feature

Published in **February/March 2006** issue of Chip Design Magazine

Easily Implement PLL Clock Switching for At-Speed Test

By taking advantage of pattern-generation features, a simple logic design can utilize phase-locked-loop clocks for accurate at-speed testing.

By Ron Press & Jeff Boyer

Nanometer technologies enable higher-frequency designs and more integration. Yet they also introduce a higher population of timing-related defects. At geometries of 130-nm and below, the timing-defect population has grown to the point that at-speed testing has become a requirement at many companies.[1,2] Utilizing scan to perform the at-speed testing is a proven method to detect timing defects. In fact, at-speed scan testing has replaced at-speed functional testing for the same reasons that stuck-at scan testing replaced functional testing.

Scan-test strategies are based on very simple structured-design approaches. Scan implementation is very straightforward. Automated tools support both scan insertion and pattern generation. As a result, specific circuit knowledge isn't necessary. In addition, automatic test-pattern generation (ATPG) produces high test coverage. Conversely, functional-pattern usage is being reduced in the industry because the results are less predictable. They also require high-performance test equipment and are costly to develop and fault grade.

Scan architecture segments a circuit's complexity into small combinational logic blocks between sequential elements. Using scan, the sequential elements become control and observe points. The analysis for ATPG is thus greatly simplified. Each scan pattern is independent because the circuit's sequential nature is removed during test. Troubleshooting test-pattern mismatches or diagnosing failures therefore becomes a simple process. In contrast, every cycle of a functional test-pattern set could be dependent on any other cycle. Troubleshooting functional-pattern problems is extremely complicated.

Timing defects occur between a sequential launch and capture point. Consequently, they fit nicely into scan-test methodologies because the sequential launch and capture elements are controllable and observable as scan cells. An at-speed scan pattern loads an initial value at a launch point. It also prepares to launch the opposite value from the launch point. Usually, it performs this launch within one scan-chain load.

Figure 1 depicts a basic example of an at-speed scan pattern. In this example, the at-speed test is performed on the logic between scan cells B and C. An initial value of 0 is loaded into scan cell B. During the same load, a 1 is loaded into scan cell A. The value at scan cell A results in a 1 at the functional D input to scan cell B.

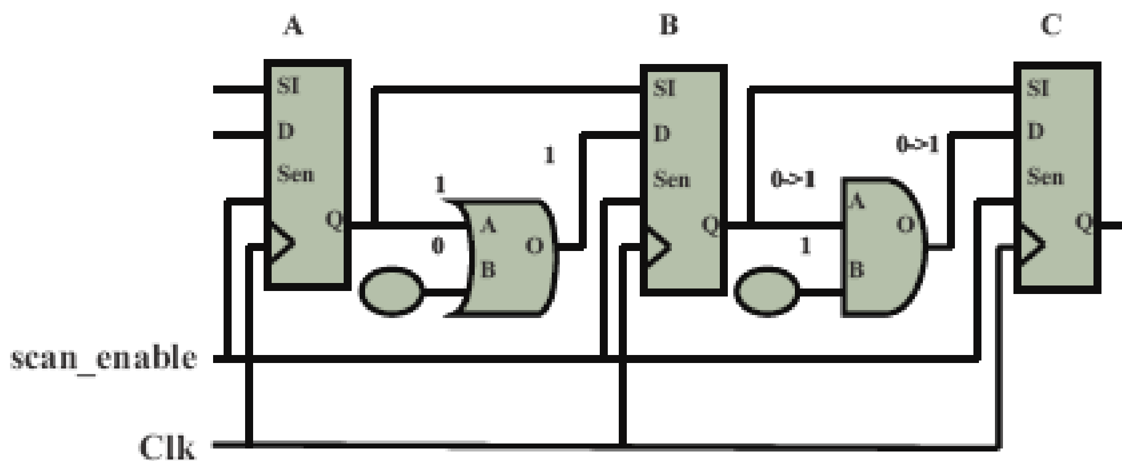

Figure 1: Here is a simple example of a broadside at-speed scan pattern. A 0 to 1, at-speed test is performed on the logic between scan cell B and C.

