ScanBist:
A Multifrequency Scan-Based BIST Method

BENOIT NADEAU-DOSTIE
DWAYNE BUREK
ABU S.M. HASSAN
Bell-Northern Research

The authors present ScanBist, a low-overhead, scan-based built-in self-test method along with its performance in several designs. A novel clock synchronization scheme allows at-speed testing of circuits.

BUILT-IN SELF-TEST is gaining popularity as a means to address test issues at the different packaging levels of digital systems. BIST does not require that patterns be stored in the test equipment, which simply provides a clock and a few control signals. This feature becomes especially important when testing high-performance systems. BIST makes the chip, board, and system more independent of the specific test resources available at each manufacturing stage. It also provides a convenient method of applying more test patterns to compensate for the weaknesses of the stuck-at fault model.[1]

Specialized self-test circuits for embedded blocks (such as memories and programmable logic arrays) have reached a good level of maturity, and existing methods provide the basis for making adequate quality-area-performance trade-offs. However, BIST has a great deal of room for innovation, particularly in unstructured or random logic that requires testing for performance when a variety of design styles apply. At the same time, the new techniques that provide this capability should not significantly increase either the implementation cost or the design interval of the system.

In telecommunications applications, designers often create circuits that operate at different frequencies derived from at least one reference frequency. For example, designers embed communication protocols in a hierarchical fashion in a single frame of data. Several finite-state machines, working at appropriate subfrequencies, then extract the embedded protocols by simple decimation of the input sequence. Using subfrequencies derived from a master clock instead of the master clock itself saves power, reduces electromagnetic interference, and reduces silicon area.

Traditionally used to generate very high quality tests for faults detectable at low speed, Scan has also demonstrated efficiency in testing for delay faults by accurately controlling all clock phases from a tester.[2] However, BIST does not lend itself to this approach. LeBlanc developed a scan-based BIST solution, LSSD on-chip self-test (LOCST), that is applicable at all levels of system integration.[3] However, he used a single test frequency and limited the speed of application of the scan patterns generated by the BIST circuitry to the lowest frequency used in the system. Therefore, part of the logic operates at an untested speed in systems with more than one frequency.

Overview of ScanBist

Figure 1 (next page) shows a block diagram of the ScanBist implementation. The IEEE Standard 1149.1 test access port (TAP)[4] connects to a macro circuit called the ScanBist core. In turn,

0740-7475/94/$04.00 © 1994 IEEE

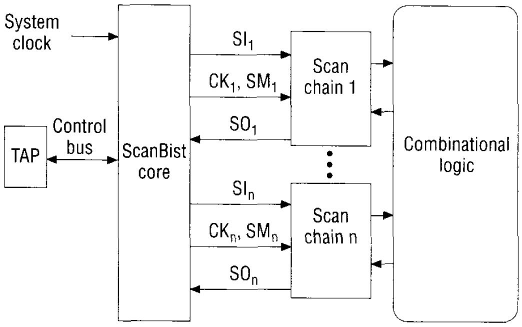


***Figure 1.** ScanBist implementation overview.*

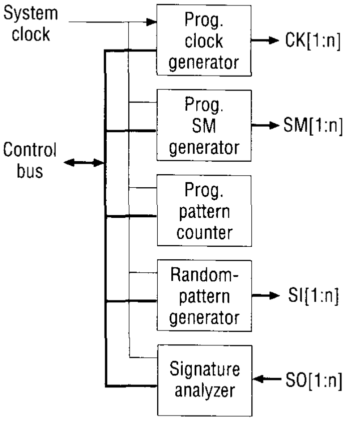


***Figure 2.** ScanBist core structure.*

the ScanBist core connects to the various scan chains (1 to *n*). We use the multiplexed-scan methodology.[5] The scan mode signals determine the configuration of the registers (either shift registers or parallel-load registers). For a given register, this scan methodology uses the same clock for normal operation and test. Each scan chain has its own scan-in and scan-out pins. In principle, each scan chain operates from a separate clock and scan-mode signals. In practice, several chains are likely to share the same clock and scan mode.

Some of the scan chains are segments of the boundary-scan chain. This chain isolates the chip under test from the tester and other chips in the system. The ScanBist core provides the clocks for these segments, choosing them to reflect the system timing as closely as possible. However, we avoid complex partitioning of the boundary-scan chain and clock distribution schemes.

The ScanBist core is itself scannable. A single scan chain controlled from the TAP can test the complete circuit (including the ScanBist core). This option provides compatibility with the test method currently used at Bell-Northern Research. This might prove useful when we cannot use the system clock for chip retesting during board manufacture. Also, it provides better diagnostic capabilities at the chip manufacturing stage.

ScanBist allows synchronization of multiple chains running at different frequencies, generated by division of a reference (system) clock, driven from primary inputs or both. The technique is a significant improvement over the LOCST method[3] developed by IBM. The synchronization of the various scan chains allows performance testing of the circuit in an environment that closely reflects reality. A fault simulator for combinational circuits performs an exhaustive stuck-at and transition fault coverage analysis.

