Debug War Stories

Darrell Carder
Freescale Semiconductor, Austin Texas
darrell.carder@freescale.com

1. Introduction

Anyone that has been involved in Semiconductor test has their own debug war that they have survived. The worst of these usually occur the week before you are leaving on vacation with a customer in line down situation. It use to be that debugging customer failures would involve exchanges of code sections with the customer, timely set up of functional test patterns for containment in production and exercising the fault for failure analysis. With high fault coverage scan methodology a large part of the effort can now be achieved very efficiently using the scan-based test logic.

2. Setup

The debug war story I’m relating to for this pane is based on a customer return device. During product acceptance testing the customer identified issues on several devices when writing to memory. Re-testing of the failing devices to all functional, memory / memory BIST, and scan patterns passed for all test conditions.

3. The War Story

With all patterns passing, how should the debug effort start? The first step was to duplicate the customer failure. Portions of the failing code were supplied by the customer and when ran on an evaluation board confirmed the failure. Analysis of the code indicated that the memory write was a late write based on the completion of an internal timer. During the wait period of the timer all clocks are gated off. Adjustment of the timer value identified that for a shorter wait period the device would start working.

After duplicating the failure and gaining additional knowledge on conditions for failing, the logical location of the failure was next on the battle plan. To help identify the failure location between digital logic and memory what would be the next step be? Our approach was to utilize the customer’s functional code on a Tester and make use of the scan structures of the design. For this device all clocking was supplied by the tester and the scan mode entry is via static combinational logic. By selecting different stop points in the code the functional code and clock would be stopped, scan mode entered and the scan chains unloaded. Comparing runs with long wait times (failing condition) and short wait times (passing condition) a set of scan registers were identified that diverged between the two runs. The logical location of the failing scan registers were on an input databus and control logic of the memory. All failures were of the same library cell type. Now the failure mode conditions are taking shape. Failure is: dependent on wait time with no clock (clock held low); prior to actual write to memory (indicates digital logic); fails when good data is low; all failures are of same library cell. What type of a failure mode is this indicating?

With the latest battle ground intelligence the war strategy changed to physical fault identification and containment. The failure information was submitted to physical failure analysis. The task of containment again looked at utilizing the scan logic to simplify the task. The scan chain test was utilized for a containment pattern by adding repeats during the clock dead cycle between load and unload of the scan chains, and loading all chains with 0’s. The result caught some of the failures but not all. Why did some of the failures pass the containment pattern? Scan chain inversion! To compensate for inversion on the scan chain a second containment pattern was generated, additional failures were caught but not all that were failing the functional code. Now what? This specific design contained negative edge clock domains generated by inverting the clock and using positive edge registers. The last registers to correlate were on a negative edge clock. The four final containment patterns included load data of 1 and 0 with clock stopped while in low and high states. Results caught all failing registers seen with functional code with some additional failures of same library cell in other logic.

With scan-based containment patterns in place the next step was to identify corrective actions. Physical FA indicated a narrow active area in the failing register that was being pinched off causing an open in the pull-down path of the register. OPC changes were done to fix the narrow active region.

4. Victory

The debug war was won! This story shows that even though the beginning of some debug battles look to be next to impossible to progress. The use of scan logic can greatly improve the odds. In other battles the use of scan diagnostics have helped identify tester timing issues, resolve functional timing issues. Adding volume diagnostics and analysis has correlated wafer edge failures to chain failures and cold gate failures.