After the scan cells are loaded, the circuit is placed into functional mode (scan_enable = 0). The first functional clock pulse will cause cell B to capture the 1 at its D input. The 0-to-1 transition will propagate toward cell C. A second clock pulse will capture the value at cell C.

Next, the captured values are unloaded and shifted out for verification. If a 1 was captured into cell C, the transition propagated within the desired time between the launch and capture clocks. The circuit is therefore functional. If a 0 was captured, a timing defect exists. The 0-to-1 transition obviously didn't propagate to cell C within the required time.

The accuracy of the at-speed scan-pattern application is only dependent on the accuracy of the launch and capture clocks. Scan-chain loads can be performed at an entirely different frequency. If a stuck-at fault exists in any of the logic being tested by the at-speed scan test, the test will fail. An at-speed scan test will thus result in a thorough stuck-at coverage test.[4]

COMPLICATIONS RELATED TO AT-SPEED SCAN TESTING
ATPG tools can be used to automatically create at-speed scan tests. But several special considerations can complicate the ATPG process or test accuracy. Much is related to the launch and capture clocking. Circuits with multiple clock domains may not want an ATPG tool to arbitrarily select launch and capture clocks. In fact, some domain-to-domain interactions may not be specified. They should be avoided during at-speed test.

The effectiveness of the at-speed test is directly related to the accuracy of the launch and capture clocking. Testers can be used to provide these clocks. However, tester frequency and accuracy must be considered. For example:

- A desire exists to lower the "cost-of-test"/"cost-of-manufacturing" by using older, existing (completely amortized) testers with lower-limited clock frequencies. In addition, costs can be reduced by using lower-cost testers.
- At frequencies above 100 MHz, it becomes more difficult to deal with the

interactions between tester skew, pad delays, and internal-clock insertion delays. Many circuits utilize internally generated clocks from phase-locked loops (PLLs). If these internally generated clocks could be controlled by the ATPG tool and subsequently applied on the tester, this would:

- Simplify the tester requirements and dependencies by eliminating the tester-skew/pad-delay issues
- Have a test that is more "true-to-life" by exercising the same clock paths and timing as is done for normal functional operation

Other considerations during at-speed scan testing are the sequential depth of the design and how to handle multi-cycle and false paths.

EFFECTIVE AT-SPEED SCAN TESTING

Today, two fault models are commonly used for at-speed scan testing: the path-delay and transition-fault models. The patterns for both tests operate essentially the same. The difference is based on how and where the faults are targeted.

The path-delay fault model targets specific paths that are known to have very little timing slack. Each path usually begins at a scan cell, includes combinational logic, and ends at a scan cell. Because of the systematic variations in timing for a chip or wafer, the paths are potential locations of timing defects. Often, these tests are used for speed binning. Only the most critical set of paths is targeted. In some cases, this set could be 20 to 100 paths. In some very-high-speed circuit designs, however, many paths have little slack. This fault model provides little value in defect detection beyond the transition-fault model.