We do not restrict the subfrequencies derived from the input clock to multiples of one another. However, the synchronization of the different chains requires that the number of clock cycles needed to load the scan chains and perform the sampling operation are divisible by all frequency ratios (ratio of the highest frequency over a particular frequency). Another constraint calls for a fixed phase relationship between the subfrequencies during the test.

ScanBist provides compatibility with various clocking methodologies such as edge-triggered, two-edge, and a flavor of two-phase developed at BNR.[6] The complete circuit is tested in one pass (there is no scheduling involved). This method has little impact on the way the circuit is usually designed.

A relatively simple protocol invokes ScanBist.

1. Initialize ScanBist core through the TAP.
2. Scan-in maximum pattern count and/or seed for pattern generator and signature analyzer.
3. Enable BIST.
4. Wait for the DONE flag indicating the end of the BIST test.
5. Scan-out signature.
6. Iterate the procedure starting at step 2 if multiple signatures are necessary.

ScanBist core

The ScanBist core circuit (Figure 2) consists of a random-pattern generator, a signature analyzer, a pattern counter, a scan mode signal generator, and a clock generator that the system clock generator can share. Each component of the ScanBist core works at the frequency imposed by the system clock input during self-test mode. We designed the ScanBist core to ensure better portability of the technique from one technology to another. In the current implementation, we have limited the number of scan chains to 16. This provides a good compromise between the size of the ScanBist core and the test length required to test the circuits we describe later. A future version of ScanBist will make the number of chains a user-programmable option.

A conventional 24-bit linear-feedback shift register (LFSR) implementing a primitive polynomial known by the automatic test pattern generator provides random-pattern generation. The cycle length exceeds 16 million, which is appropriate for random-pattern testable designs. It is acceptable to go over the cycle more than once as long as the length of a scan pattern is not a multiple of the cycle length. In this case, the LFSR reapplies the same patterns to the circuit.

The signature analyzer is a conventional multiple-input signature register (MISR) based on a primitive polynomial. We decided not to hardwire the signatures because of the extra area required to implement the corresponding comparators and the difficulty of verifying the design. In addition, the design of the ScanBist core only becomes final at the very end of the design cycle, thus possibly creating a bottleneck. Instead, we scan out the final signatures using the TAP.

The programmable pattern counter indicates when to stop the BIST. It causes the ScanBist core to hold its state until the tester reads the signature and other registers from the TAP. Using more than one maximum count allows for tester time optimization. The number of patterns applied before sampling a new signature progressively increases to detect faulty chips as quickly as possible during manufacture. For example, the tester examines the signature after 16, 256, and 64,000 scan patterns. Loading appropriate seeds in the pattern generator and always using the same state as the final one could eliminate the programmable pattern counter. However, the ATPG program would need an extra preprocessing step to compute the appropriate seeds using the reciprocal polynomial, significantly increasing the runtime.

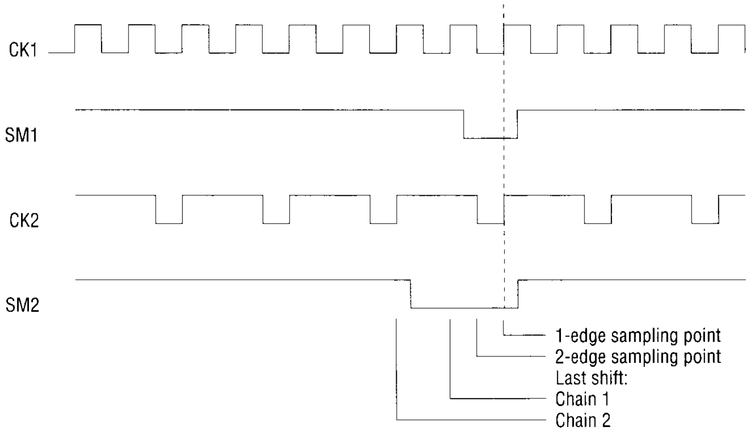


Figure 3. Fixed pulse-width timing.

Having the BIST technique work in a fully configured system where the system clock(s) will run asynchronously relative to the TAP clock requires a mechanism to compute the number of patterns. As the TAP clock runs significantly slower than the system clock(s), it is impossible to stop the BIST circuit at the right time from the TAP.

The clock generator enables the scan mode signal generator. The number of clock cycles for one scan pattern (including the capture cycle) must be a multiple of the ratio of the highest to lowest frequency in the circuit. Gating the output of this counter with the clock generator output generates the scan mode signal associated with each subfrequency. To ensure that a last-minute change of the circuit (specifically, the addition of flip-flops in the scan chain with the longest effective length) will not require a modification to this counter, its cycle length should be longer than required.

Having the cycle length longer than necessary guarantees that new data has flushed out all scan chains and the signature analyzer contains the response from the previous scan pattern. Of course, the price to pay is slightly longer execution time. A second solution makes the counter programmable and scans in the appropriate value during the initialization of the ScanBist core.

We prefer a single counter generating subfrequencies that are powers-of-two as the clock generator. Nonpowers-of-two subfrequencies require separate counters plus one extra to trigger the scan mode signal generator.

