Scan

This is a sample of the first 15 pages of the Scan chapter.

Note: The book is NOT Pinted in color.

Objectives:

This section provides:

- An overview of Scan
- An introduction to Test Sequences and Test Conditions
- DFT Terminology

Manufacturing Test and DFT

As mentioned previously, once the circuit design has been proven and characterized, the circuit moves into volume manufacturing. At that time emphasis shifts from design concerns to manufacturing concerns. Screening for manufacturing defects becomes the main focus of test. Structural Vectors become important in this phase to ensure high fault coverage and quality. These tests are sometimes called Defected-Oriented tests because they are designed to detect specific defects. Structural vectors are often based upon a gate level circuit model, and their goal is to verify correct functionality of the gate structures and wires, and to detect defects, or potential reliability risks.

For structural test to be effective the circuit must be controllable and observable. This is accomplished through DFT design methodology, one common approach involves adding Scan circuitry to the chip design.

The terms Structural Test and Scan Test are sometimes used interchangeably, although Structural Test may include additional design and test techniques. Scan vectors are often designed to detect Stuck-At defects, but they may also be oriented toward exposing other types of defects. A combination of Scan vectors and functional vectors are generally used in manufacturing test. There are different methodologies associated with DFT. Next, we will look a few examples.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

Ad-Hoc and Structured DFT

Ad-Hoc techniques (non-structured DFT) are device specific. They are created for a one-time use, or special purpose, such as adding control and observation test points to a circuit. As an example, logic can be added to shorten the test of a long counter. A 30 bit counter will take 2^{30} clock cycles to be fully tested. At a frequency of 100 MHz, this would take over 10 seconds of ATE test time. Adding test logic to partition and test the 30 bit counter as three 10 bit counters would provide a test time of 30 us (10 us per counter). Ad-Hoc techniques are sometimes referred to as common sense additions to a specific circuit design.

Structured DFT requires the acceptance of a standardized design and test methodology, and requires the insertion of additional test circuitry. This enables effective use of EDA tools for automated test logic insertion and vector generation. Structured DFT improves quality, lowers the cost of test, and provides improved methods of debugging/trouble-shooting. In contrast to Ad-Hoc testability techniques, structured DFT usually applies to a broad range of circuit designs.

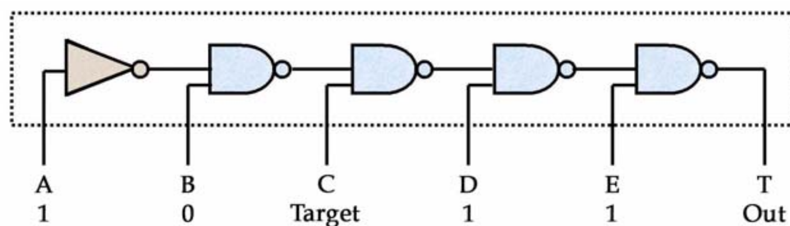

Figure 2-1 Old Style Structured DFT

In the past some companies used a simple form of structured DFT, as shown in Figure 2-1. This example shows a *gate tree* that consists of a series of linked together gates, each having a control input. This circuit may have been included on every circuit design produced by a semiconductor manufacturer as a process monitor to evaluate input thresholds and speed. By including this on every die a great deal of data could be gathered regarding fabrication and circuit performance. Some companies still use this form of DFT.

Historically, structured DFT techniques were rejected due to the negative impact on die size and performance (mainly speed). Structured DFT is now quite common due to the continued increase in design gate count making the task of generating high fault coverage functional vectors overwhelming. Also, advances in semiconductor process technology such as increased metal layers, and deep sub-micron gate widths have reduced the die area and lessened the performance penalty of adding DFT circuitry.

Additionally, integrated circuit users have increased their requirements for high test coverage, which has also helped speed the acceptance and implementation of DFT.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

Types of Structured DFT

There are three common structured DFT methods currently used in chip designs. These are:

- Scan
- Boundary Scan (JTAG)
- Built In Self Test (BIST)

Scan Design

Scan is a design technique that converts sequential logic to combinational logic. Scan design effectively converts a sequential logic (clocked) design to a combinational logic (non-clocked) design allowing ATPG tools to produce high fault coverage test vectors efficiently. It enables a complex circuit to be partitioned into individual combinational logic functions, each logic function can be tested separately, and all sequential logic elements can be directly controlled.

Sequential logic circuits can require extensive conditioning which is both time consuming and expensive in terms of simulation and test time. To implement a *Full Scan Design*, each flip-flop within the design is converted to a specialized Scan flop. Scan flops are easily controlled, and become additional inputs and outputs, similar to adding internal test points. A circuit is considered to be a *Partial Scan Design* when Scan Flip-Flops are used, but not to the extent that allows full control of the entire circuit.

Scan connects all state elements (flip-flops) of a design into one, or multiple shift registers which are referred to as *Scan Chains*. Scan flops may also be added to any internal circuit paths that contain feedback signals in order to allow direct control of the fed back signal path. Feedback paths are sometimes referred to as *Secondary State Variables*.

Once the Scan chain is configured, it allows both controllability and observeability over the entire circuit, assuming a Full Scan design. For ATPG purposes, the Scan flip-flops are considered virtual inputs and outputs.

During Scan test, input data is shifted serially into the device and then applied in parallel to all inputs of the target logic blocks. Outputs of the logic blocks that are not directly accessible, via primary outputs, are captured in parallel into the Scan chains, then shifted out serially for evaluation. This method of testing greatly reduces the number of data cycles required to condition and test the logic when compared with using only the primary inputs and outputs. It also improves the diagnostic capabilities of the test.

