This results in capturing and logging an enormous volume of irrelevant failure data.

Thus, it is generally not practical to use a traditional scan diagnosis approach for faults in the state logic. When test determines that the state logic is failing, generally any additional attempt to capture and analyze results are abandoned, deeming these failures "undiagnosable." However, since 30 percent or more of faults can occur in the state logic area, an alternative has to be found. ATE can be used to interpret the types of faults being observed and can then apply specific tests to identify the location (flipflop) in the logic where the fault manifests itself. This process requires the ATE to have access to specific information about the scan chains and to be able to associate a specific failing cycle in the scan-out data to the flipflop location in the scan chain. This "scan structure" information is available in the test pattern files created by the ATPG tool; it needs to be ported into the test environment.

Figure 1 depicts typical faults that can occur in state logic. These can generally be broken down into stuck-at and delay faults, although there is also potential for bridging faults. To detect a state logic fault, the first test to perform is a "chain integrity" test. This test scans in a simple repetitive pattern, skipping the capture cycle, and tests for the same pattern being scanned out of the chain. This test can detect both a stuck-at fault and a delay fault in the clock tree. Any "stuck-at" fault in combinational logic will result in a constant 0 or 1 data state at the outputs. Typical causes of a stuck-at fault include a defective clock buffer, a defec-

tive flipflop, an open in a clock trace or an open in the scan trace between two flipflops. The ATE can generally isolate the stuck-at location of a chain by observing the data signature coming out of the chains during normal combinational logic test patterns. Testing normal scan output data against the normal "expect" data is not applicable in this case since even a "good" flipflop's data can be corrupted by bad stimulus data. Instead, by noting the particular characteristics of the data stream coming out over a reasonable set of patterns, the ATE is normally able to isolate down to the flipflop in the chain where the stuck-at state resides.

A clock tree delay fault will create an excessive clock skew between two specific state flipflops in the scan chain, with the downstream flipflop's clock delayed from the clock on the previous flipflop. This fault can be caused by a slow clock buffer or a resistive trace defect (e.g., partially blocked via) causing an RC type of delay. This clock skew generally manifests itself