Clock synchronization

Two methods are available to generate subfrequencies of the input clock: fixed pulse-width and 50% duty cycle. Both methods offer different characteristics that make them useful in different contexts. We can use both methods on the same chip.

Figure 3 shows the fixed pulse-width method. The timing diagram indicates the relationship between the control signals of two scannable registers, one running at half the speed of the other as happens during normal operation of the system. The registers belong to different scan chains. In this diagram, CK1 and CK2 represent, respectively, the clocks applied to chains 1 and 2. We assume that chain 1 works at the rate imposed by the master system clock frequency.

Furthermore, we assume that all flip-flops update their output on the rising edge of the clock. SM1 and SM2 denote the scan mode signals for chains 1 and 2. These signals are active high. That is,

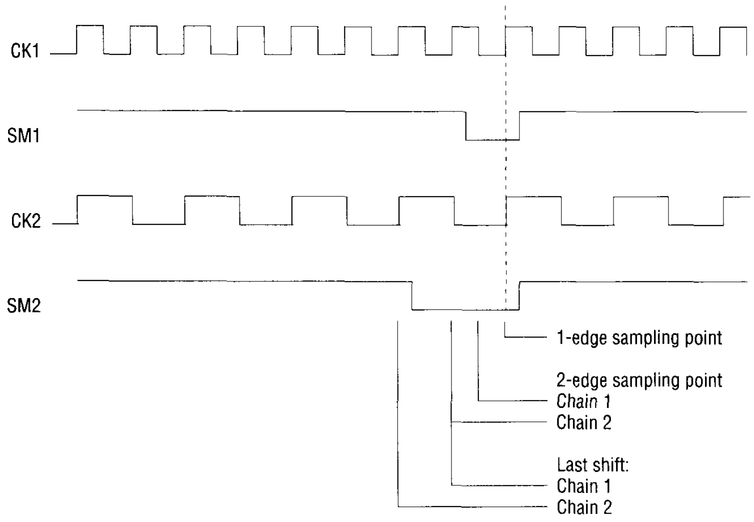


***Figure 4.** 50% duty cycle timing.*

data shifts in and out of a chain when the corresponding scan mode signal is high. The system samples the output of the combinational circuit when the signal is low. We call this mode the scan sample mode.

In this example, we obtain CK2 by gating the negative-going pulse of CK1 every two clock cycles. Therefore, the resulting clock is not of the 50% duty cycle type. This method of generating the clock allows the use of a mixture of clocking techniques at the interface between the two frequency domains (as shown by the position of the sampling points on the timing diagram). The interface with the boundary scan chain, which uses a two-edge clocking technique, generally requires this approach. Clock skew between the frequency domains provides another reason to use two-edge timing at the interface.

This timing diagram shows an important constraint on the synchronization between the different chains. During the scan sample mode cycle, all flip-flops sample after the pattern generator loads new data (that has had time to propagate through the combinational circuit) into the scan chains. In addition, sampling must occur before any flip-flop updates its output. On the timing diagram, all single-edge triggered flip-flops are sampling (updating) their functional input (output) at the time marked "one-edge sampling point." The two-edge flip-flops are sampling their functional input at the time marked "two-edge sampling point." Again, all flip-flops update their output at the same time. Under these conditions, we can still analyze the circuit as a strictly combinational circuit whose inputs have different arrival times for a given test pattern.

We must control the skew between clocks to allow maximum performance of the circuit during both test and normal operation. The arrival time of the scan mode signals at the flip-flops simply needs to meet the normal setup and hold constraints. Therefore, both scan chains could use SM1 if it proves more convenient.

Figure 4 shows a timing diagram for the case where we use the 50% duty cycle subfrequency generation method for the same frequencies as before. However, consider the case of a signal originating at a chain clocked by CK1. If a flip-flop captures this signal in a chain clocked by CK2 and using two-edge timing, it may capture invalid data. This can occur because the sampling operation on the chain clocked by CK2 coincides approximately (within clock skew limits) with the last shift on chain CK1.

The above conclusions concerning two-edge clocking also apply to a flavor of two-phase clocking developed at BNR. Wilcox et al. describe this in detail.[6] Compatibility of this version with other two-phase clocking schemes requires more analysis.

The cycle time of the scan mode signals remains the same for all scan chains in both the fixed pulse-width and the 50% duty cycle methods. That is, the various scan chains will sample circuit response the same number of times. Only the duty cycle of the scan mode signals might change. The cycle relates to the effective scan chain lengths. We define the effective scan chain length as the time it takes to shift data through a scan chain and perform a sample cycle expressed in terms of clock cycles of the highest frequency used to activate the flip-flops in the circuit.

For example, suppose that CK1 clocks four flip-flops and CK2 clocks only two. The effective length of the scan chains is

$$\begin{aligned} L_{\text{eff1}} &= 4+1 = 5 \\ L_{\text{eff2}} &= (2+1) \times 2 = 6 \end{aligned} \qquad \textbf{(1)}$$

It then takes six clock cycles to perform a scan cycle (that is scan in, scan out, and capture on all chains). If we change the number of flip-flops in the first chain to six, the effective lengths are $L_{\text{eff1}} = 7$ and $L_{\text{eff2}} = 6$. The effective length of the first chain is now larger than that of the second chain and should impose the length of the scan cycle. However, the scan cycle must be a common multiple of all frequency ratios. Therefore, eight clock cycles are necessary.