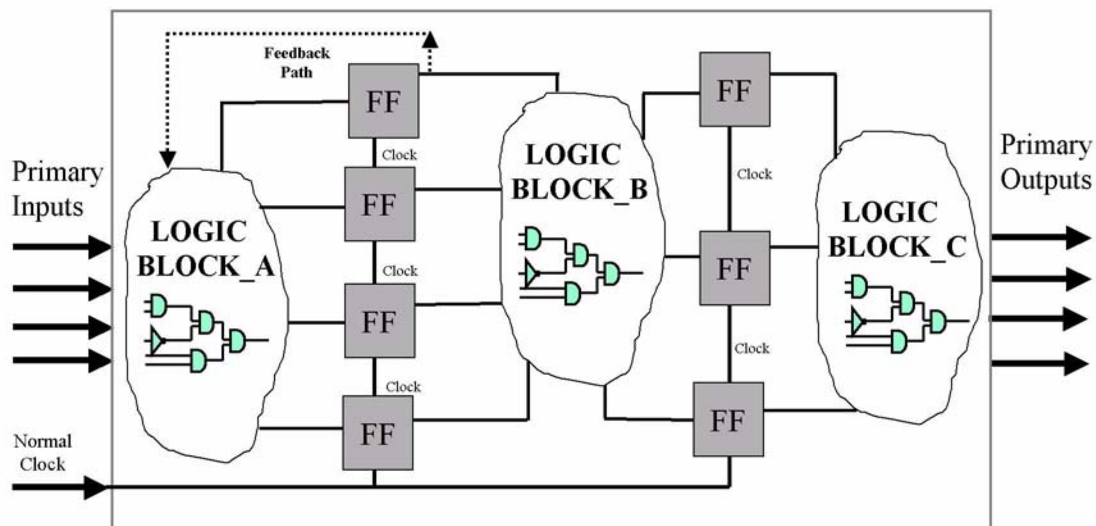

Figure 2-2 Non-Scan Circuit

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

Non-Scan Logic

Figure 2-2 illustrates a simple circuit containing several blocks of logic, each with its own function, and the associated sequential logic shown as “FF” (flip-flops). Ideally each logic block will be exhaustively tested for faults.

From the diagram you can see that the inputs to Block A can be directly controlled via the primary inputs, but the output response must propagate through Blocks B, and C before being observed on the primary outputs. Logic Block B can be neither accessed nor observed directly. Block C’s outputs can be observed, but the inputs cannot be directly accessed. This design makes it difficult to achieve high fault coverage with a limited number of clock cycles.

If any block within the circuit fails, the ability to diagnose or localize the problem is very limited. This circuit would be much easier to test, and troubleshoot, if each logic block could be independently controlled and observed. Inserting Scan logic into this circuit will overcome these problems. Inserting Scan logic involves several modifications to the circuit. First, let’s look at how the flip-flop design is modified to become Scan enabled.

Scan flip-flops

The left panel of Figure 2-3 illustrates a standard D flip-flop. This must be modified to become a Scan flip-flop (right panel). When Scan is added to a circuit design all normal flip-flops are replaced with Scan flops. The Scan flips-flops are “D Type” flops with a multiplexer (mux) on the input. A Scan Enable signal is added to the circuit to control Scan Mode testing.

When Scan Enable is held high the Mux is in Scan Mode, and data is clocked serially from the Scan Data In through the flip-flop chain, the normal data input path is inhibited via the mux. When Scan Enable is held low, the normal data input passes through the mux, and the Scan input is inhibited. The output of the last Scan flip-flop in the chain is connected to the Scan Data Output pin.

The panel on the right shows a typical Scan flop, but there are a number of different variations of this design. Flops may have preset, and clear inputs, and they may have individual clocks for normal and test modes.

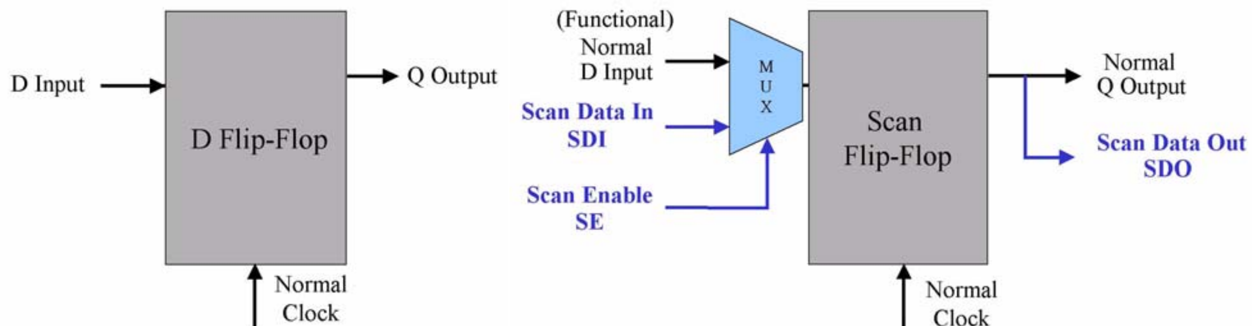

Figure 2-3 Normal verses Scan D Flip-Flop

There are two common types of Scan design techniques used, the most common technique is called Muxscan, the type of Scan flop shown on the right in Figure 2-3. Another technique still in use today is called *Level Sensitive Scan Design or LSSD.* LSSD was invented by IBM. LSSD is specifically aimed at reducing the dependency of system operation on AC parameters such as clock edge rise or fall times, which are difficult to simulate in the design environment, and difficult to monitor in a manufacturing environment.

The LSSD technique provides direct control of two clocks. One clock controls the acceptance of data into the flip-flops while the second clock controls their outputs. With direct clock control the timing of the circuit can be controlled to avoid potential race conditions resulting from the primary or secondary inputs propagating through the combinational circuits. LSSD is more clock skew tolerant, but has a higher die overhead penalty.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

Scan Logic

Figure 2-4 shows the same circuit as Figure 2-2 on page 2-3 only with Scan logic inserted. Notice that each D flop has been modified to a Scan flop, then linked together to form a chain. Three pins have been added: Serial Data In (SDI) connected to the mux input of the first flop in the chain, Serial Data Out (SDO) connected to the output of the last flop in the chain, and Scan Enable (SE) connected to the mux control of each flop.