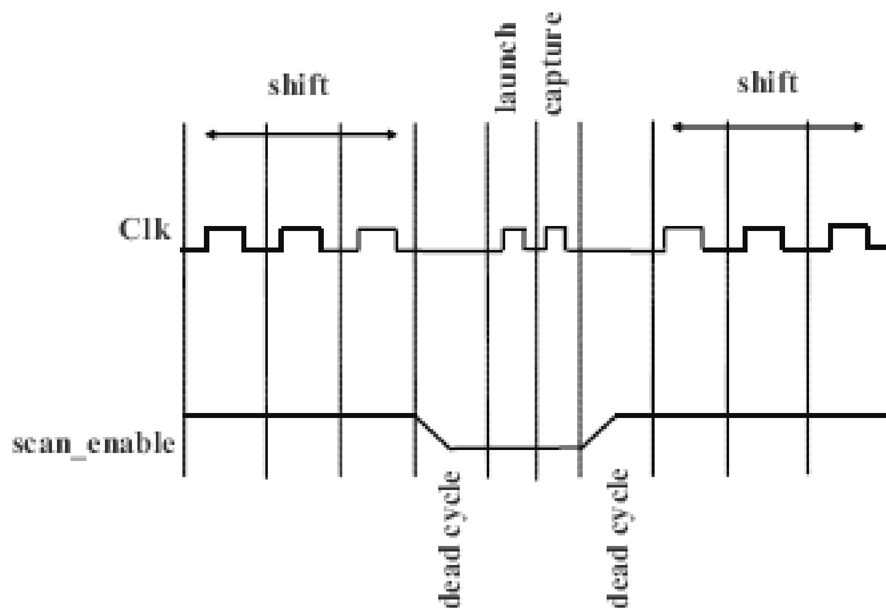

Figure 2: In this broadside at-speed scan pattern, the scan chain is loaded at a slow frequency. Scan enable is de-asserted and the functional clocks are pulsed.

The transition-fault model targets a gross delay defect at every gate terminal. It is the primary test used to detect timing defects that could exist anywhere in the

device. The ATPG tools will target a transition fault and then automatically find a path where the transition from the launch point will propagate through the targeted fault location. Clock application is usually applied by doing the following: loading the scan chains at a slow frequency, switching to functional mode, and pulsing a launch and capture clock with the desired timing. This sequence is referred to as a broadside or launch-from-capture pattern. The timing of the scan-chain load is completely independent of the at-speed launch and capture cycles. An extra "dead cycle" can be used if additional time is desired to allow the scan-enable signal to settle prior to applying the launch-clock pulse.

Another method to apply the at-speed scan pattern also is possible. Called "launch-off-shift" (LOS), it is usually avoided by most companies. In a "launch-off-shift" pattern, the transition occurs because the last shift in the load-scan chains is the launch cycle. The scan-enable must turn off very quickly. One clock is pulsed in functional mode to capture the response at the end of a path.

These types of patterns are very easy for ATPG tools to create. Unfortunately, they require that the scan-enable signal be routed as a global clock or pipelined throughout the circuit. This requirement is both unnecessary and unreasonable. In addition, many faults cannot be detected by using the dependency in a shift as a launch. In fact, these patterns can detect very little that isn't possible to detect with broadside patterns. But they will detect many non-functional faults, such as scan-path faults.1 As a result, testing non-functional faults and requiring scan-enable to operate as a clock will result in unnecessary yield loss.

At-speed scan testing must focus on realistic defects. All circuits will include many non-functional or non-testable fault sites. Transition patterns shouldn't target faults within slow-speed scan paths. Furthermore, high-speed scan patterns will rely on accurate clocking. But the input and output signals often cannot operate at the clocking frequencies. The testers may not support high-speed IO and/or the device pads may not be designed to support internal high frequencies. As a result, at-speed scan testing will often need to ensure that the primary inputs are stable prior to the at-speed launch cycle. Outputs are not measured.

ACCURATE CLOCK PULSES

Utilizing PLLs for at-speed scan testing is clearly the best approach to applying accurate launch and capture clocks--especially for devices that operate at high frequencies. ATPG tools and clock switch designs evolved to support the use of PLLs during at-speed scan testing. PLL clock switches had to implement some controls that enable specific clock sequences to be defined and pulsed during scan tests.[3] The ATPG tools could then understand the behavior of the PLL clock switching based on a procedure that describes the internal-clock pulses. Those pulses result from a sequence of primary input cycles. These named capture procedures provide enough information for ATPG processes to select the appropriate procedure based on targeted faults and the corresponding internal clock to detect them

Named capture procedures enabled PLL clocking for at-speed tests. Some of the clock-switch-logic designs started getting complicated, however, because some devices required hundreds of control bits to be initialized. One way to load a large number of control bits is to perform many cycles at the primary inputs. Although this approach is effective, it sometimes requires hundreds of additional cycles for every pattern. Another approach would be to add many additional primary inputs to directly control these bits. Obviously, this approach isn't acceptable. Several

designers started using the boundary-scan TAP port to load PLL control registers. No additional signals were required, but each pattern required many cycles through the TAP.