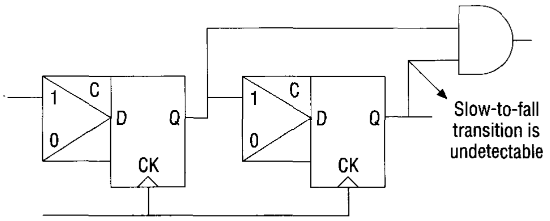


Figure 5. *Undetectable transition fault.*

When a circuit has more than one system clock connected to its primary inputs, the clock with the highest frequency generates the subfrequencies for the whole chip. If the ratio of the frequencies of the various clocks is not an integer, the designer needs to use a frequency as close as possible to the operating frequency. However, this frequency must not exceed the constraints imposed by the longest paths in that frequency domain. We do not account for metastability issues due to phase uncertainty between the various system clocks with this technique. Note, however, that this constraint exists for most digital testers.

Fault coverage considerations

One of the main reasons to use the self-test approach is to increase coverage of faults of all types compared to conventional approaches. In our environment, however, the only objective criteria we have to evaluate self-test fault coverage are the stuck-at and transition fault coverages reported by the fault simulator. The strategy adopted here maximizes the coverage of faults for which we have a metric. In addition, it uses some heuristics to roughly determine the coverage of other faults. For example, Waicukauski et al.[7] showed that applying 30 times as many patterns as required for a stuck-at test gives an excellent coverage of other faults. Pancholy et al.[1] obtained similar results with each stuck-at fault detected at least twice (most of them 710 times) to generate the reference test set. Note that this reference test set detected more faulty chips than the single stuck-at fault test set.

Applying the patterns at the normal operating frequency of the circuit might significantly increase the effectiveness of the test. This motivates the use of ScanBist since we only need to provide one high-speed signal (the master clock) to the circuit. This greatly reduces the requirements for sophisticated pin electronics on testers.

Using the test point insertion tool described by Seiss et al.[8] (see Results section) has resulted in consistently high stuck-at fault coverage. To reach the specific test objectives, we have added control and observation points as necessary. Consider the case where we require a 99% stuck-at fault coverage after 64K scan patterns. The insertion of test points requires 10 times less area than implementing test circuitry used to generate weighted random patterns with a single five-value weight set.[9]

The correlation between scan data inputs of different scan chains can have a significant impact on the stuck-at fault coverage. Coverage reductions up to 5% occurred when we connected scan data inputs to adjacent flip-flops on the random-pattern generator. Note that Exclusive-Or gates (used to implement the polynomial) did not separate these flip-flops. This caused two or more inputs of a logic cone driven by different scan chains to systematically receive the same data. In turn, this reduced the number of possible input combinations and made the detection of some of the faults impossible.

For the circuits described later, we found an appropriate subset of the outputs of the pattern generator (LFSR) in a few iterations. However, preventing this problem requires a more rigorous method like LFSR segmentation[10] and cellular automata random-pattern sources.[11] We prefer cellular automata because of its effectiveness, ease of implementation, and reasonable size.

A nontrivial problem in scan-based designs involves applying ordered pairs of vectors that lead to detection of stuck-open or delay faults.[12] The correlation between consecutive bits of a scan chain reduces the number of these faults that can be detected (see Figure 5). This explains why we don't achieve as high a coverage of transition faults for the circuits described later. We are currently investigating various options that would improve the transition fault coverage and, indirectly, the coverage of stuck-open and delay faults.

A trivial solution would involve inserting extra flip-flops between the ones needed to implement the circuit's function. These flip-flops could be the same ones used to implement control and observation points. Dervisoglu and Stong[13] propose a solution where they use specially designed flip-flops that have an extra latch to hold the output during scan shifting. A third solution would entail reordering the scan chains to eliminate or at least minimize the correlation effect. The first two solutions are hardware intensive, and the third requires sophisticated software support. These alternatives require further evaluation.

Using ScanBist results in a further reduction of delay fault coverage along paths originating at flip-flops clocked at low frequency and ending at those clocked at high frequency. After completion of the last shift on the slowest flip-flops, the fastest flip-flops are still shifting data through the scan chain, ignoring the effect of the last change of the slowest flip-flops. However, during

Design rules

Synchronous design:

- All memory components of the circuit are D-type flip-flops connected to a scan chain.
- The rest of the circuit can be levelized.
- Asynchronous set/reset pins of flip-flops are not driven by other flip-flops.
- Clock inputs do not fan out to flip-flop D inputs.
- Gated clocks are not used except in clock generators.

Boundary scan:

- These rules are dictated by the IEEE 1149.1 standard.[4]

Tristate bus rules (Optional, see Nadeau-Dostie et al.[16] for details.):

- All bus lines are pulled up or down by weak drivers when not driven.
- All bus drivers are turned off during scan-in/out operations.