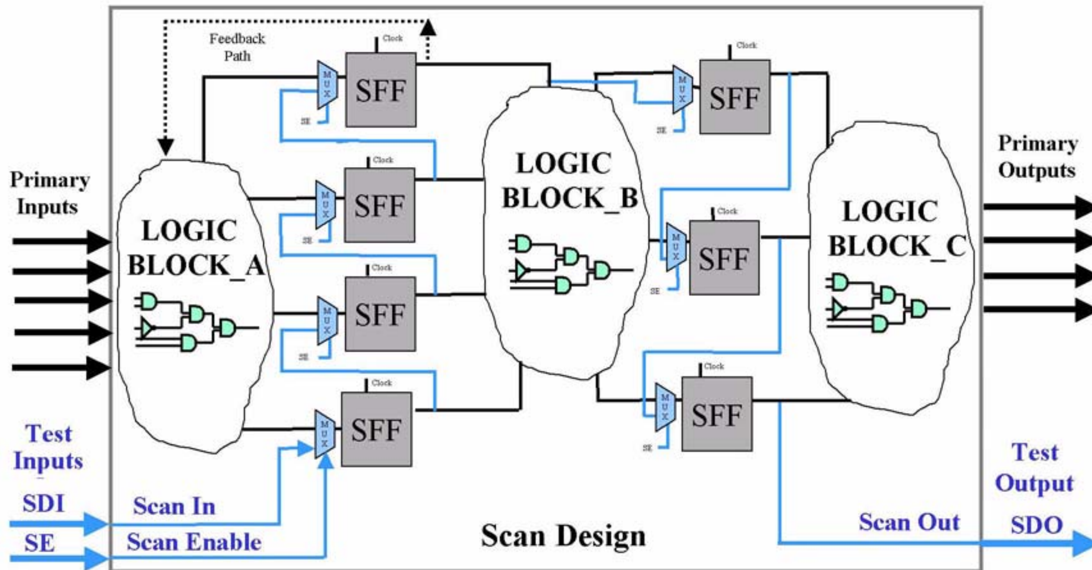

Figure 2-4 Circuit with Scan

This circuit can now be tested using the Scan chain. Logic Block A can be tested by applying data to the primary inputs then capturing the results into the first 4 flops (closest to SDI), this is accomplished in normal functional mode (SE set low). Once the results are captured the SE pin will be set high (Scan mode), and the captured data shifted and observed at the SDO pin. During this test the four Scan flops are acting as observation points (outputs), the remaining Scan flops are not used.

To test Logic Block B, input conditioning data can be shifted into the first 4 flops (SE high). SE is then set low, and the chip goes into normal operational mode, the data propagates through the combinational logic and is captured into the Scan flops located at the output of Block B on the next clock. Once the data is captured, SE is set high and the data is shifted (clocked) through the Scan chain and observed at SDO. During this test the first four flops act as conditioning inputs, the next three flops act as observation points (outputs).

It is possible to test all logic blocks at the same time. This test begins by applying conditioning data to the primary inputs for Block A. SE is then set high and conditioning data is shifted into the entire Scan chain. SE is set low and the test results are captured and primary outputs evaluated. SE is set high and the results shifted out. The first four flops act as both inputs for conditioning Block B, and as outputs for observing the results of Block A. The remaining flops act as inputs to condition Block C.

Converting normal flops to Scan flops greatly increases the effective number of inputs and outputs. Replacing the normal flip-flops with Scan flops may have a negative impact on the operating speed of the circuit. Adding Scan logic also increases die size and pin count, it reduces yield and complicates the routing of interconnect wiring. Like many issues in test, it is a cost/benefit trade off.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

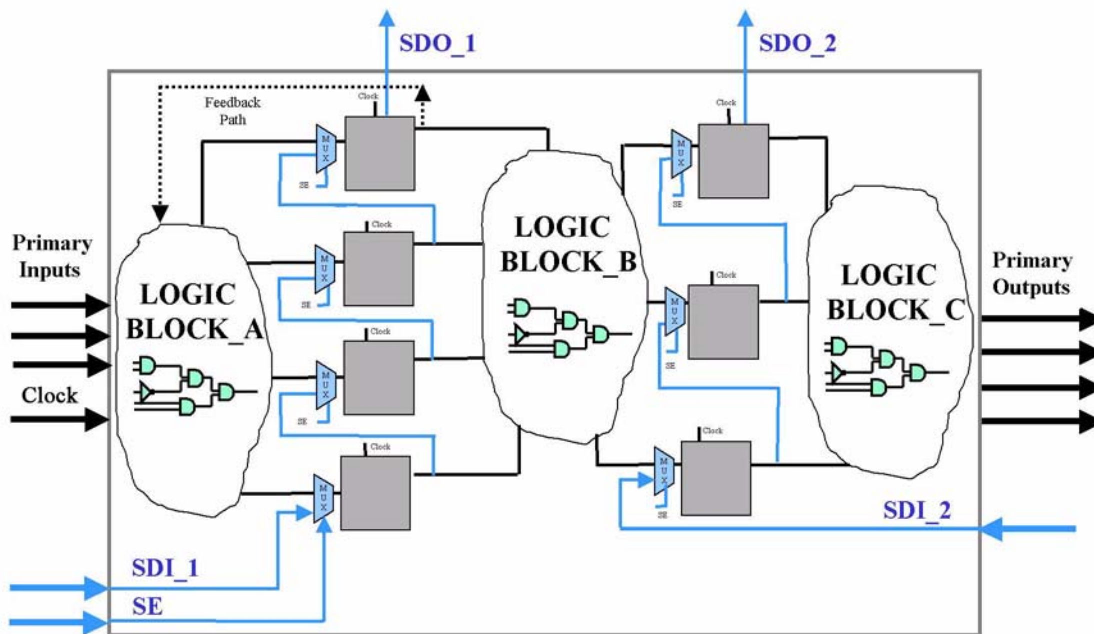

Figure 2-5 Multiple Scan Chains

Multiple Scan Chains

In early Scan designs three dedicated pins were added to the circuit. These were: Scan Data In (SDI), Scan Enable (SE), and Scan Data Out (SDO). The flip-flops worked in two modes, normal functional mode involving parallel data loads, or in Scan mode, configuring the flip-flops into a shift register (Scan chain register). Most all test data was transferred on only 2 pins: SDI and SDO

Adding Scan to complex devices often results in very long Scan chains. The longer the chain the longer it takes to shift the data in and out of the device. In order to reduce test times it is common to break a long Scan chain into multiple smaller chains. These chains can then be loaded and tested in parallel. This technique can dramatically reduce test time and test costs.

Scan chains within a chip can be designed in several ways: there can be one long (global) chain, containing many Scan cells. There can be parallel chains, that are short containing a reduced number of cells per chain. Or there can be multiple chains to service different clock domains such as the core and peripheral areas. Scan pins no longer need to be dedicated test pins, more on this later in this chapter.

