Follow these guidelines to design testable ASICs, boards, and systems. (includes related article on automatic test-pattern generation basics) (Tutorial)

From: EDN Date: August 19, 1993 Author: Venkat, Kumar

Application-specific integrated circuits (ASIC) have increased in complexity so it is very important to test them prior to installing them into systems. The higher performance expected of today's chips likewise requires more careful testing. Designers of ASICs have begun implementing a method called design for testability (DFT) that utilizes the test features built into the chips. DFT guidelines emphasize the importance of the insertion of a scan chain into the net list inasmuch as a scan is the basis for many integrated circuit test procedures. Another DFT guideline pertains to the employment of flip-flops and latches. Procedures implemented in the design of testable ASICs are top-level chip partition, the design of a register-transfer-logic/hardware-description language, logic synthesis, scan synthesis, post-scan timing verification and test generation.

Gone are the days of throwing designs over the wall to manufacturing. In this era of complex ASICs, surface-mounted parts, and multilayer boards, the only way to make parts, boards, and systems testable is to design them that way from the start. Here are some guidelines

As ASIC designs get more complex and target higher performance, adequate testing is becoming an issue of increasing concern. Obtaining high fault coverage by testing with manually generated functional-test vectors is simply not practical. Moreover, inadequate chip testing can lead to problems in testing assembled systems, especially those produced in high volume. Even worse, the problems might show up only after systems have been shipped to customers. To avoid such costly predicaments, ASIC designers are adopting an approach named DFT--design for testability. ASIC designers aren't the only ones embracing DFT, however. Taking advantage of the test features built into ICs requires that boards and systems also be designed for testability.

Because most chip-test methods rely on some form of scan, inserting a scan chain into the netlist is the most important DFT step in chip design (see box, "Automatic test-pattern generation basics"). This operation can be completely automated; however, to ensure that the entire chip is highly testable, the overall design process should follow DFT guidelines.

The first DFT guideline relates to the use of flip-flops and latches. To construct state elements and registers, most designs use edge-triggered flip-flops rather than latches. Although latch-based designs can be made scannable, edge-triggered flip-flops are the most popular for scan designs. To support scan inputs, flip-flops can easily be converted to scan equivalents that have built-in multiplexers. All flip-flops in the scan chain must be clocked in the same way--on the clock's rising or falling edge. The cleanest approach is to use rising-edge flip-flops as much as possible.

In designs that mainly use rising-edge flip-flops, you sometimes must use a few falling-edge flip-flops or latches. In such cases, you can achieve high testability by taking special precautions: In the scan mode, if the latches are forced to be transparent and the falling-edge flip-flops are allowed to clock freely, data can still propagate through them. With some ATPG (automatic test-pattern generation) tools, you may have to model the falling-edge flip-flops as buffers. Another alternative is, during scanning, to bypass the outputs of these elements with the outputs of scannable flip-flops.

Designs that use an on-chip phase-locked loop (PLL) for clock generation require a clock-bypass mode if you want to use a test clock for scan tests. Within the chip, you should avoid clock gating so you can completely control all flip-flop clocks from an external input. In addition, to avoid having flip-flops asynchronously reset each other during test, you should either force any asynchronously resettable flip-flops to be inactive in the scan mode or directly control their reset lines from a primary input.

Mutually exclusive control signals

Control signals that select multiplexer paths or activate 3-state line drivers should be mutually exclusive. ATPG tools will not cause conflicts between these paths, but conflicts can occur as scan vectors are shifted in and the chip goes through indeterminate states. The result of these conflicts is high supply current and possible electromigration in metal interconnections. Electromigration can reduce a chip's long-term reliability. Even with mutually exclusive selection of such multiple paths, electromigration can take place if switching between paths occurs frequently (for example, during scan-shift operation) and there is some overlap between drivers as one path turns off and another turns on.

Many ASICs include RAM cells, which most ATPG tools treat as black boxes. Make sure that the RAM does not adversely affect the testability of other logic in the chip, particularly in full-scan tests. One way to assure testability is to observe the address, data, and control inputs by capturing them in scan flip-flops in the RAM and to control the inputs by using scan flip-flops to bypass the RAM outputs (Fig 1). With sequential ATPG, partial scan lets data propagate through the RAM, thus eliminating the need for output bypassing. In either case, the scan does not test the RAM itself. The RAM must be easily accessible from external pins for a comprehensive test using functional vectors.

