PARADIGM®
WORKS

At Speed Testing with Scan Vectors

At Speed Testing With Scan

At .13 micron and below IC manufacturers are starting to see more defects that are not caught by traditional stuck at fault testing. Defects like high impedance metal, high impedance shorts and cross talk are not caught by traditional stuck at scan vectors. Instead they show up as timing failures that can only be caught by at speed testing. Some ASIC vendors require at speed test vectors and many others have plans to add that requirement.

In the past at speed testing was usually done by running a small number of functional vectors. This approach is time consuming and it produces poor coverage. DFT tools from EDA vendors can be used to generate at speed scan vectors with good coverage. These tools allow two types of at speed testing, Transition Delay testing and Path Delay testing. Both work by generating scan patterns that can be scanned in at a slow speed. After a scan vector is scanned in, two or more capture clocks are applied at full speed and then the captured result is scanned out, usually at slow speed.

Launch-Off-Shift vs Broadside

There are two ways to transition from scan shifting to capture clocks. With the launch-off-shift approach one clock is generated exactly one at speed clock cycle after the last shift clock. This can work with muxed flop scan chains but it requires that the scan chains run at speed and the Scan Enable signal must work at speed. The launch-off-shift approach doesn't really make sense for LSSD designs. The launch-off-shift approach is also sometimes called last-shift-launch.

With the broadside approach a pattern is scanned into the chains at a slow speed. Some time after the pattern is scanned in, two or more at speed capture clocks are applied to the scan flops. Then some time later the pattern captured in the scan flops is scanned out at slow speed. This approach allows the scan clocks and the Scan Enable signal to run at a slow speed. It also works well with LSSD designs.

Clock Control Requirements

As with any DFT tool you need to understand what requirements it places on the design of the chip. If the chip is already designed to use scan and you plan to use broadside capture, the only additional requirement is clock control. It must be possible to control the clocks so that they produce two consecutive at speed clock pulses. If the clocks are derived internally there must be an external pin that, when asserted, causes the clock logic to generate two at speed clock pulses. If the clock is sourced externally you need to be sure that the target tester can run at the required clock speed.

In some cases it is useful to use more that two consecutive at speed clock pulses. This allows the loaded patterns to pass through non scanned flops and latches on the way to a scanned capture flop. The number of at speed capture clock pulses is usually less than ten. As this number increases the run time and memory requirements of the ATPG tool increase quickly.

Transition Delay Methodology

Transition Delay testing is used to test as many paths as possible at speed. The DFT tool inserts transition faults for all cell inputs in the chip and determines which flops to use for launch points and capture points. From the users perspective, the process used to generate these scan vectors is very similar to generating scan vectors for stuck at faults. Transition Delay vectors do not replace stuck at fault vectors but they can reduce the number of required stuck at fault scan vectors. This is done by generating Transition Delay vectors first and then generating stuck at vectors for faults that were missed by transition delay vectors.

Path Delay Methodology

Path Delay testing is used to test known critical paths at speed. For Path Delay testing the DFT tool generates a pattern which, when scanned into a chain, sensitizes the path of interest and sets up the value to be sent through the

path. Many of the gates in the path will have several inputs in addition to the one that is part of the path and the scan vector must also control those inputs. Sensitizing the path means that those additional inputs will be driven to values that allow the signal from the launching scan flop to propagate to the capture scan flop. After the pattern is scanned in you must generate two or more capture clocks whose period matches the operating frequency. Then the chain is scanned. The test passes if the capture flop captured the correct values. Scanning a pattern into and out of the scan chains can occur at slow speeds. Most DFT tools can not generate Path Delay scan vectors for paths that include RAM. This may change in the future since you can create RAM models for FastScan or Tetramax that allow these tools to control RAMs. Both FastScan and Tetramax require an additional license to generate path delay scan vectors.

Path Delay Flow

The path delay flow starts with the output of a static timing verifier. This information must be converted into a .pathdef file that is essentially a netlist for each flop to flop path or flop to output path with critical timing. There are scripts available that read in a Prime Time report and generate .pathdef files for FastScan or Tetramax. The DFT tool reads in the .pathdef file along with the .do file, the .testproc file and the verilog netlist.

The DFT tool checks the .pathdef file to verify that it can trace the path. It then checks whether the path is real. A surprising number of the paths reported by static timing analyzers are not real paths. When the DFT tool checks the path to verify that it can be sensitized it often finds that one gate input in the path needs to be forced high and another needs to be forced low. If those two points have the same source then the path is not real. After the paths are verified the ATPG tool attempts to generate scan vectors to test the valid critical paths.

Even if the chip clocking does not support critical path analysis this tool can be very useful when writing functional vectors for at speed testing. You can simply declare the clocks as primary inputs and use the tool to identify which paths are real and which are not. This should not be taken as an endorsement of using functional vectors for speed testing. That approach is terribly inefficient and it generally produces poor results.

.Pathdef File

The .pathdef file describes every cell in a critical path starting with a scan flop and ending with another scan flop or a primary output. It also contains information on which cells are inverting. The following shows a typical critical path using the format for FastScan

```
PATH "block_name" =
PIN /<path_to_module>/launch_flop_name/output_port + ;
PIN /<path_to_module>/gate1_name/input_port + ;
PIN /<path_to_module>/gate1_name/output_port - ;
PIN /<path_to_module>/gate2_name/input_port - ;
PIN /<path_to_module>/gate2_name/output_port - ;
PIN /<path_to_module>/gate3_name/input_port - ;
PIN /<path_to_module>/gate3_name/output_port + ;
PIN /<path_to_module>/gate4_name/input_port + ;
PIN /<path_to_module>/gate4_name/output_port + ;
PIN /<path_to_module>/gate5_name /input_port + ;
PIN /<path_to_module>/gate5_name/output_port + ;
PIN /<path_to_module>/capture_flop_name/d_input + ;

END;
```

The + and – symbols indicate the direction of the signal transition. The format for Tetramax is very similar to the FastScan format.

Size of Vector Sets

The vector sets for Transition Delay Fault testing tend to be three to five times as large as stuck at vector sets. A portion of this can be recovered since Transition Delay Fault testing catches many of the same faults that would be detected by stuck at testing. In other words the number of stuck at vectors could be reduced but that still leaves you with a larger vector set.

Also Path Delay vectors are very inefficient so the number of paths tested would probably be limited to a few dozen per chain. There are some new compression tools and methods of sorting the vectors that can help reduce the size of the vector sets.

Conclusions

There are several reasons for doing at speed testing:

1. Screening out parts that have subtle problems that cause timing failures. These problems are becoming more common at .13u.
2. Characterizing the process.
3. Debugging speed problems.
4. Speed binning parts.

Transition Delay scan testing is the most efficient way of generating at speed test patterns. It can provide good coverage and the time and effort required to do this is short and predictable.

Path Delay scan testing is useful when you want to create vectors for specific critical paths. It is less efficient than Transition Delay scan testing but the paths it checks are true critical paths. It is particularly useful for generating patterns for speed binning.

The broadside approach is the easiest to implement because it allows the scan clocks and the Scan Enable signal to run at slow speeds for muxed flop scan designs. It also works well with LSSD designs. If the capture clocks are driven by the tester then you must be sure that the tester can operate at the required frequency. If the capture clocks are generated internally the required clock control needs to be designed into the clock generation logic and you need an input that triggers the generation of two or more capture clocks.