Early ATE systems had only Parallel vector memory which is inefficient at handling Scan vectors. Scan vectors have hundreds of cycles with only a few pins changing data, unlike parallel vectors where many pins change within each cycle. As Scan became more common, a separate Scan Vector Memory was added to handle the large amounts of Scan drive and expect data. On some ATE systems this data was routed to dedicated tester channels for Scan In, and Scan Out device pins. On current ATE systems most any digital pin can be used for Scan IO and most test systems support multiple Scan paths.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

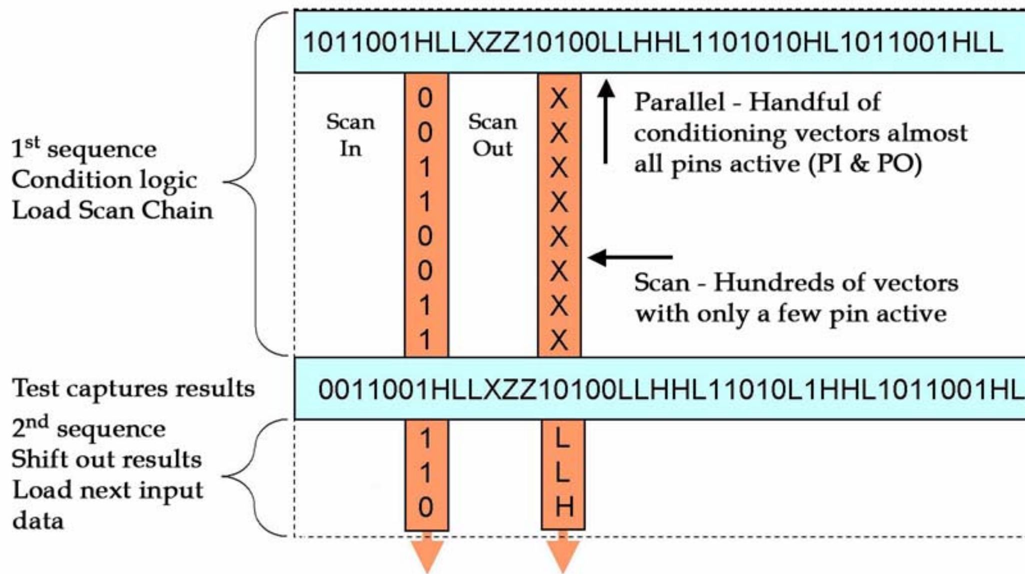

Figure 2-6 Scan Vector Sequence

Scan Test Sequence

As shown in Figure 2-6, a typical Scan test begins with a small number of parallel vectors applied to the device to condition the logic in preparation for the actual Scan test. These vectors are stored in normal parallel vector memory and applied to the primary I/Os. During this sequence SE is held low, and the circuit is in normal operational mode.

Once the device is properly conditioned, SE is set high and the Scan input vectors are applied to SDI. Input data are shifted into the Scan chain until the chain is full. On the last Scan clock the data held in the Scan chain is properly placed and applied to the circuit's combinational logic, this begins the test.

SE is then set low, returning the circuit to normal functional mode. Time is allowed for the data to propagate through the combinational logic, and one clock is then applied. This clock captures the output response from the combinational logic into the Scan chain (observation points), and at this time the primary outputs are evaluated. The results of testing the combinational logic are now stored in the Scan chain and must to shifted out of the device for evaluation.

SE is set high, returning the circuit to Scan mode, and the test results are shift out of the device and observed at the SDO pin. While the captured results are being shifted out, input data for the next test are shifted in. This sequence then repeats for each additional Scan vector sequence.

Note: The terms *Vector* and *Pattern* are used in several different ways. The term *Vector* generally refers to data for a single tester cycle, stored in a single location in ATE parallel vector memory. The term *Scan Vector* refers to a sequence of cycles consisting of Scan in, a Parallel operation (test), and Scan out. This sequence represents an entire Scan test. A *Pattern* in STIL (Standard Tester Interface Language–pronounced STYLE) would have many logical Scan vectors and would contain a complete test sequence.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

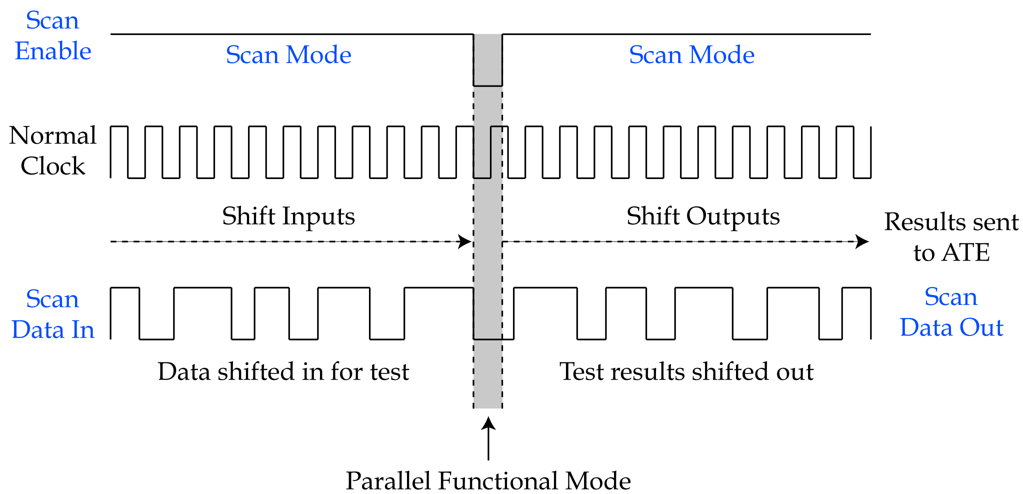

Figure 2-7 Complete Scan Test Sequence

Scan Timing

Figure 2-7 shows the basic timing sequence of a Scan test. First, Scan mode is enabled via the Scan Enable signal and serial input data is shifted into the device to condition the circuit under test. The frequency used to shift data in and out is often slower than the actual device operational frequency.

Next, Scan is disabled and the input conditioning data are allowed time to propagate through the logic under test. A long capture delay time may be referred to as *DC Scan* which verifies correct operation of the logic structure.The capture delay time may also be adjusted to match the AC timing delay specifications of the logic. This type of timing is referred to as *AC Scan.* AC Scan verifies both logic structure and timing performance. The amount of delay time needed is dependent upon the path delays of the specific circuit being tested.

With Scan Enable held inactive, the next clock will cause the flip-flops to capture the output logic response in parallel (non-Scan mode). During this cycle the actual test takes place. Some of the test results will propagate to primary outputs and are tested at this time. Other test results are captured into the Scan chain.

Last, Scan Enable is returned to its active state, and the captured output response is shifted out serially for comparison by the test system. During this sequence the data for the next Scan test are shifted in.

Initial Scan Test Sequence

As mentioned, there is DC Scan, and AC Scan. We will begin by gaining an understanding of DC Scan before looking at AC Scan. As we look at the examples keep in mind that the rising edge of the clock captures data into the flops.

