At-speed Testing of SOC ICs

Vlado Vorisek, Thomas Koch, Hermann Fischer
Multimedia Design Center, Semiconductor Products Sector
Motorola Munich, Germany

Abstract

This paper discusses the aspects and associated requirements of design and implementation of at-speed scan testing. It also demonstrates some important vector generation and implementation procedures based on a real design.
An innovative method of scan pattern timing creation based on the results from Static Timing Analysis is presented. The paper also describes the usage of a clock control module on J750 tester, which creates fast clock by combining two tester channels with high edge placement accuracy.
These methods allow a short test pattern preparation time and the use of low-cost test equipment, while providing the high quality at-speed testing.

1. Introduction

High application frequencies, automated timing closure, new deep submicron (DSM) effects (dominance of wire delay, crosstalk, etc.), and finally the decrease of the absolute slack on the critical paths require testing of the IC at application frequency. On the other hand, the test cost must be kept on a minimum involving low cost tester solutions. Additionally, due to the short time-to-market window, all the tasks must be fulfilled just or even before the first prototypes arrive from the wafer fab. This paper presents a possible solution to this challenge: scan based ATPG testing at application speed (at-speed).

The first part of the paper (Section 2-3) contains a methodology tutorial which summarizes state-of-the-art of at-speed scan. The second part (Section 4-6) presents the author's contribution based on a real case design. Particular sections are organized in the following way: Section 2 discusses scan test and associated fault models. It defines the at-speed test as it is understood in this case. Section 3 presents the requirements that need to be met in order to implement proposed test method. Special scan hardware architecture is described there as well. Section 4 focuses on the most challenging tasks of pattern creation process, namely pattern timing creation, pattern verification and debug. Finally, Section 5 shows how low cost tester equipment can be used to implement the proposed at-speed test method. At the end, Section 6 draws some conclusions.
The case design is a real production audio processor supporting MP3 playback. The architecture is centralized around a Motorola 32-bit RISC ColdFire V2 processor, which has the following attributes:

- Fabricated in 0.18µm CMOS technology
- Maximum application frequency is 140MHz
- 96kB of memory + 8kB of instruction cache RAM
- 5 clocks (2 uses posedge and negedge) + PLL

2. At-speed scan test and the fault models

Scan test in general is a structural methodology to detect manufacturing defects by using Automatic Test Pattern Generation (ATPG). The test has two operation modes: shift and capture. In the shift mode, all sequential elements are connected in one or more shift registers. The shift mode is used to control the circuit and observe the result. In the capture mode, the values stored in the sequential elements during shift are propagated from the source via the functional logic to the sink. Each pad or sequential element in the scan chain is able to operate as

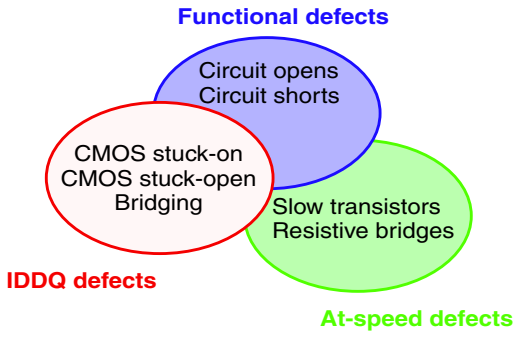

Fig. 1 Manufacturing defect space

Proceedings of the Design, Automation and Test in Europe Conference and Exhibition Designers' Forum (DATE'04)
1530-1591/04 $20.00 © 2004 **IEEE**

source as well as sink.

An ATPG tool provides methods to detect different manufacturing defects. Fig. 1 gives an example of possible device defect classes. In DSM technologies, there is an increasing number of faults related to speed defects. One contributor to the increasing number of speed defects is the decreasing margin between inaccuracy of timing calculation and minimum gate delay.

2.1 Stuck-at fault model

The most common fault class is the stuck-at fault class. The fault model covers functional defects generated by shorts or opens in the device interconnect.