To simplify PLL clocking control, the ability to control internal registers was developed as part of a named capture procedure.[4] Condition statements are used to specify certain register values. To be valid, those values must be initialized for a named capture procedure. The state of several hundred bits within control registers can therefore be defined for each named capture procedure.

If these registers are scannable and part of the scan chain, it's very easy for ATPG tools to automatically initialize them. For example, consider a circuit with several clock switches--each with several control signals. This could add up to a significant number of bits that must be controlled prior to loading the scan chains for each pattern. Or they must be controlled using extra cycles during the functional-mode capture cycles. Manually working out how to specify the correct cycles in order to control these signals could be very tedious. With the use of condition statements, however, the registers that drive these signals just need to be scannable and defined with condition statements. The ATPG tools will automatically take care of loading the correct bits during shift without the need for extra cycles.

CLOCK-SWITCHING DESIGN

This section presents a simple clock-switching design that facilitates PLL clocking during at-speed test. It doesn't add any additional timing constraints to the design. In addition, it has a control mechanism that works directly with ATPG. There are no special requirements for ATE to support patterns using this strategy. This circuit can be used for at-speed tests using PLL clocks or stuck-at tests. Even though the design and examples are based on mux-D scan design, they're easily adapted to LSSD scan.

Figure 3 shows a high-level diagram of the PLL clocks and scan-clock switches. The scan_enable is asserted to put the circuit into scan shift mode when loading and unloading the scan chains. Clk_test is used as the shift clock to load/unload scan chains. Scan_enable is de-asserted in functional mode, which allows the PLL functional clocks to propagate into the device. The scan-clock switches are architected to allow exactly two pulses to propagate during at-speed scan testing for full-scan designs. They are easily modified to allow additional pulses for partial-scan designs.

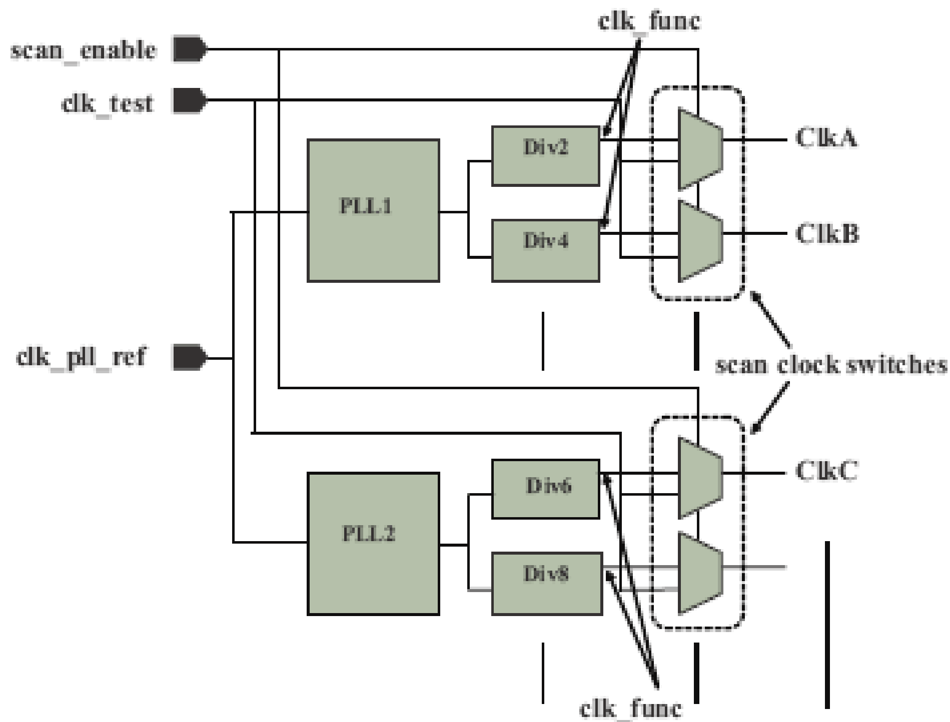

Figure 3: A high-level diagram of PLL and clock switches is shown here.