As shown in Figure 2-8, the first Scan sequence normally begins by executing a number of conditioning cycles applied to the primary inputs. During this sequence the circuit is in normal functional mode, SE is held low.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

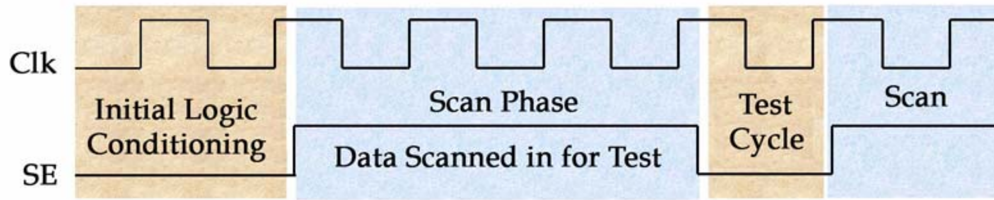

Figure 2-8 First Scan Sequence

Once conditioned, SE is set high and the circuit enters Scan mode. Data are clocked into the Scan chain, via SDI, in preparation of the first test. The test is *launched* (begins) on the last rising edge of the clock, while the circuit is in test mode.

SE is then set low, returning the circuit to normal functional mode. On the next rising clock edge the test results from the combinational logic, are captured into the Scan chain.

The sequence ends by setting SE high, placing the circuit back into Scan mode. The test results are then shifted out, via SDO, for evaluation.

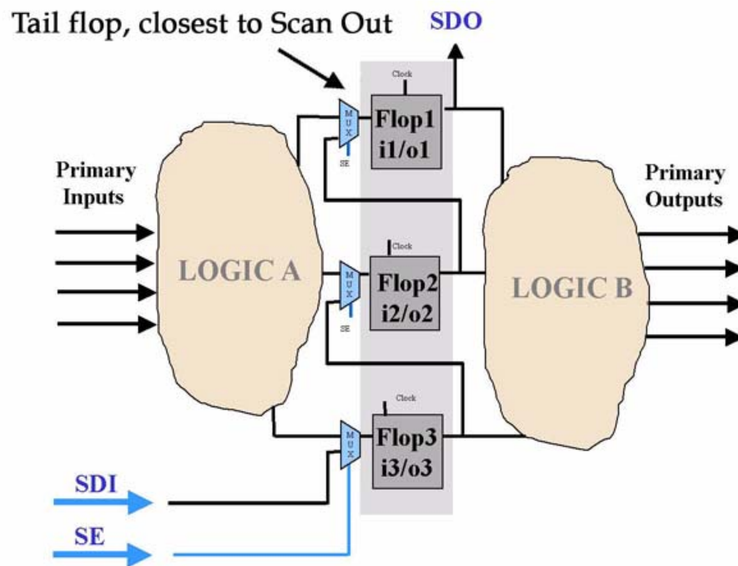

Figure 2-9 Scan Chain Example

Figure 2-9 shows a three flop Scan chain register. This will be used for our following timing examples. Notice that SDI is connected to the input of a flop labeled Flop3. The flop closest to the SDI is called the *head flop*, the flop closest to SDO is called the *tail flop*. Notice also that the tail flop is labeled i1/o1. In this case the 1 indicates its position within the chain, it is the first flop. The lower case i and o will be used to differentiate input data (i), from output data (o) in our timing diagrams.

Scan Input / Output Timing

The left panel of Figure 2-10 shows three data bits being clocked into the Scan chain on the rising edge of the clock, via the SDI input. The input signal timing on SDI is normally relaxed timing, providing maximum setup and hold time. In Figure 2-10 (left) time begins on the left with the first data bit clocked in, labeled i1, ending up in the tail flop (Flop 1). After the third Scan clock all data bits are in place and ready for test which begins on the last rising clock edge.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

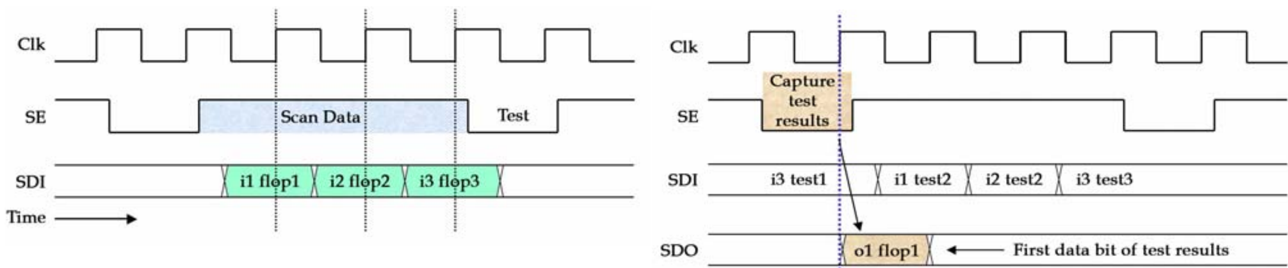

Figure 2-10 Scan Input /Output Timing

The panel on the right, in Figure 2-10, begins with the capture cycle. Because SE is low the circuit is in normal functional mode, so the flops receive data via their normal D input. On the rising clock edge of this cycle the test results are captured into the Scan chain register. The result held in the tail flop is immediately available at SDO, before the first Scan clock. The primary outputs are also evaluated at this time.

Scan Output Timing

Looking at the left panel in Figure 2-11, the ATE strobe positioning is seen. Normally the strobe is positioned just before the rising edge of the clock, providing maximum time for the output data to propagate to SDO. Attempting to test for the tail flop propagation delay is not a consideration during a Scan test.

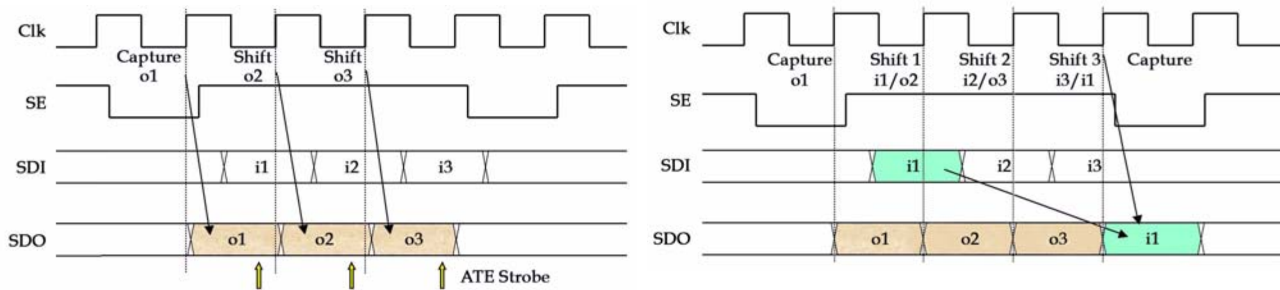

Figure 2-11 Output Timing

It is the rising edge of the clock that times the capture of data into the Scan chain. The tester sees the results as they are scanned out of SDO, cycle-by-cycle. After the data are captured in parallel the result captured by the tail flop (o1) is seen on SDO. The first shift clock shifts the data in the next Scan flop data (o2) into the tail flop and is presented on SDO. This continues until all the data are shifted out of the Scan chain and examined by the tester. You can think of the Scan chain acting like a very simple ATE comparator to capture the test results. The main differences are that the Scan chain register has only a single non-programmable threshold to evaluate the captured logic levels, and the capture timing is shared by the entire chain.