ATPG programs always use the single stuck-at 0/1 fault model for the vector generation.

2.2 Transition delay fault model

The transition delay fault model enhances the stuck-at fault model by introducing a time assessment. It is possible to launch the transition by a change in one or more sources (registers or pads).

The transition delay fault requires three test cycles (see Fig. 2) for detection:

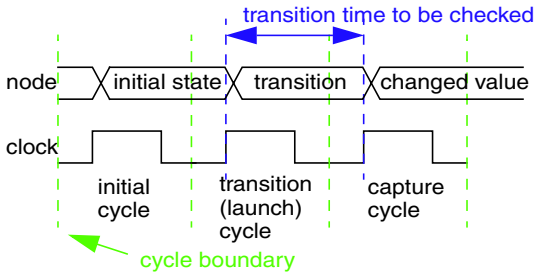

Fig. 2 Transition delay waveform (generic)

- **Initialization cycle** sets the initial value for the fault.
- **Transition (launch) cycle** triggers the transition to be verified.
- **Capture cycle** is identical to the stuck-at fault pattern, receives the final transitioning value from the source to the sink.

2.3 Path delay fault model

Unlike other fault types, path delay faults do not have localized faults, they test the AC performance of specific paths. They are often used for speed selection of a device. Nevertheless, there is a similarity to the transition fault model. The path delay fault model requires a state change from the launch event to the capture event. To get a valid path delay vector, a single change of a source (register or pad) must trigger the transition of the path.

2.4 IDDQ fault model

The IDDQ test measures the quiescent supply current to detect failures not easily found by functional testing (CMOS transistor stuck-on faults or adjacent bridging faults), which result in a higher power supply current. The high leakages of the small process geometries question the accuracy of this fault class.

2.5 Definition of at-speed scan test

In contrast to running the chip at application speed, at-speed testing means that the time between launch event and capture event is one application period apart (see Fig. 7 in Section 5). The clock period of the test program does not necessarily need to be the application period.

One of the reasons the test period might not be equal to the application period, is that the cost of testing at high clock rates increases dramatically (expensive test equipment) as well as introducing problems with loadboard design. In addition to that, the pin load of a tester is much higher than the application pin load, this limits the tester period as well.

3. Design requirements for at-speed test

At-speed scan testing requires special DFT methodologies to be considered before starting the RTL design entry. A few important issues related to the at-speed topic are discussed in this section. Regular scan requirements derived by the DRC rules must be followed as well.

3.1 Multicycle timing

For at-speed scan testing, multicycle paths should be avoided (all timing paths fit into one clock cycle). This requirement is driven by the current ATPG tools which assume that all paths are single cycle.

3.2 Uncontrollable logic

Other areas of consideration are uncontrollable parts of the design during scan mode. These parts may decrease the fault coverage (e.g. by ‘X’ injections into the logic), increase the vector count or can even disable scan testing at all. There are several methods to minimize the influence of such blocks. All these methods have one common drawback, they require additional test vectors to verify the undetected faults within the uncontrollable logic, or at the interface of the uncontrollable logic.

The simplest way is to ensure that all outputs of such blocks drive a known value during scan test by just adding an AND or OR gate to the output and control this gate by the scan mode signal.

Another method is to bypass the block. This means that the inputs are connected to the outputs via a bypass multiplexor controlled by the scan mode signal. This method is often selected for RAMs, where the data input is connected to the data output.

A third possibility (for RAM/ROM) is "write through". If used, the ATPG tool must support clock sequential test pattern generation. It is good practice to modify the RAM/ROM interface to support this feature. The RAM/ROM itself should be verified by Build in Self Test (BIST) to detect manufacturing defects within the structure.

