Test is a Four Letter Word

By Bruce Swanson, Mentor Graphics

Test. When you read or hear that word, what do you think of or how does it make you feel? Do you remember your school days where you maybe sweated through a test, exam, or quiz? Yikes, those are also four letter words. Have you ever tested the water temperature of a swimming pool by dipping in your big toe? Perhaps you remember test driving a brand new car for the first time and the new car smell. Or maybe you think of getting the results of a medical test and all of its implications. The connotations attached to that one short word can be many.

If we limit the focus to the semiconductor world for industrial application for example, test still has various meanings. You might want to investigate or emulate different architectures or configurations. Or you could check or analyze the circuit timing. The physical layout could be assessed or examined for potential problems.

I'd like to focus your attention on the testing of a manufactured integrated circuit (IC). How do you test a chip? The first thing to do is to add another four letter word – plan. The time involved with putting together a test plan is essential and well worth it.

Of course no single test plan can be used for everything because every IC design is different. But some common strategies can be considered for the majority of test plans.

A good place to start is to decide which types of test methodologies make the most sense for your design. Things to consider include:

• What type of automatic test equipment (ATE) will be used and what are its capabilities and limitations?

• Are there any limitations on area overhead for test logic or for the number of I/O pins that can be used?

• Can the design be made full-scan?

• What are the test quality metric goals?

• Are there memories and how should they be tested? Are the memories repairable?

• Is there an on-chip PLL (phase locked loop) or other clock generating circuitry? Will these on-chip clocks be used in test mode?

• What types of fault models should you test for?

• For large designs or when using at-speed test patterns, it is common to include some on-chip compression technique to reduce test cost and test time. Does this apply to your design? What compression levels are needed?

• Are there analog or mixed-signal blocks, and how can those be tested?

• Is there a need to test the device within a system, or just at manufacturing time?

- What are the parametric tests that should be done?
- What voltage and temperature ranges should the IC be tested with?
- Is testing done at wafer probe, packaged part, or both?
- Should the test logic include boundary scan circuitry? Can it also be used to control other internal test logic?
- What will be done with devices that fail on the tester? Will failure logfiles be saved and used for diagnosis?

These are some of the main things to consider, but there are certainly more. Some of these questions, or their answers, may lead to still more questions.

In most cases, the completed logic design will require the insertion of scan chains to make the design more testable. Scan chains are created by changing the design's flip-flops and registers into scan cells and then configuring the scan cells into scan chains (shift registers) that interface to the I/O pins which connect to the ATE scan channels. The scan chain cells act as both control and observation points inside the design during test mode. Known logic values are shifted into the device from the tester and then, after the test is performed, the captured responses are shifted out for comparison with expected results.

Another question to ask is whether the device needs to have tests run while it's in the targeted system. If it does, then you could consider using logic built-in self test (LBIST). With the LBIST approach, additional test logic is added on-chip to create the test patterns internally and compare the expected results with the actual test responses. The LBIST technique requires more test logic than basic scan.

Scan insertion can be done by a variety of electronic design automation (EDA) tools. Some do the insertion as part of the synthesis process, and others insert the scan logic after synthesis is completed. There are pros and cons to both approaches.

If you're not using LBIST, the scan-inserted netlist can be used with an automated test pattern generation tool (ATPG) that creates test patterns for a variety of fault models. Fault models mimic actual defects in the chips. The most common fault models used today are the stuck-at fault model and the transition fault model; the latter is a test run at the system clock rate and is often referred to as at-speed testing.

At-speed or transition test patterns are a requirement for most of today's advanced chips at 130 nm and smaller. Unfortunately, they increase the test pattern volume considerably. Because of this, EDA tool providers offer tools and techniques to add some test compression logic around the core design so that tester time and memory requirements can both be reduced substantially. Carefully evaluate the different compression solutions because they are not all the same. Some can easily handle indeterminate states in the captured values of a chip being tested (referred to as “X-states,” or “Xs”), while others cannot.

Many of today's designs also include a large amount of on-chip memory, frequently exceeding 50% of the total chip area. Memories' small feature sizes leave them susceptible to minor defects, so it's imperative to test them thoroughly. Most medium and large memories are tested with memory BIST, and there are EDA tools that can quickly create and insert memory BIST controllers in the design. Be sure to test the memories at full clock speed if possible, and choose a tool that can support a wide variety of memory test algorithms. For small embedded memories where adding BIST logic isn't cost justified, there is a technique called macro testing supported by some ATPG tools that can employ scan chain logic to test memories in lieu of BIST.

This is a quick overview of some test questions and topics that I think should help you plan the manufacture testing of your designs. As the saying goes, “Most people don't plan to fail, they just fail to plan.” By using some simple four letter words like test and plan, you can avoid failing chips.

Author biography

Bruce Swanson is a Technical Marketing Engineer in the Design-For-Test division 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 over 20 years of experience in EDA and computer hardware design.

For details on Mentor's product offerings, please check out Mentor.