Recall that as the captured output data are shifted out for evaluation, input data for the next test are shifted in. After all output data are shifted out, one more cycle is needed to shift in the final input data bit of the next test. Once the final data bit is clocked into place, the Scan chain register is fully loaded. At this time the first data bit shifted in (i1) is residing in the tail flop, and also appears at SDO. The panel on the right side of Figure 2-11 illustrates this sequence. On SDO the data sequence is a string of test result bits, followed by the first input data bit of the next test.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

Input Timing

On the left side of Figure 2-12 it can be seen that when the circuit is in normal functional mode, SE low, the primary inputs are used to apply data to the device. At this time SDI is not active, data applied to SDI is ignored.

The right side of Figure 2-12 shows that when the device is in test mode the primary inputs are not active, input data is applied to SDO.

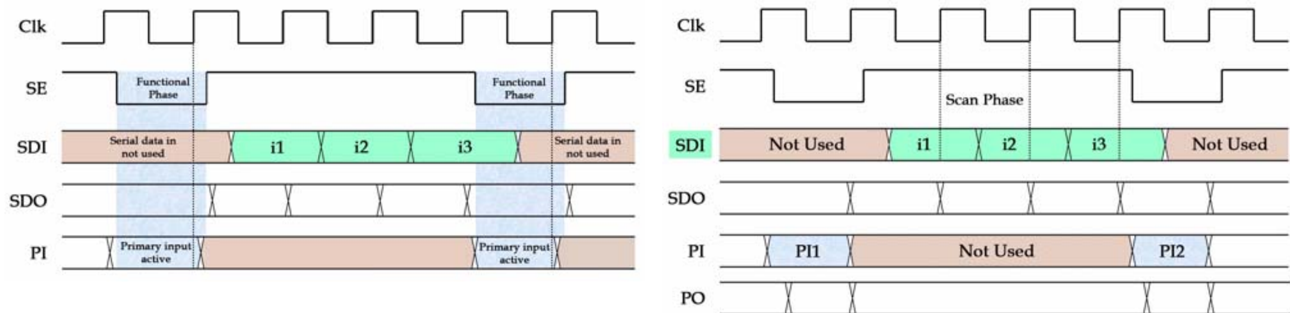

Figure 2-12 Functional and Scan Input Test Timing

The primary inputs are used to condition the logic before data is scanned into SDI, and they are used to apply data during the Functional capture cycle. Data does not change on the primary inputs during the Scan cycles unless the primary inputs share their functions as Scan pins (more on this soon). When the primary inputs are active the Scan inputs are not used, and when the Scan inputs are used, the primary inputs are not. Their use is mutually exclusive.

Output Timing

On the left side of Figure 2-13 it can be seen that when the circuit is in normal functional mode, SE low, the primary outputs are evaluated. At this time the data that appears on SDO is the first bit of data (o1) that was loaded into the Scan chain register. This is not a test result, evaluation of this data is not important.

The right side of Figure 2-13 shows when the device is in Scan mode the primary outputs are not active, output data is evaluated at SDO.

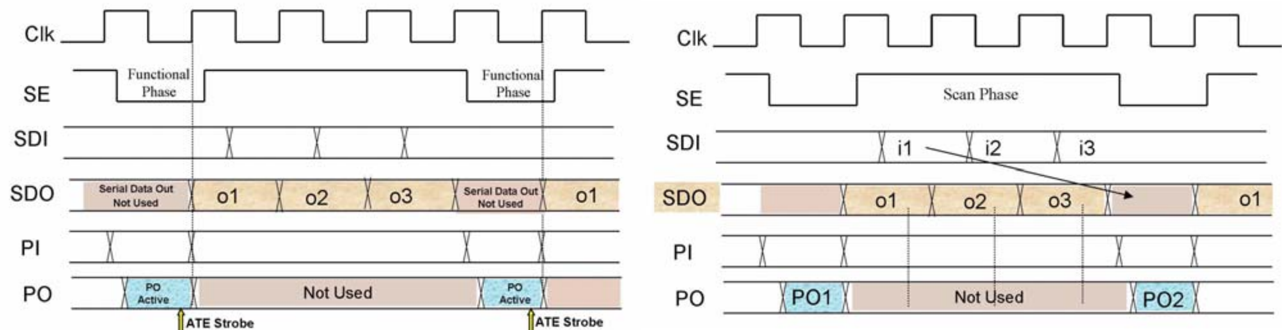

Figure 2-13 Functional and Scan Output Test Timing

When the primary outputs are active the Scan outputs are not evaluated, and when the Scan outputs are evaluated, the primary outputs are not. Their use is mutually exclusive.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

These timing examples illustrate how primary IO's can also be used as Scan pins. Since their use is mutually exclusive there is generally no conflict between when a pin can be useful for Scan input and Functional input, and likewise for Scan output and Functional output.

Depending on the circuit design, certain pins may be able to share their activities between normal functions and Scan functions. This is sometimes referred to as a *borrowed* Scan interface. The circuit can be designed so that a normal input also acts as a SDO, or a normal output also acts as a SDI. The primary inputs and outputs can be designed as IO circuits, even if not required for normal operation, so almost any pin function combination is possible.

Additional circuitry must be added to handle the change in role between Functional and Scan mode, with the borrowed Scan interface. Additional timing signals are required to control the exact timings involved in the role change. Sometimes these signals are generated internally (on chip), other times they are generated from the ATE system.

Testing Scan Logic

During a Scan test the first test sequence is often applied to the Scan logic circuitry, not the combinational logic. The purpose of the test is to verify the correct operation of the Scan (test mode) logic. You can think of the Scan chain registers as being an extension of the "tester" located inside the chip. Testing the Scan circuitry is similar to running the tester diagnostics, if these tests fail, all other results are questionable. Once proper operation of the Scan circuitry is verified, testing the combinational logic begins.

This test sequence differs from what has been discussed so far. Until now we have discussed targeting likely faults, then testing the combinational logic in an attempt to detect specific faulty behavior. When testing the Scan circuitry, the approach is more like a functional test - does the circuitry function as a shift register? This test sequence does not involve a functional test cycle. Scan Enable is always held high and the data entered at SDI is expected at SDO.