SCAN-CLOCK SWITCHES

The scan-clock switches shown in Figure 3 are more than simple muxes. Figure 4 shows the logic within each switch. The last mux in the diagram is the output-clock mux. Typically, it already exists in a design to allow scan operation. All of the scan-clock-switch logic is outside of the functional-clock tree path (i.e., clk_func). As a result, this implementation has no impact on the PLL functional clocks or the performance of the functional design. The IO of the scan-clock-switch design is as follows:

- Scan_speed_testmode: static signal used to indicate that an at-speed test is to be performed. It is mutually exclusive with scan_stuck_testmode.
- Scan_stuck_testmode: static signal used to indicate that a stuck-at test is to be performed. The primary input clock, clk_test, is selected from the switch mux when scan_stuck_testmode is active.
- Scan_enable: The primary input test clock, clk_test, propagates to ClkA when it is active.
- Scan_capture_enable: activated when the switch is activated to allow a capture clock to propagate. A capture clock cannot propagate to clk_scan when it is inactive.

When a device only has one or two scan-clock switches, the scan_capture_enable signal can be controlled directly from a primary input. If multiple scan-clock switches are used, however, this signal is controlled by internal DFFs. The DFFs are placed within scan chains and automatically loaded during ATPG using condition

statements within named capture procedures. It is then very easy to select which clock is to be active during capture cycles. With this very simple but powerful mechanism, internal capture clocks can be isolated from one another on a pattern-by-pattern basis within a single ATPG run for any fault type.

The dual-stage synchronizers (or closely placed back-to-back non-scan DFFs) are initialized to 0s and are normally driving 0s. When the scan_speed_testmode is active and scan _enable is de-asserted, a 1 is captured into the "trigger" DFF preceding the top synchronizer. This trigger DFF is only necessary in Mux-D scan designs and not LSSD designs (although its inclusion will not hurt). The purpose here is to delay the generation of the at-speed capture pulses so that the transitions on scan_enable have enough time to propagate to all scan DFFs prior to their application.