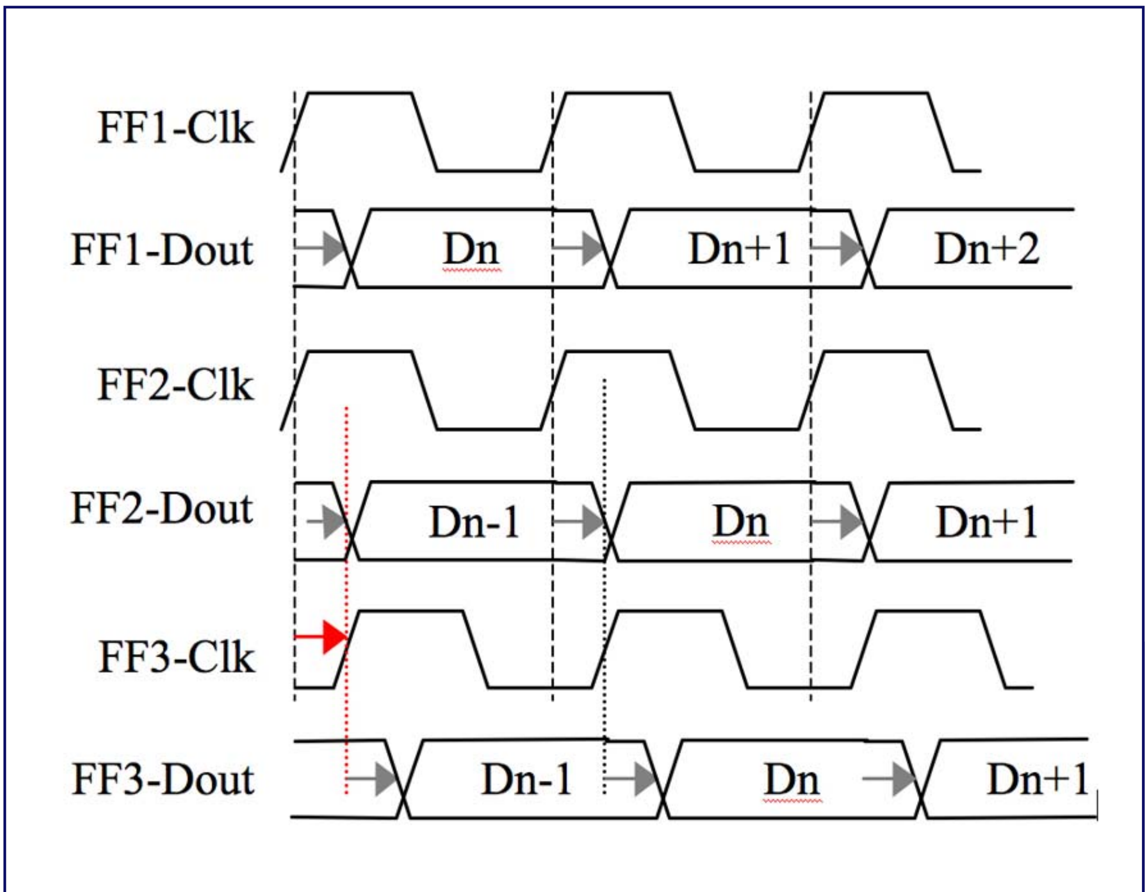

Figure 2. Clock Skew Causing a 'Hold-Time' Fault

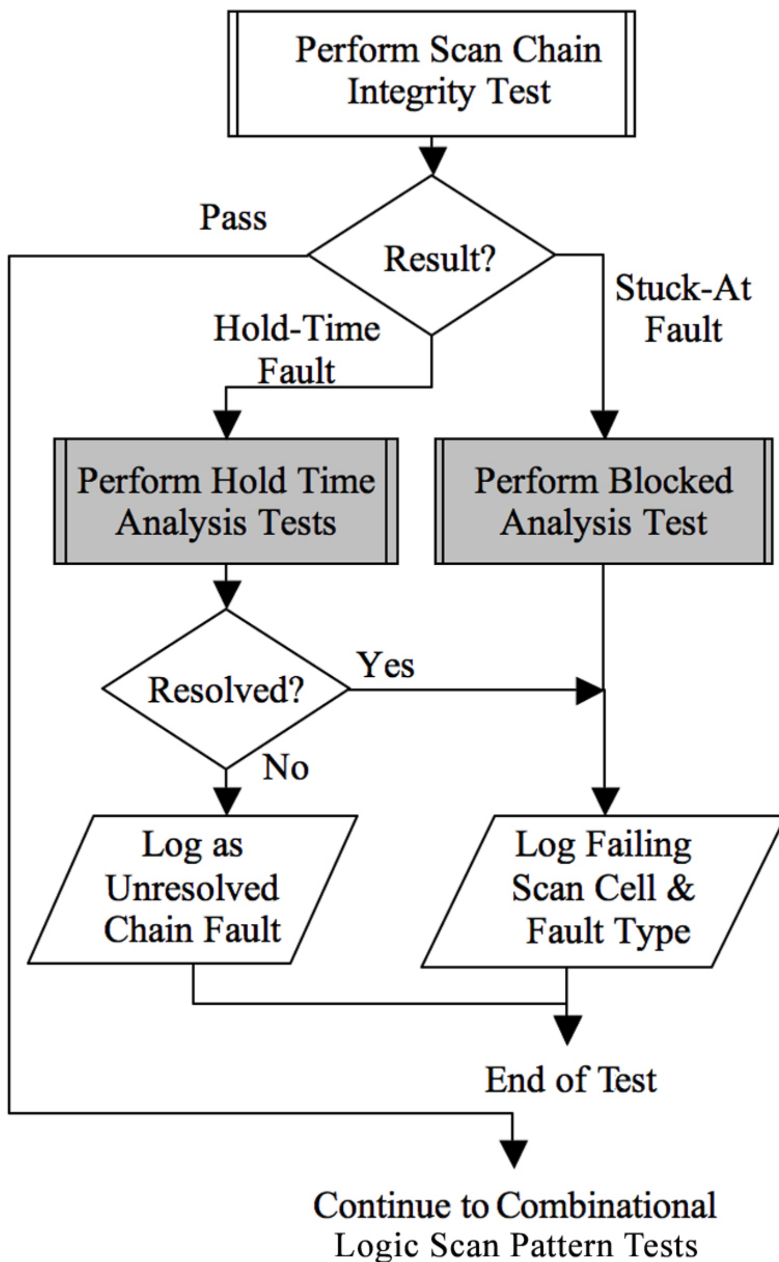

Figure 3. Inserting the Diagnosis Steps in a Test Flow

as a "hold time" failure in the downstream flipflop of the flipflop pair when scanning the data through the chain. Figure 2 depicts this type of fault where the clock on flipflop #3 is delayed to the extent that the low-to-high data transition from flipflop #2 occurs before the clock arrives. This causes flipflop #3 to miss the previous data state that was captured in flipflop #2. While a hold-time failure is relatively easy to detect and distinguish, it is generally more difficult to isolate than the stuck-at fault. Isolation of the specific flipflop with the hold-time failure may be possible on the ATE by varying the Vdd of the circuit to find a "safe" level where the skew is neutralized (i.e., the clock shows up before the data), either by speeding up the clock buffer (with a higher Vdd) or by increasing the flipflop clock to data delay (with a lower Vdd). If a safe operating voltage can be found, the test pattern can be scanned into the chain at this operating voltage. Then, by changing the voltage back to the nominal value where the fault manifests itself and shifting the data out, one can observe at what cycle the data error occurs. If a safe operating voltage cannot be found, other more elaborate schemes can be used that iteratively create test pattern data based on knowledge of the data relationship of "source" and "destination" flipflops.

Applying The Process

Figure 3's representative test program flow demonstrates how these real-time fault analysis processes are brought into action. The first test is the scan chain integrity test. If that fails, the next step is to analyze the resulting data failure pattern to determine if it was a stuck-at, hold-time or unresolvable type of scan chain failure. From there, the appropriate analysis procedures are applied to the IC.

Conclusion

By applying these real-time analysis processes while the failing IC is still accessible to the ATE, it is possible to isolate the faults within the sequential logic much more effectively than any off-line post analysis approach based on fault simulation. The extra time required depends heavily on the architecture of the ATE being used. More important, however, is that the ATE incorporate the required software processes and has access to the IC's scan structure.

About the Author

Phil Burlison

Phil Burlison is a director of advanced technology for Verigy, a leading semiconductor test company. He co-founded Inovys, and was CTO when it was acquired by Verigy in December 2007. Mr. Burlison has 40 years of experience in semiconductor design and test, specializing in grasping the significance of technology trends and applying innovative solutions to address those opportunities. ■