A common testing strategy for verifying the Scan logic is to set Scan Enable high to configure the flip-flops into a Scan chain register. The status and operation of each flip-flop is tested using the SDI, SDO, and Clock signals. Suitable tests for Scan chain registers are:

1. *Flush Test* (see Figure 2-14 left) - In this test all flip-flops are initialized to logic 0, and a single logic 1 is clocked through from SDI to SDO. The procedure can then be repeated with a single logic 0 flushed through a background of logic 1's. This sequence checks the ability of each flip-flop to assume both logic states.
2. *Shift Test* (see Figure 2-14 right) - In this test, the data sequence 00110011... is shifted through the register. This sequence exercises each flip-flop through all combinations of logic states.

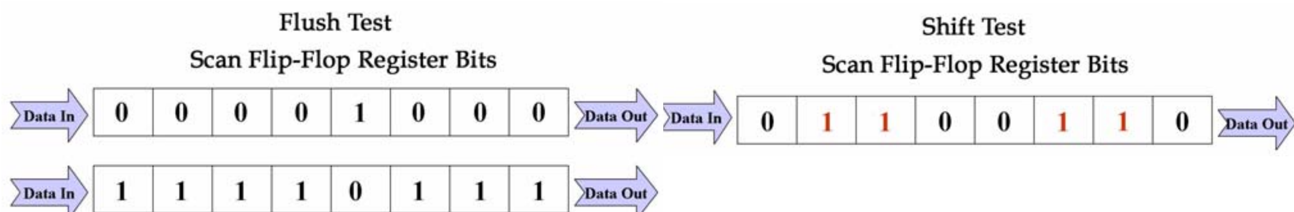

Figure 2-14 Verifying the Scan Logic

There seems to be two schools of thought on when to execute the Scan logic test. Some companies run these tests first, followed by the combinational logic tests. Others only run the Scan logic tests if there is a failure of the combinational logic. Either approach will determine the cause of the failure.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

Scan Logic Fault Detection

Figure 2-15 shows a Scan chain containing a wire-AND bridge fault. Recall that a wire-AND defines a 0 level as dominant. If the 11001100 shift pattern is clocked through the register chain would the bridge fault at nodes E and C be detected? Would either of the Flush patterns be capable of detecting this fault? If so which pattern?

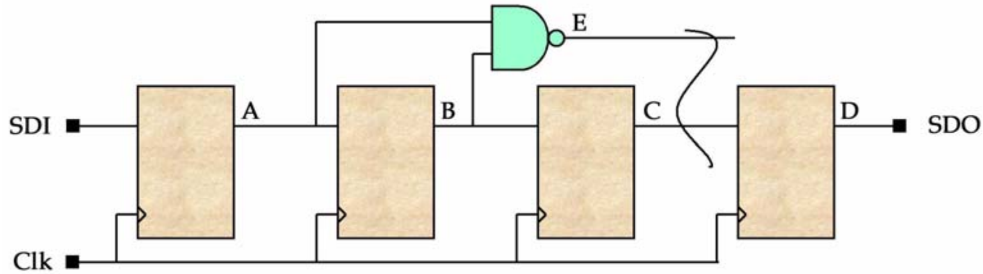

Figure 2-15 Shift Test Problems - Bridge Fault

Answer: This bridge will only be detected when A, B, C are all set to 1, so the 00110011 pattern does NOT detect it. Node E does not propagate to SDO output, so Node C must be forced to a 1 while node E is forced to 0. The Flush pattern that walks a 0 through a background of 1's will detect this fault. When nodes A, B, and C are all 1's node E will be at 0. This will over drive node C, forcing it to 0, and corrupt the data in the Scan chain.

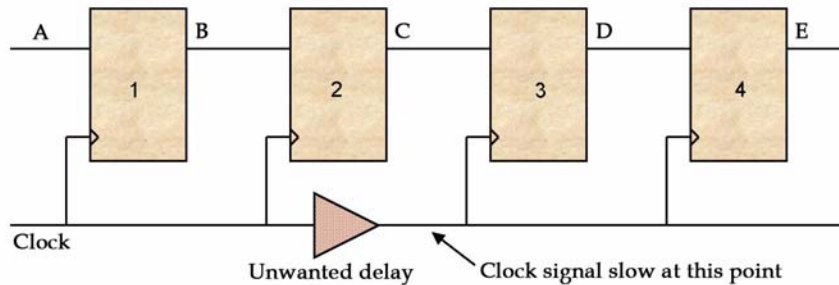

Figure 2-16 Shift Test Problems - Clock Skew

Figure 2-16 illustrates a delay fault in the clock line that occurs between flops 2 and 3. After the next Scan clock all data should advance to the next flop. The flop outputs should be A, B, C, D, but the delay fault will cause the data to become corrupt. The data on the inputs to flops 1 and 2 will advance correctly. The clock will arrive late to flop 3, by the time the clock arrives the output of flop 2 will be B, and B will be the input to flop 3. C is now lost. The data in the flop chain will be A, B, B, D, instead of A,B, C, D. The Shift and Flush tests are designed to detect these types of problems.

Scan Data Alignment Principles

When multiple Scan chains are used it is likely that not all chains will have the exact same length (number of bits). Since data is shifted through all chains in parallel, sharing a common clock, data on all but the longest chain must be adjusted so that all chains contain the proper data on the last Scan clock. Recall that as the results are shifted out from the last test, data are shifted in for the next test.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

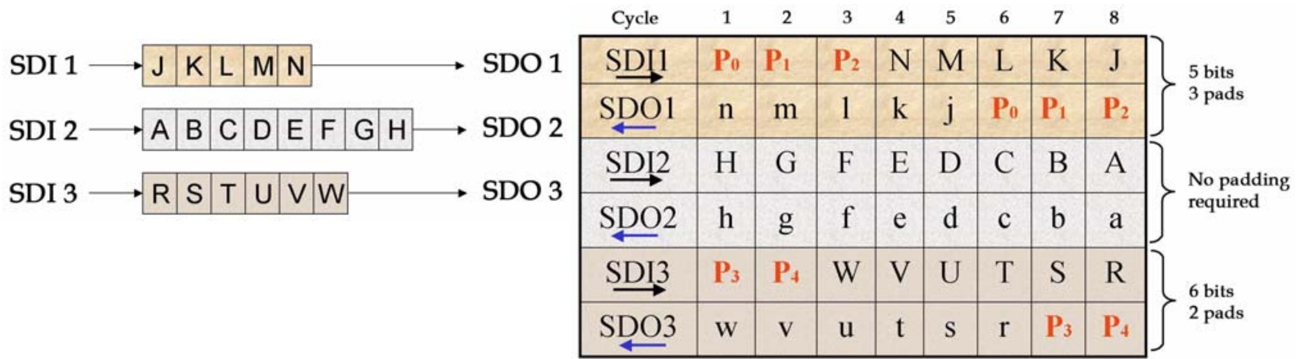

Figure 2-17 Scan Data Alignment Across Multiple Chains

The left side of Figure 2-17 shows three chains of different lengths, 5, 8 and 6 flip-flops. Notice where SDI, and SDO are shown on the diagrams above. Data comes out as shifting begins, and the data closest to SDO appears first. Data goes in at the end of the shift sequence, and the data that is closest to SDI in entered last. Adjustment bits called *pad data* are needed when the chain lengths differ. Pad data is used to properly align the input data.