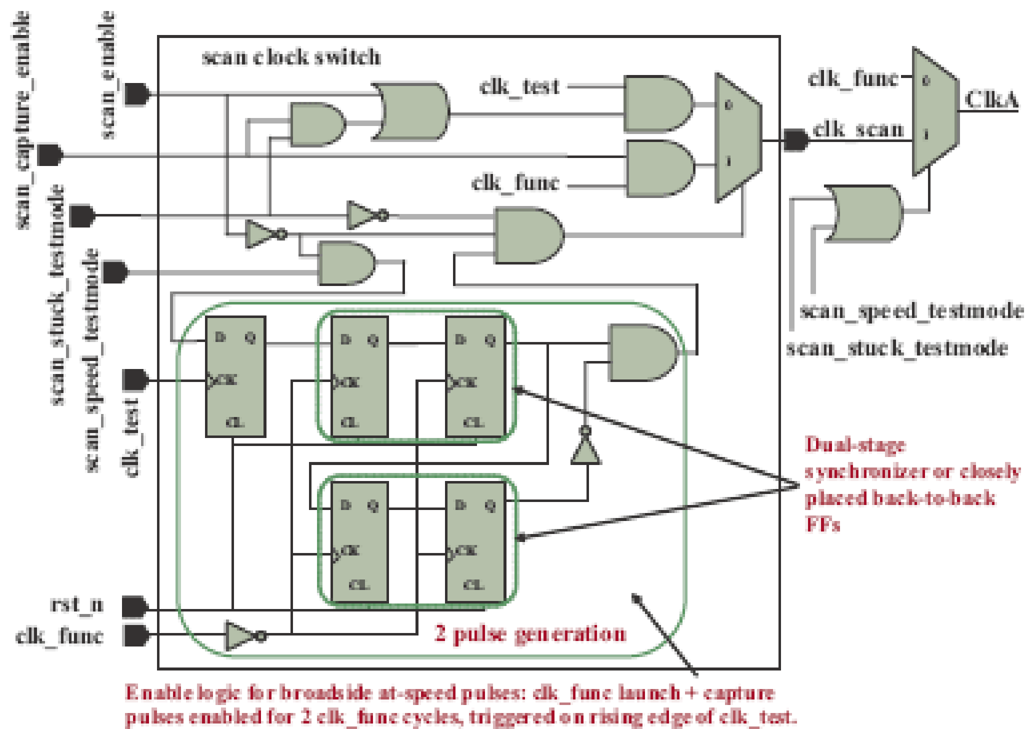

Figure 4: This graphic depicts a scan-clock-switch circuit.

The pair of synchronizers will produce a 1 at the output of the AND gate for exactly two cycles between two falling edges of clk_func. The paired synchronizer design will always produce a 1 for two cycles of clk_func as long as the scan_speed_testmode is active and scan_ enable is inactive for at least two cycles of clk_func. The timing of the functional PLL clock, clk_func, is completely independent of the primary input test signals. No special synchronization at the tester is necessary.

The static 1 that is produced for two cycles from the paired synchronizers feeds into the scan-clock-switch mux. It allows exactly two PLL clock pulses (clk_func) to propagate out the switch. This type of design is glitch-free because the PLL functional clock is used to gate itself, but remains 180 degrees out of phase.

SCAN-CLOCK-SWITCH VARIATIONS

The basic architecture of the scan-clock switch described in the previous section can be used as a template for expanded clock-switch capabilities. Any number of pulses can be defined. Figure 5 shows a simple variation to produce more than two PLL functional-clock pulses.