3.3 Transition delay and path delay requirements

ATPG tools support two different modes for the transition delay vectors. The first mode is known as "last shift" or "launch-off shift" mode. It uses the common combinational ATPG engine. The cycle before the last shift acts as the initialization cycle. The last shift is used as the launch cycle. The measured transition occurs between the clock in the launch cycle and the capture cycle. This time must be equal to the application period. It requires an at-speed scan enable signal.

The second mode, known as "system clock" or "launch-off clock" mode, uses the clock sequential engine. It has more relaxed timing on the scan enable signal by adding a dedicated launch cycle.

Both modes should be used (allow ATPG to choose) to get the highest possible fault coverage.

Because the narrow window from the cycle boundary to the capture event is often too small for the IO paths, all faults on the pad input logic need to be masked for at-speed transition delay testing. The high load on the output pads, due to the tester environment, also requires the masking of fault in the pad output logic for at-speed transition delay testing.

3.4 Stuck-At Faults Requirements

By definition, every stuck-at fault vector which causes a '0' to '1' or '1' to '0' transition from the last shift to the capture cycle is also a valid transition fault vector. These faults should be covered by the transition fault engine. However, not all possible transition faults are covered due to CPU or vector size limitations. These remaining faults may be covered by the stuck-at pattern. Due to this fact, it is good practice to run the stuck-at test at-speed (if the design supports the "last shift" mode) to add the timing assessment as well. For any fault which does not cause a state change from the last shift to the capture cycle, either running at-speed or running at a lower frequency will not make any difference in the fault coverage.

The chip may contain faults which require a clock sequential algorithm, i.e. faults around memories, to increase the fault coverage. Running the stuck-at vectors at-speed for this kind of faults make sense, because these faults may contain transition faults not recognized by the transition fault engine.

3.5 Scan architecture of the case design

Fig. 3 shows the scan architecture used for at-speed stuck-at, transition and path delay vectors. The circuit uses an at-speed scan enable signal (SE) to control the shifting and a Bus Scan Enable signal (BSE) to control the IO logic [1].

Achieving a scan enable with a small propagation delay is not always a simple task. One possible solution is to insert one scan enable per clock domain to meet the timing constraint by reducing the fanout of each scan enable.

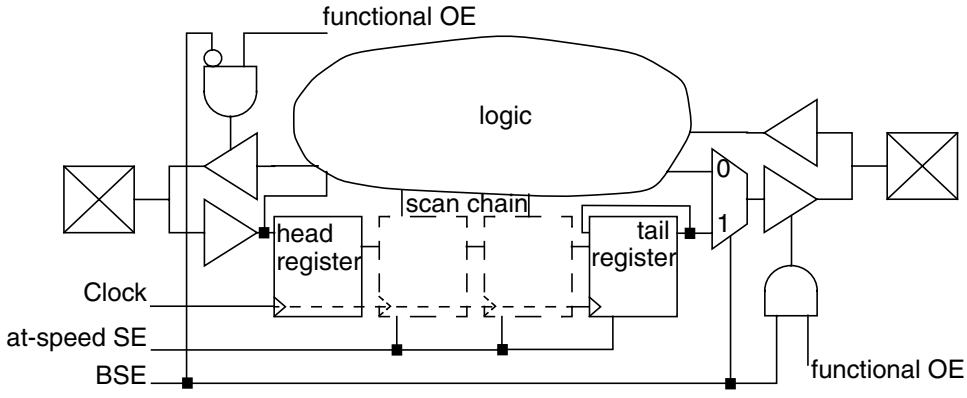

Fig. 3 At-speed scan architecture

Due to the fact that the separate scan domains are not evenly distributed around the chip, it is good practice to use a separate signal called BSE. This signal is used to control logic for the scan inputs and scan outputs instead of using the individual scan enables for this task. Having this BSE signal is helpful in meeting timing constraints for the at-speed scan enables.

For at-speed scan testing the BSE signal is forced to '1'. While running special BSE vectors covering faults in the IO logic, the BSE signal is forced to '0' during the capture cycle. Such vectors usually do not run at-speed.