Scan path rules:

- Each scan path starts from a primary input and ends at a primary output.
- The scan-out pin of any flip-flop can only fan out to one scan-in pin of another flip-flop.

the normal operation of the circuit, we may require that the effect of signals originating at the slowest flip-flops be propagated within the clock period associated with the fastest flip-flops. Within the limitations imposed by the correlation effect, correct coverage occurs for paths that originate at the fastest flip-flops and end at the slowest ones except in the cases where multicycle operations are used.

The aliasing associated with the signature analyzer may reduce the fault coverage. Exact fault simulation results obtained by Rajski and Tyszer[14] indicate that, under the single stuck-at fault assumption, the fault coverage decreases by 2^{-k} (on average) given a signature analyzer in steady state. This means that 2^{-k} does not bound the fault coverage loss. However, the simulation results did not show fault coverage losses significantly higher than 2^{-k}.

Determining if the oversampling of certain outputs clocked at a lower frequency than the signature analyzer might have a negative effect on the final fault coverage will require further analysis. We should also investigate the impact on the transition fault coverage. Computing the fault coverage after compaction using the efficient multiple-signature scheme proposed by Lambidonis et al.[15] controls the aliasing effect. The same method optimizes tester time. We have not yet implemented this method in our tool set.

Software support

The BNR scan design system, a modified version of the one we reported earlier,[16] supports ScanBist. The standard set of rules covers the categories described in the adjacent box.

In addition to the standard full-scan-based rules, some rules apply specifically to ScanBist.

- The control signals of the ScanBist core are hooked up correctly to the TAP.
- All flip-flops of a scan chain are driven from the same frequency clock.
- Each scan mode signal used with a scan chain is compatible. One may use a scan-mode signal from a higher frequency but not from a lower frequency.
- Boundary scan provides complete isolation of the chip under test. All paths must originate and end at a flip-flop.
- All scan paths start from and end at the ScanBist core.

The rules checker (Scanchek) also generates three files for the ATPG program: bistinfo, scaninfo_bist, and tgfile_bist. The bistinfo file contains information on the TAP settings required to activate ScanBist, inspect signatures, and so on. While most of this information is standard, the possibility of customizing exists. The scaninfo_bist file gives a list of the flip-flops in each scan chain as well as their associated clock and scan mode signals. Finally, the tgfile_bist file describes the combinational logic.

The existing ATPG was already able to generate pseudorandom patterns (applied through a single scan chain controlled via the TAP) and compute the fault coverage and the corresponding signature. This feature proves particularly useful for board- and system-level testing.[17] For ScanBist, we modified the sections of the ATPG emulating the random-pattern generator and the signature analyzer to handle several chains running at different frequencies. Also, we defined some criteria to determine which input bits the fault simulator requires for the initialization pattern. However, there is no change in the fault simulation procedure or in the calculation of fault coverage.

A 24-bit LFSR feeds a maximum of 16 scan chains through 16 taps on the LFSR. The lengths of these chains vary in terms of the number of flip-flops connected together. In the data structure, we use dummy scan cells together with flip-flops of different scan chains. Note

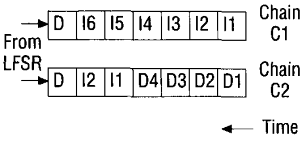


***Figure 6.** Scan chain alignment (single frequency case).*

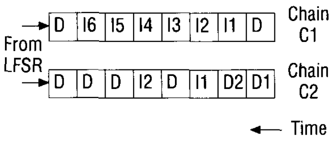


***Figure 7.** Scan chain alignment (multifrequency case).*

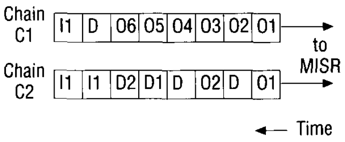


***Figure 8.** Output sequence for different frequencies.*

that dummy scan cells are a software artifact only. The model uses these cells to facilitate the generation of proper input bits in software; they do not exist in the hardware.

Let us consider an example of two scan chains, C1 and C2, of the same frequency. Suppose, C1 has six flip-flops and C2 has two flip-flops. Both chains must finish loading bits from the LFSR before going into scan sample mode cycle. C1 needs six clock cycles and C2 needs two clock cycles to load the bits. Therefore, C2 must wait four extra clock cycles before going to scan sample mode cycle. The first four bits of C2 will go directly into the signature analyzer and the remaining two bits will load into the flip-flops of the chain. For this example, the model appends four dummy scan cells at the beginning of the data structure of C2's input list, as Figure 6 shows. This ensures that the correct bits load into the flip-flops. D1, D2, D3, and D4 are the dummy cells. For chains of different lengths, the longest length chain determines the length of the input list of each chain. The model inserts dummy bits based on the longest length and the length of individual chains. Notice that one dummy cell is appended at the end of each chain to account for the scan sample mode.