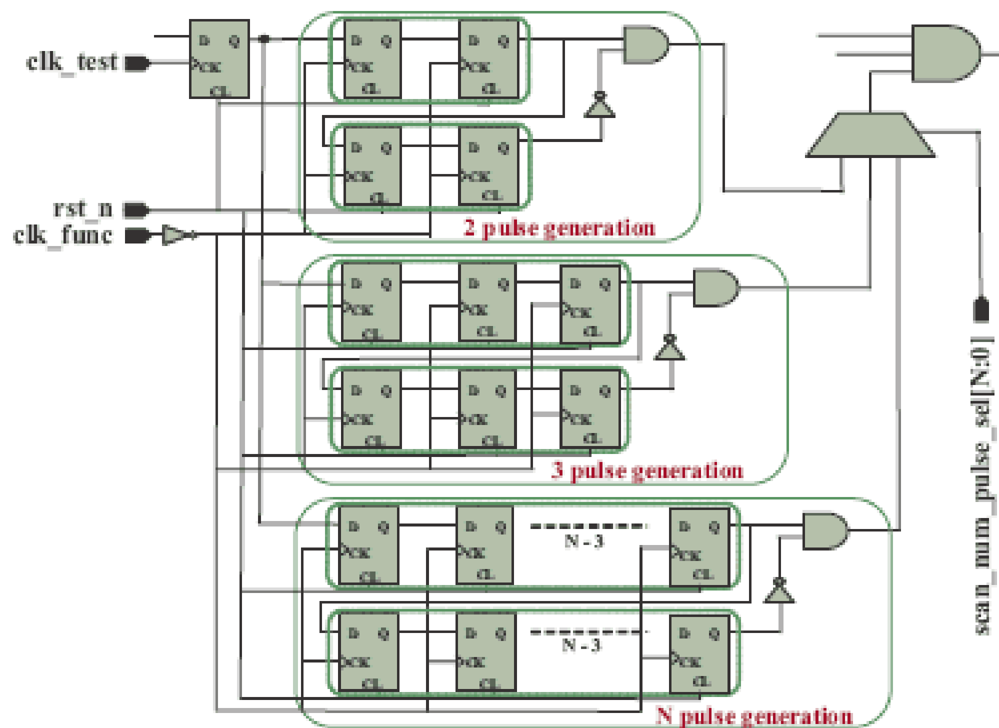

Figure 5: This scan-clock-switch circuit has an option to pulse more than two PLL functional clocks.

The scan-clock-switch design also can be used to perform at-speed scan tests when multiple clock domains exist. Figure 6 shows the logic to support multiple clock domains. The various scan_capture_enables produced by the logic of Figure 6 are connected to the corresponding scan_capture_enable input ports on the scan-clock switches. Again, each is specified as a condition in a named capture procedure. Only the non-interacting clock domains are allowed to pulse simultaneously for a given pattern.

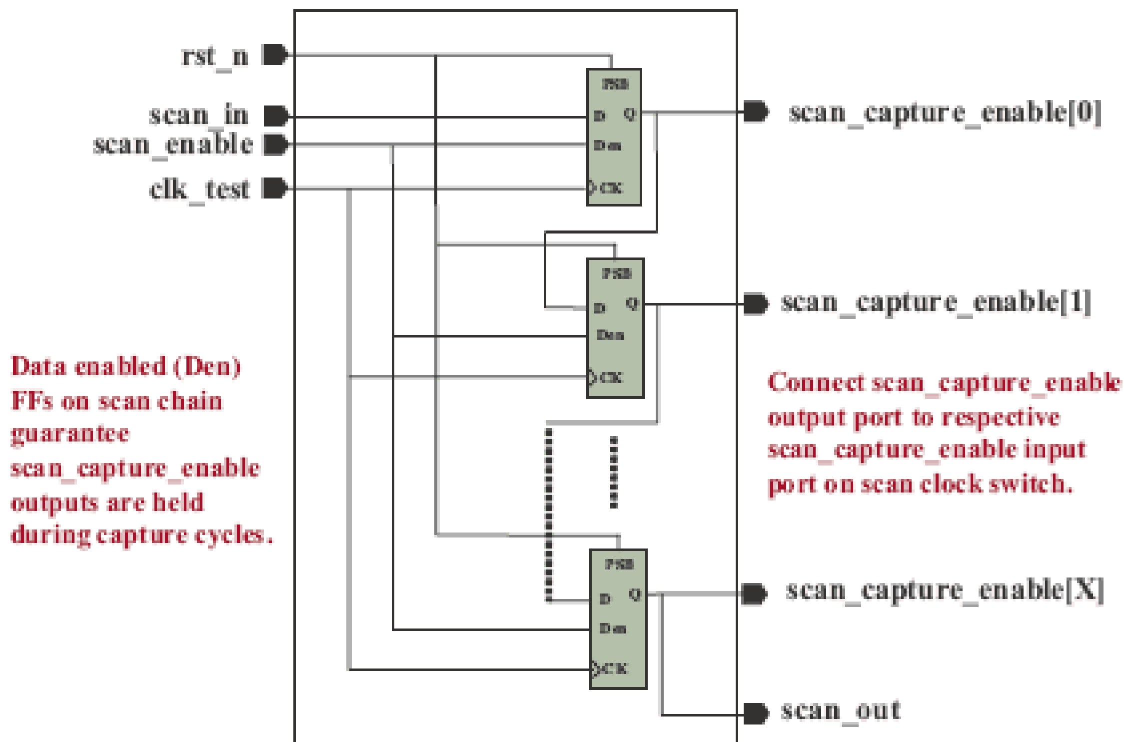

Figure 6: The scan-capture clock enables control logic for multiple clock-domain designs.