To cope with the input delay of the scan enable through the pad cells, a head register is added to the scan chains which allows to switch the scan enable during the last shift cycle without losing any fault coverage (head register contains the value from the first scan register).

To deal with the "slow" output pads, a tail register is added to the scan chains with a "hold function" which ensures that the output pad will not change its value during the capture cycle as long as the BSE signal is '1'.

4. Test pattern creation

The detailed explanation of whole at-speed pattern generation flow is outside the scope of this paper. This can be referenced in [1]. Rather, the focus is on the most challenging tasks only, especially, the at-speed pattern timing creation, pattern verification and debugging flows.

4.1 At-speed pattern timing

Running the pattern at-speed is in most cases right at the edge of circuit maximum speed. The Static Timing Analysis results were used to properly design the timing waveforms for at-speed patterns. The exact clock, scan enable, and scan input waveforms can be then calculated based on the insertion delay and skew numbers of these signals.

This is very important if at-speed capture time is close to the insertion delay of the clock (see Fig. 4 and Fig. 5). Then the active edge (pos or neg) of the shift cycle can actually arrive at the flipflops in next cycle. Cycle boundaries need to be viewed here from the device point of view (dark black waveforms). Gray waveforms in Fig. 4 and Fig. 5 shows the same timing from the tester/pattern point of view. Two things that are especially important in such situation are the time window for Scan Input (SI) and Scan Enable (SE) signals.

To be able to properly shift through the scan chains, the SI signals need to arrive at the first flipflops within the time window for SI (see Fig. 4). This window is usually limited by the active shift edge from the left side (negedge in Fig. 4) and by the strobe time of Scan Outputs from the

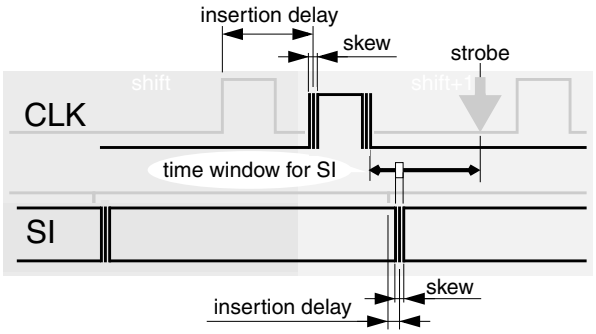

Fig. 4 Time window for Scan Input signals

right side. The strobe time limits this window because ATPG tools usually require a force-strobe-pulse event order, which means that the strobe time must occur before the clock pulse. SI arrival time can be set by adjusting the force time in the ATPG protocol file. The insertion delay and skew of the worst SI path must be considered here as well.

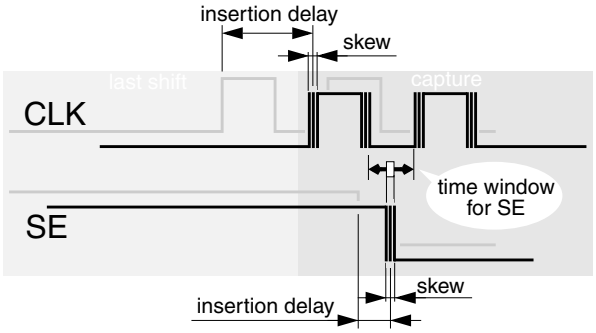

Fig. 5 Time window for Scan Enable signal

To be able to properly switch all the flipflops into the capture mode, SE signal need to arrive within the time window shown in Fig. 5. This can be set by adjusting SE waveform in the ATPG protocol file. The insertion delay and skew of SE signal must be considered as well.

To avoid these kinds of problems earlier in the design, the time window for SE can be partially relaxed by separating the posedge and negedge flipflops into two separate scan domains. The posedge and negedge scan domains then need to have independent scan enable signals, so they can both have unique waveforms. Using last shift launch ATPG mode can relax the time window for SE signal as well.