In designs that contain multiple clock domains driven by different clocks, synchronizers pass signals between domains. In the scan-test mode, one way to handle such designs is to bypass the synchronizers and apply the same test clock to all domains. This approach lets you use a single scan chain throughout the chip. Because different domains use different clocks, clock skew between domains in the scan mode can be high. Adding delay in the bypass path usually compensates for the clock skew and avoids hold-time violations (Fig 2). Another way to handle multiple clock domains is to implement one scan chain in each domain.

[I.sub.DDQ] (quiescent-drain-current) testing imposes two design requirements: To prepare the chip for monitoring small changes in supply current, all static current paths must be turned off under the control of a primary-input pin. Static paths include on-chip PLLs, differential-input receivers that draw bias currents, and sense amplifiers. The second requirement is that the tester be able to detect small changes in the supply current. If your tester does not perform this function, you'll need on-chip supply-current monitoring circuits.

Board-level testing

Board testing has become a difficult problem for several reasons:

* modern pc boards are densely populated and contain complex ASICs

* new chip-packaging technologies do not allow access to ASIC pins

* multilayer boards permit only limited access to internal nets.

Board-level testing requires using ASICs that implement boundary scan so that you can control their output pins and observe their input pins. Compliance with the JTAG (Joint Test Action Group) standard for boundary scan (IEEE 1149.1) ensures that you can test a board on many vendors' automatic test equipment (ATE). As with chip testing, the primary problem is detecting process faults that show up as open nets and short circuits between nets. Boundary-scannable ASICs simplify the testing of board inter-connections. The penalty is increased delay on outputs (and possibly on inputs) and the additional area for boundary-scan registers.

Boundary scan requires that each output signal be multiplexed in a boundary-scan element (consisting of shift and update components). This requirement also applies to internal output-enable signals that control bidirectional or 3-state pins. Each input requires a boundary-scan element that can capture the value driven into the input pin. Fig 3 shows a generic boundary-scan cell you can use as either an input or output cell. Most ASIC libraries provide optimized cells for boundary scan, but you can implement a subset of the JTAG specification with cells built from basic flip-flops and gates.

If boundary scan is required only for board test and you have no plans for using it to verify ASIC functions, you can reduce overhead by implementing only the "extest" and "sample/preload" instructions (in addition to the "device identification" and "bypass" instructions). This approach can eliminate the multiplexer in the input cell because you need to observe only the external input (Fig 4a).

Depending on the system design, you may be able to reduce boundary-scan overhead further. For example, if most or all of the chip inputs and outputs are registered in flip-flops near the I/O pads, the pad registers can double as boundary-scan shift registers (and become part of the internal scan chain for chip testing). In this case, the boundary-scan registers can be clocked only in the JTAG shift and capture states to ensure that the data are retained in all other states. You can also eliminate output multiplexing for the update registers of any unidirectional output that won't affect other onboard devices during boundary-scan shift operations.

Optimizations such as these reduce the chip over-head for board-interconnection testing. Fig 4b shows an optimized output cell for a design that has an output-pad register. With any deviation from the full JTAG specification, the most important requirement is ensuring that outputs that toggle during shifting don't cause conflicts between chips. Such outputs include bidirectional and 3-state pins on ASICs as well as any other 3-state onboard lines that ASIC outputs indirectly control.

Your selection of ATPG tools for chip-test generation will depend on such issues as fault models for test generation, methods of test, foundry support, and vector formats. Quality requirements will determine the range of faults you must consider in test generation. Base a decision on whether to use full or partial scan on area and timing constraints and keep in mind that fault coverage also indirectly affects this decision.

Once you have made the basic decisions, you must apply additional criteria to select your tools. Actual fault coverage in test cases is obviously important for comparing tools. You may have to use a proven fault simulator to confirm the fault coverage reported by each tool. In large projects, test-generation speed is another criterion of practical importance.

Good design-rule checking to provide feedback to designers is valuable for making designs more testable. Frequent rule violations include clock gating, combinatorial loops, asynchronous resets, and clocks that primary inputs cannot control.