An example waveform for a device with several PLL functional clocks is shown in Figure 7. The scan_enable signal isn't timing critical. This waveform shows that scan_enable is de-asserted for long enough for all clock pulses to propagate. Tests aren't performed on logic where a clock from one domain is at the launch point and a different domain is at the capture point. Although such tests are possible, care must be taken to ensure that the timing between the clock domains is well understood.

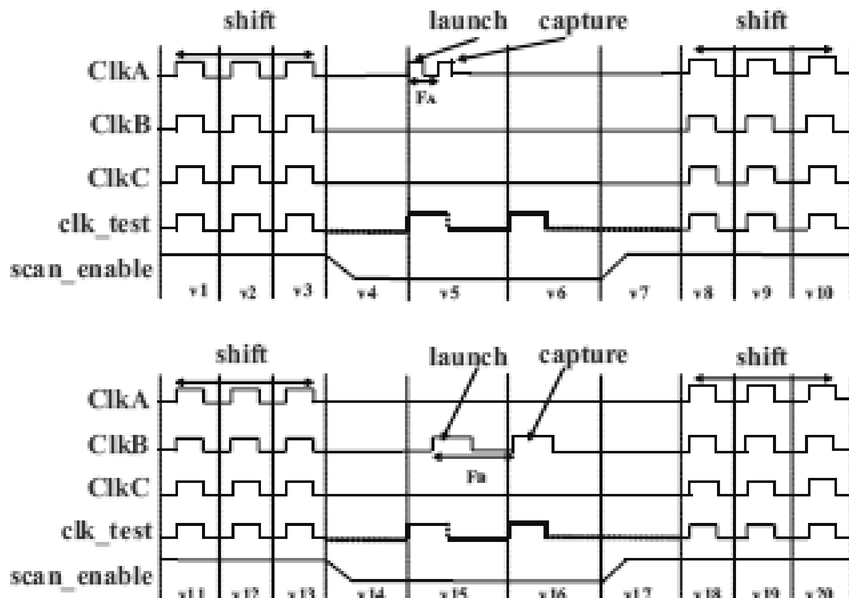

Figure 7: Figure 7. Here are the launch and capture pulses for multiple clock domains.

EXPERIMENTAL RESULTS/CAVEATS

The circuitry described above is optimized for completely asynchronous clock domains. The "inter-domain" logic is not tested at-speed. In other words, the case in which ClkA is the launch and ClkB is the capture clock isn't necessary.

The presented clock-switching technique has been implemented in all PXA25x, PXA26x, and PXA27x Xscale processors. These nearly full scan designs contain >24 internal-clock domains--most of which are asynchronous to one another. Roughly a quarter of the domains are high speed (100-to-400-MHz capture, 13-MHz shift). For those domains, targeted at-speed transition scan patterns are applied using the broadside clock-switching technique presented herein.

The PXA25x and PXA26x Xscale processors have seen a resulting ~300% decrease in DPM. Data gathering is ongoing for the PXA27x series of processors. Ultimately, the clock-switching technique has enabled outgoing DPM quality to be below DPM goals. It has subsequently been adopted for future-generation products.

In summary, accurate at-speed tests are possible without significantly complex logic. The clock-switch design presented herein is a simple method to allow clock pulsing without compromising the clock tree. Named capture procedures provide a simple method to model the PLL and clock switching for automatic controlling and handling during ATPG. Results have shown that at-speed scan tests using these clock switches resulted in dramatic improvements in DPM.

The at-speed scan tests using PLLs can be expanded to other scan-test techniques. Macro testing--a technique that converts a desired pattern sequence for an instance into scan-chain loads--can use the same PLL clock switching and named capture procedures.5 This capability enables the at-speed application of specific test sequences, such as march algorithms for many high-performance memories in parallel, without adding any additional logic or controls around the memory. In addition, compression techniques like embedded deterministic test can use the same PLL controls and named capture procedures.[6]