Described pattern timing creation flow minimizes required ATPG iterations and debugging activities, which saves a lot of time and effort during the design process.

4.2 Scan pattern verification

The generated patterns are usually simulated by an ATPG tool. However, it is still necessary to verify the patterns with full backannotated timing (often referred as pattern re-simulation) to check for setup and hold time issues and to identify possible false paths. Both best and worst case conditions must be used. To ensure proper timing, the timing netlists should be extracted with the same (higher) capacitance pad loads as the targeted tester with a loadboard, to simulate the tester environment as closely as possible. This test pattern verification strategy gives the ability to shorten the test program creation time and saves a lot effort during test pattern debugging.

Because the complete simulation of long scan patterns would take very long time, only first few vectors are simulated in serial mode to verify shifting, and the remaining patterns only in parallel mode. In parallel simulation all scan chains are forced/readout in just one cycle, which drastically reduces the simulation time.

4.3 Scan pattern debugging

Even if every clock domain is separated into its own scan domains, some patterns may fail in the timing simulation due to false paths. These are the paths which are not executed in application mode (no timing constrains applied) but can be accidently examined during scan mode. All such false paths need to be understood and properly handled (ATPG capture masking or call constraining). To avoid re-generating the patterns again and again, to keep the simulation time short, and the dump file size small, it is recommended to use the debugging flow as shown in Fig. 6. In this flow, an ATPG tool reads the pattern in and writes the same pattern out with modified timing waveforms. Also, only the failing vector is simulated and analyzed during the debugging. A recommendation is to start from the first failing vectors of worst case serial resimulation. By fixing the first failing vector, many others very often also disappear. It should be kept in mind that full serial simulations of just a few scan vectors (middle size design) may easily take several hours and could produce a very large gigabyte sized dump file. Therefore it is very important to go through failing vectors one-by-one and never try to fix two different issues at the same time.

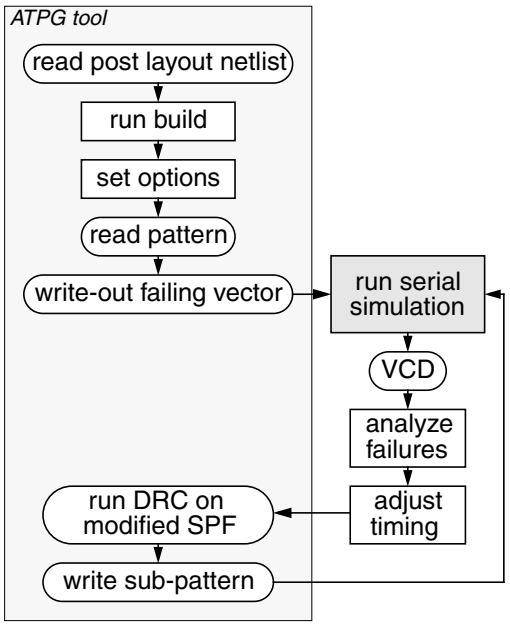

Fig. 6 Pattern debugging flow

5. Test program and tester implementation

One of the challenges of SOC testing is cost. The price of Integrated Circuits is strongly influenced by the Automated Testing Equipment (ATE) capital cost.

In order to keep the test cost down, a low cost tester platform was used for probe and final test. Teradyne J750 tester with a maximum test frequency of 100 MHz was chosen.

The IDDQ test pattern and the RAM and ROM BIST patterns could be ported to the tester without limitation, because the test frequency was below 100MHz. In the at-speed scan test pattern however, the time between launch event and capture event is around 7ns, which is equivalent to a frequency of 143MHz. How can the test be performed under these circumstances?

As mentioned before, the test can run at lower speed, as long as the application speed can be maintained for one cycle. This kind of testing is often called Slow-Fast-Slow-Test (Fig. 7) and relaxes the tester requirement, because only one tester channel has to run at-speed for a single cycle [2], [3].