Similarly, the model inserts dummy scan cells between input cells of slower frequency chains in a multifrequency environment. Let us again consider the example of two scan chains C1 and C2. Figure 7 shows this variation. C1 has frequency F1 and C2 has frequency F2, where F1 is twice as fast as F2. According to Equation 1, the scan cycle length for these two chains is eight. Therefore, C1 has one dummy bit at the beginning, six flip-flops, and one dummy bit at the end to represent the scan sample mode. Notice that the LFSR always runs at the fastest frequency (in this case, F1). C2 loads every second bit from the LFSR (because its frequency is half the frequency of the LFSR). This requires insertion of a dummy scan cell after every flip-flop in C2. C2 has two dummy bits at the beginning. Each flip-flop is followed by a dummy cell because its frequency is half that of the LFSR. Finally, two dummy cells appear at the end of this chain to represent the scan sample mode.

On the output side, the output bits load in parallel from the 16 scan chains into a 16-input MISR. The MISR also runs at the fastest frequency, similar to the LFSR. Dummy scan cells, introduced in the output data structure of the scan chains, select the correct bits going into the MISR. For multiple chains with the same frequency, the longest length chain determines the length of the scan shift cycle. Dummy scan cells inserted in the shorter scan chains make all the chains an equal length. For the multiple frequency environment, the frequency and number of flip-flops together determine the effective scan shift cycle. On the basis of this effective length, the model introduces dummy scan cells where necessary. Dummy cells also occur between outputs of every slower frequency chain to represent the scan sample mode.

Figure 8 shows the sequence of bits going into the MISR for the chains in Figure 7. In C1, output bits O1 through O6 enter the MISR followed by the dummy input bit. The first input bit I1 of the next vector goes into the MISR during the scan sample mode. Each output bit of C2 should load twice into the MISR. One dummy bit after each output bit fulfills that requirement. Dummy bits D1 and D2 follow the loading of the output bits. Finally, the first input bit of the next vector I1 loads twice into the MISR.

As mentioned earlier, the LFSR has 16 taps to feed 16 scan chains in parallel, and the MISR has 16 taps to receive output responses from 16 scan chains in parallel. However, if fewer than 16 chains exist in the circuit, unused LFSR taps connect directly to the unused MISR taps. This allows the MISR to directly compact the LFSR bits from the unused taps. Dummy scan cells represent these bits in the data structure for calculating the correct signature generated by the MISR.

As discussed earlier, the dummy cells make sure that the correct bits load from the LFSR into the circuit for fault simulation, and from the circuit into the MISR for signature generation. These cells also ensure synchronized operation of different frequency scan chains.

Transition fault testing requires a two-pattern test.[18] The first pattern initializes the node under test to 0. The second pattern applies the opposite value (1) to test that a 0 to 1 transition occurs at the node for the desired clock frequency. A 1 to 0 transition can be tested in a similar fashion by interchanging the 1s and 0s in the above explanation. Notice that clock speed is an important factor in transition testing of delay faults. Thus, for the multifrequency ScanBist envi-

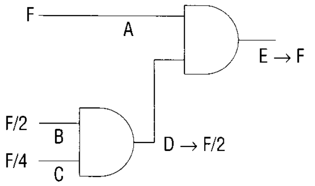


Figure 9. Transition fault testing (multifrequency case).

ronment, we must determine for which frequency the system tests a node. In this implementation, the system tests each node fed by multiple frequencies for the fastest frequency only. The example circuit in Figure 9 has frequencies *F*, *F*/2 and *F*/4. Two inputs, B (frequency *F*/2) and C (frequency *F*/4), feed node D. The system tests D for frequency *F*/2. Thus, C remains stable and transition values propagated along B-D. Similarly, two inputs, A and D, feed node E, which the system tests for frequency *F* only.

Shulz and Brglez[18] discuss three different modes of transition fault tests: simple, robust, and hazard. For the simple transition fault model, the system tests a transition only at the faulty node. However, patterns generated for simple transition faults do not cover CMOS-specific faults inside a gate very well. They use the robust transition fault test mode to detect these transistor level stuck-open faults. However, we use the CMOS fault mode for this type of faults. In this mode, we check a transition at the faulty node as well as at its output. These fault models were already available in the existing ATPG. Our modifications allow the ScanBist environment to use these models.

The tool generates any specified number of vectors simulating the LFSR. It provides the fault coverage simulating those pseudorandom vectors and the final signature (as well as intermediate signatures) of the MISR. We calculate the fault coverage for both the stuck-at and transition fault models.

Because the circuit under test differs when configured as a single-scan chain and as a multiple-scan chain, comparing performance overhead due to the implementation of the ScanBist features is difficult. However, experimental results show that the ATPG requires 15-25% more CPU time for the same number of generated test vectors when using the ScanBist option. We can generate results overnight even for large chips (100K gates).

Results

Table 1 shows pertinent circuit statistics for the various chips after insertion of scan chains but before the insertion of ScanBist logic and test points. The third column indicates the number of flip-flops clocked by a particular frequency during normal operation of the chip, with the frequency ratio indicated in parentheses. (A ratio of 1 means system clock rate *F*, 2 means *F*/2, and so on.) We see that the proportion of flip-flops clocked by the highest frequency varies between 15 and 75% of the total number.