The following is a brief summary of the capabilities offered by some commercial tools:

* Sunrise Testgen (Sunrise Test Systems, Sunnyvale, CA): Generates full- and partial-scan vectors for the stuck-at fault model. Designers can exclude scan in selected areas and determine the percentage of flip-flops to be scanned. Testgen now also provides [I.sub.DDQ] fault testing.

* Aida (Crosscheck Technology, San Jose, CA): Generates full-scan vectors for the stuck-at model, provides [I.sub.DDQ] fault simulation for full-scan vectors, and supports double-latch-based full scan for delay faults.

* Test Compiler (Synopsys, Mountain View, CA): Generates full- and partial-scan vectors for the stuck-at fault model. Partial scan considers overall area and timing constraints in selecting scan elements.

Most ATPG tools include scan-synthesis software that inserts full- and partial-scan chains into nonscan netlists. Make sure that scan-synthesis tools consider timing and physical-design issues. For example, global scan-control signals must be buffered properly to avoid excessive delays and slow rise and fall times. In some designs, clock skew can be so large (>0.5 nsec) that, under fast conditions, it causes hold-time violations within scan chains. This problem might call for adding delays along the scan paths based on timing analysis. Because the scan chain takes up significant area, you may find it necessary to use floorplanning or a design hierarchy to determine an optimal threading of the chain. For designs implementing JTAG boundary scan, synthesizing the boundary-scan chain must be part of overall scan synthesis.

DFT, scan synthesis, and ATPG must be integrated smoothly into the overall ASIC-design approach and tool set. The following list identifies where these test-related activities fit into the design process and how they affect the process.

Top-level chip partition: Partition the chip into logical areas that mesh with the scan implementation. Typically, these areas include the core of the chip, the periphery or boundary, any on-chip scan-control logic, and any on-chip clock-generation circuits (PLLs, for example). In the core, you find all of the internal scan logic. In the periphery are all of the boundary-scan elements. These elements could also include the common-pad registers that are used in boundary scan, internal scan, and the normal operating mode. The clock-generation circuits include provisions for bypassing the PLL with a test clock.

RTL/HDL design (register-transfer-logic/hardware-description-language): To ensure a highly testable design, you must apply all of the relevant DFT guidelines at the RTL level. Most guidelines are so fundamental that you cannot apply them at a later stage.

Logic synthesis: Enforce the choice of flip-flops and other storage elements at this stage. The ASIC library must contain scannable equivalents of all flip-flops that you will later replace with scannable cells. Perform pre-scan timing analysis and optimization at this stage. Make some allowance for scan. Area estimates must also allow for scan overhead.

Scan synthesis: The scan-synthesis tool will convert flip-flops into scannable equivalents and thread the scan chain. If the chip is partitioned as described above, you can do this task separately on the core and on the boundary. You should use a design hierarchy or a floorplan to optimize the scan-chain order and routing. The netlist interface for your scan-synthesis tool (input and output formats) must be compatible with the other tools you're using.

Post-scan timing verification: Perform final timing verification with all of the scan overhead included.

Test generation: The ATPG tool will generate test vectors for the full chip. The netlist and test-pattern interfaces must be compatible with the other tools and the foundry requirements. As a final verification step, perform a logic simulation on the netlist using at least a portion of the vector set.

References

[1.] Abramovici, M, M A Breuer, and A D Friedman, Digital systems testing and testable design, Computer Science Press, 1990.

[2.] Williams, T W and N C Brown, "Defect level as a function of fault coverage," IEEE Transactions on Computers, Vol C-30, No. 12, December 1981.

[3.] Varma, P, "Test generation for nonclassical faults," Teradyne users' group, April 1992.

[4.] Soden, J M and C F Hawkins, "Electrical properties and detection methods for CMOS IC defects," First European Test Conference, 1989.

[5.] Dervisoglu, B I and G E Stong, "Design for testability: Using scan-path techniques for path-delay test and measurement," International Test Conference, 1991.

[6.] Venkat, K, "ATPG Tools Are a Varied Lot," Electronic Engineering Times, September 21, 1992.

[7.] IEEE standard test-access port and boundary-scan architecture (IEEE Standard 1149.1,) 1991.

[8.] Broderick S, and K Wills, "The intricacies of boundary-scan solutions," Evaluation Engineering, September 1991.