Because J750 is not able to generate even a single 7ns cycle but offers a high edge placement accuracy, the following method was applied.

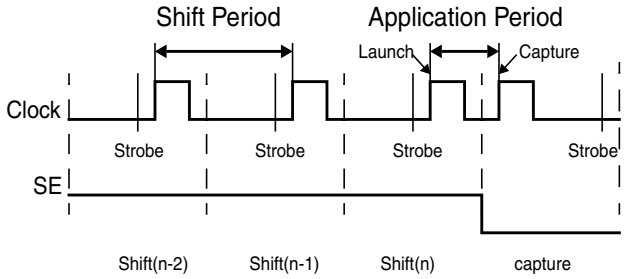

Fig. 7 "Slow-Fast-Slow" timing diagram

In the at-speed scan pattern, a postprocessing script subdivides the clock signal into two independent signals, ClockA and ClockB. These are generated with two separate digital tester channels, fed onto the tester loadboard, and combined with a XOR gate in order to regenerate the original clock shape. This signal is then fed to a bipolar pin driver device with exceptional slew rate and propagation delay specifications, and a variable output voltage range (Fig. 8). The desire to eliminate propagation delays is to minimize the offset that must be applied to the tester values.

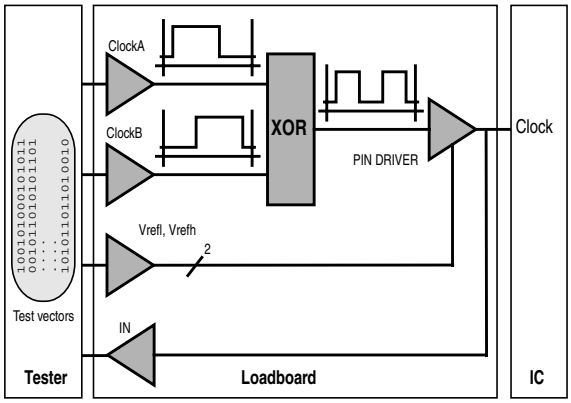

Fig. 8 Tester environment

Two advantages of this method are the larger pulse widths of the two individual signals which get less degradated than a single narrow pulse, and the possibility to "refresh" the clock edge rates close to the DUT.

For Vih and Vil level control, and for calibration purposes, three more tester channels are needed. The calibration routine measures the path lengths and skews of ClockA and ClockB and applies them to the calibration and deskew registers for best accuracy.

Testing multiple devices in parallel increases throughput, which contributes to the reduction in testing costs. Therefore a dual-site test methodology was used to test two parts at a time. On the loadboard, two Teradyne 200 MHz Clock Modules, which work on the described principle and mainly consist of an ECL-XOR gate, a high-speed driver, glue components, and drop-in calibration software modules, have been integrated.

For debugging of the loadboard, the calibration software and the postprocessing script, a 200MHz test sequence was used. Prototypes could be verified at-speed within a few days, loadboard and test program are successfully running in a production test environment.

6. Conclusions

The at-speed scan test motivation, background and main implementation guidelines have been presented in this paper. The innovative way of scan pattern timing creation based on the results from Static Timing Analysis was presented. This new approach saves a lot of time and effort during the design phase.

The paper also describes the usage of the clock control module of the J750 tester as an example that high quality at-speed test can be implemented using available commercial tools and a low cost tester. The proposed at-speed technique has been very successfully implemented into real production design. All at-speed patterns were successfully running on the tester on time.

7. References

[1] V. Vorisek, T. Koch, "At-Speed ATPG for SOC-Designs", SNUG Europe, Paris, France 7-8 March 2002

[2] T. Chakraborty, V. Agrawal and M. Bushnell, "Delay Fault Models and Test Generation of Random Logic Sequential Circuits" in Proceedings of Design Automation Conf., pp. 453-457, 1993.

[3] H. Fischer, "Delay-Fault-Test", Test Kompendium 2003, publish-industry Verlag GmbH, 2003