There are also a significant number of gates (15 to 50%) with fan-in cones driven by flip-flops clocked by more than one frequency for all multifrequency circuits (see column 6). The maximum number of clock frequencies involved is indicated in parentheses.

These numbers justify the development of the clock synchronization scheme described. We can potentially improve the delay fault coverage of all these gates by applying test vectors at the system operating speed. Note that we clocked the boundary scan chain segments at the *F*/2 rate for all circuits.

Column 7 shows that there are a large number of gates with fan-in cones driven by flip-flops being part of different scan chains. In addition, in certain cases the number of chains involved (shown in parentheses) is quite high, leading to the correlation effect that we discussed earlier. For some of the chips, we effectively had to change the connections of the serial input of the scan chains to gain a few percentage points of coverage.

In Table 2 we see the implementation cost of ScanBist. The relative silicon area required to implement ScanBist and the associated test points is reasonable for small circuits and excellent for larger ones. This is because the size of the ScanBist core grows very slowly with the size of the circuit. The number of test points is proportional to the circuit size but is not the dominant factor for our test time objectives.

We further reduce the relative cost of ScanBist by taking the total chip area as the reference, especially when the chip contains custom memories. For example, the relative cost for chip B2 becomes comparable to that of B3 (about 3%) based on that criterion. Note also that insertion of the test points marginally affects the circuit statistics shown in Table 1.

Table 3 shows the fault coverage achieved for a specific number of random patterns for the stuck-at and transition fault models. We obtain the total number of clock cycles by multiplying the number of random scan patterns by the appropriate scan chain length. The unrestricted transition fault coverage is shown for reference only. We assume that all scan chains operate at the same frequency and the transition faults are simulated as described by Schulz and Brglez.[18] In the case of restricted transition fault coverage, we clock all scan chains at their respective frequency and simulate them as described earlier.

In all cases, we achieve a very high (> 99%) stuck-at fault coverage of the detectable faults by inserting the appropriate number of test points to achieve the test time target.

The transition fault coverage is more variable. As we have no specific target for designers to achieve nor adequate tools to achieve an eventual target, we did not take special precautions to improve the transition fault coverage. This table's most interesting feature is the significant difference in coverage between columns 6 and 7 for all multifrequency circuits. The unrestricted transition, which tests the output of a gate, can come from any input whereas the restricted transition must come from the inputs driven by the highest frequency. We believe this new way of computing the transition fault coverage provides a better indication of the delay fault coverage. However, this assumption requires additional investigation.

Table 1. *Circuit data before ScanBist insertion.*

Name	Gates	Flip-flops	Scan chains	Longest chain	Multifreq. (%)	Multichain (%)
B1	7,099	148 (1) 324 (2)	2 5	99 88	25 (2)	41 (4)
B2	7,686	590 (1)	10	137	0	29 (3)
B3	32,838	186 (1) 1,241 (2)	1 9	186 190	15 (2)	37 (8)
B4	5,677	176 (1) 186 (2)	2 4	93 62	17 (2)	47 (5)
B5	45,721	1,881 (1) 593 (2)	11 3	237 267	48 (2)	56 (11)

Table 2. *Implementation cost of ScanBist.*

Name	ScanBist gates	# test points	Test points gates	Total gates	Rel. area %
B1	1,028	10	112	8,239	13.8
B2	1,020	0	0	8,706	11.7
B3	1,022	25	280	34,040	3.8
B4	795	9	93	6,565	13.5
B5	1,245	54	560	47,525	3.8

Table 3. *Fault coverage results.*

					Transition faults	
Name	Patterns (K)	Stuck-at (%)	Redundants (%)	Final SA (%)	Unrestricted (%)	Restricted (%)
B1	31	98.40	1.47	99.9	93.95	85.35
B2	64	98.10	0.90	99.0	90.26	90.26
B3	31	98.25	1.26	99.5	92.55	74.83
B4	31	98.85	1.15	100.0	95.17	83.44
B5	60	97.25	1.75	99.0	93.63	93.50

THE PRIMARY CONTRIBUTION of ScanBist is its clock synchronization scheme. This design allows the testing of circuits operating at more than one frequency while retaining the combinational character of the circuit to be analyzed. We can therefore apply scan patterns that will exercise the circuit under test at the system speed, potentially providing a better coverage of delay faults when compared to other self-test methods.

Modifications to our existing transition fault simulator account for cases where inputs originating from scan registers clocked at different frequencies drive a gate. We claim to detect transition faults only if the transition originates from the inputs driven by the highest frequency clock.

ScanBist is useful at all levels of system packaging assuming that a standard TAP provides the control and boundary scan isolates the circuit from primary inputs and outputs during BIST mode. Implementation results show that the relative size of the ScanBist logic, including test points, over the total number of gates varies from 3.8 to 13.8%, for chips containing 47K to 6.5K gates. Very high stuck-at fault coverage (>99%) resulted for all circuits with less than 64K scan patterns.