The middle chain (SDI2/SDO2) is the longest with eight bits, and requires no pad data. Chain 1 has only 5 bits and will require 3 pad bits, chain 3 contains 6 bits and will require 2 pad bits. The diagram shown on the right side of Figure 2-17 shows the required padding data and their locations. Notice the cycle number when the pad data is applied to SDI and when it again appears at SDO. The pad data is the first data entered at SDI, and the last data to appear at SDO.

As Scan shifting begins, the result data from the last test appears immediately at SDO. For all but the longest chain. Input data at SDI must be delayed as the results of the last test are shifted out. The last input data bit must be clocked into the Scan chain register on the last shift cycle. Pad data controls the placement of input data, but is not used, it is simply a place holder. On all but the longest chain the first Pad data bit is shifted in, as the first test result bit is shifted out. All pad data is eventually shifted out and discarded. Pad data is essentially flushed through the Scan chain, similar to the Flush tests described earlier. In order for the Scan test to pass data alignment must be correct!

Shifting Phase - What's Important

The goal of "shifting" data through the Scan chain is to properly transfer data into the device, or to properly transfer results to the tester. It is important to understand that shifting is a part of the test sequence, but it should not be part of the "test". The Scan logic should not be stressed during the shifting phase. The routing of the signals from flop to flop, to form the Scan chains, is not the normal signal paths used in functional mode. The signal timing may be substantially different between functional and Scan modes. Shifting at too high of a frequency may cause corruption of data within the Scan chain. So, at what frequency should the shifting occur during a Scan test? The most common answer is a "Safe" frequency. Remember the test occurs during the Functional phase, not the Shifting phase.

CMOS gates use little power except when they are switching, but during a Scan operation many flops change value on each shift, let's assume about half. This is much higher than the number of state changes in normal functional mode. When shifting at a high frequency the device may consume far more power than during normal operation. This can result in excessive internal currents, and may exceed the capability of the power busses of the device, resulting in overheating.

DC Scan Test Failures

There are a number of issues that can cause a DC Scan test to fail. New silicon may contain Scan design errors. Fortunately the EDA tool are "intelligent" and tend to minimized design errors, but the possibility still exists. When design errors do occur they will generally affect all, or many of the devices. Potential design issues will include shift

Copyright © 2008 Soft Test, Inc. All Rights Reserved.

problems due to clock timing, internal signal contention when shifting, and asynchronous effects that disrupt shifting such as improper control of the preset and clear signals on the Scan flops.

The ATPG tool may have generated vector data using the wrong circuit version (let's hope not!). Script files and various translators are usually involved, and they may generate incorrect data. Problems can also exist such as input/output data misaligned, or chain length alignment can be wrong across multiple chains. And problems often occur when a single chain crosses timing domains.

The test conditions provided by the ATE system can also cause failures. One common problem is caused by running the Scan clock at too high a frequency. As the test frequency increases the power consumption increases. This generates additional power-rail noise and causes the device to heat up. If the frequency is too fast the device will misbehave. This should be considered a "Test Issue", not a defective device issue. Remember think "Safe Shift".

The timings associated with SDI, SDO, and SE should be relaxed timing, providing the best conditions for the circuit to properly function. The timing should provide maximum setup/hold time on SDI, and maximum propagation delay time on SDO. The voltage levels associated with inputs and outputs should also be relaxed. Set VIL at or near ground, VIH at or near VDD. For SDO set the comparator levels to the midpoint between VOH and VOL.

Scan tests are often executed at VDDnom, however each company must make its own decision regarding power level settings. Power goes everywhere so it can provide a global "stress condition". Power settings can amplify "analog defect effects", turning analog behavior into digital behavior. One possibility is to use a very low VDD value, making slow behavior look even slower. Care must be taken when selecting the very low VDD level, characterization data will be useful.

AC Scan

Certain defects can affect a signal's timing by slowing the signal. If given enough time, the final steady state logic level will be correct. This type of behavior may pass a DC Scan test, because DC Scan is not timing critical. Timing behavior can only be detected by creating a signal transition, then testing for the correct timing performance. There is hope that if AC Scan techniques can be perfected they will provide a less expensive way to speed grade devices. Currently "Fmax" functional testing is used for this purpose.

The frequency used to scan data in and out of the device is generally the same for DC Scan or AC Scan. For AC Scan, the time interval between the launch and capture events is adjusted based upon the clock period. "Safe Shifting" still applies to move the data into and out of the device successfully.

There are two common fault models associated with AC Scan. There is the Transition Fault, also called Gate Delay Fault, and the Path Delay Fault.

To create an AC delay test, a test is generated for a specific delay fault using either a Transition Delay, or Path Delay model. The test result must be observable at a single output, and all signals involved in the test must avoid multiple transitions that may allow test escapes. A delay test must guarantee that if the transition at the input takes longer than the allowable propagation (Tpath) delay time to get to the output, the test will fail.

When performing an AC Scan test the goal is to verify that a signal will propagate through a single gate, or a combinational logic path, from input to output within its allowable time, usually within 1 clock period. All AC tests require a signal transition. Keep in mind that input data for an AC Scan test will be loaded through the Scan chain register, one clock cycle at a time. In order to create a signal transition two data states are needed, so two clock cycles are required.

Recall from our earlier discussion of fault models the term vector pair is the data associated with two clock cycles of the Scan chain. For example, in one cycle a "0" can be applied to a gate input connected to a Scan flop. In the next cycle a "1" can be shifted into the flop, creating a 0 to 1 transition. This can be the launch event that starts the test.

Copyright © 2008 Soft Test, Inc. All Rights Reserved.