As for any scan-based approach, the correlation between inputs makes it difficult to achieve high transition fault coverage. Research in this area may provide practical solutions to this problem. In addition, we need to establish a fault coverage target that will result in a significant increase of the quality level of the circuits produced.

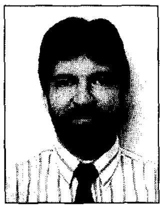
Acknowledgments

We thank Robert Hum, whose continuous support and encouragement were essential in the development of this BIST technology. Stephen Sunter provided the initial motivation to investigate the multifrequency aspect of this work. Patricia Birks and Saman Adham helped to collect the data from the various chips. The careful review of this work by the referees and their comments is also appreciated.

A preliminary version of this article was presented at the 1992 International Test Conference.

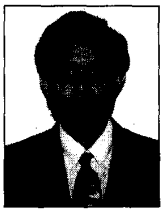
References

1. A. Pancholy, J. Rajski, and L.J. McNaughton, "Empirical Failure Analysis and Validation of Fault Models in CMOS VLSI," *Proc. Int'l Test Conf.*, IEEE Computer Society Press, Los Alamitos, Calif., 1990, pp. 938-947.
2. Koenemann et al., "Delay Test: The Next Frontier for LSSD Test Systems," *Proc. Int'l Test Conf.*, CS Press, 1992, pp. 578-587.
3. J.J. LeBlanc, "LOCST: A Built-In Self-Test Technique," *IEEE Design & Test of Computers*, Vol. 1, No. 4, Nov. 1984, pp. 45-52.
4. *IEEE Standard 1149.1, Standard Test Access Port and Boundary Scan Architecture*, IEEE, Piscataway, N.J., 1990.
5. V.D. Agrawal, S.K. Jain, and D.M. Singer, "Automation in Design for Testability," *Proc. Custom Integrated Circuit Conf.*, CS Press, 1984, pp. 159-163.
6. P. Wilcox, S. Sunter, and N. Mehta, "Circuit for Generating Nonoverlapping Two-Phase Clocks," U.S. patent No. 4,912,340, Apr. 1990.
7. J.A. Waicukauski, V.P. Gupta, and S.T. Patel, "Fault Detection Effectiveness of Weighted Random Patterns," *Proc. Int'l Test Conf.*, CS Press, 1988, pp. 245-255.
8. B.H. Seiss, P.M. Trouborst, and M.H. Schulz, "Test Point Insertion for Scan-Based BIST," *Proc. IEEE European Test Conf.*, CS Press, 1991, pp. 253-262.
9. F. Muradali, V.K. Agarwal, and B. Nadeau-Dostie, "A New Procedure for Weighted Random Built-In Self-Test," *Proc. Int'l Test Conf.*, CS Press, 1990, pp. 660-669.
10. P.H. Bardell, W.H. McAnney, and J. Savir, *Built-In Test for VLSI: Pseudorandom Techniques*, Wiley Interscience Series, New York, 1987.
11. P.D. Hortensius et al., "Cellular Automata-Based Pseudorandom Number Generators for Built-In Self-Test," *IEEE Trans. ICCAD*, Vol. 8, 1989, pp. 842-859.
12. S. Patil and J. Savir, "Skewed-Load Transition Test: Part II, Coverage," *Proc. Int'l Test Conf.*, CS Press, 1992, pp. 714-722.
13. B. Dervisoglu and G. Stong, "Application of Scan-Based Design-For-Test Methodology for Static and Timing Failures in VLSI Components," *Proc. Int'l Conf. VLSI*, 1991, pp. 10.3.1-10.
14. J. Rajski and J. Tyszer, "Fault Detection and Diagnosis Based On Signature Analysis," *Proc. IEEE Int'l Symp. Circuits and Systems*, IEEE, Piscataway, N.J., Vol. 3, 1991, pp.1877-1880.
15. D. Lambidonis et al., "Computation of Exact Fault Coverage for Compact Signature Testing Scheme," *IEEE Int'l Symp. Circuits and Systems*, IEEE, Vol. 3, 1991, pp. 1873-1876.
16. B. Nadeau-Dostie et al., "Scan Design Software for ASICs," *Proc. Canadian Conf. on VLSI*, 1989, pp. 3-8.
17. B. Nadeau-Dostie, P.S. Wilcox, and V.K. Agarwal, "A Scan-Based BIST Technique Using Pair-Wise Compare of Identical Components," *Proc. Fourth CSI/IEEE Int'l Symp. on VLSI Design*, IEEE, 1991, pp. 225-230.
18. M. Schulz and F. Brglez, "Accelerated Transition Fault Simulation," *Proc. 24th ACM/IEEE Design Automation Conf.*, CS Press, 1987, pp. 237-243.



Benoit Nadeau-Dostie is an adviser with Bell-Northern Research Ltd., Ottawa, Canada. His primary research involves the development of new design-for-testability techniques for communication systems. After receiving a PhD in electrical engineering from the Université de Sherbrooke, Québec, he joined the Department of Electrical Engineering of the Université Laval (Vision and Digital Systems Lab). He has served on the program committee of several international conferences and workshops, and is a member of the IEEE and the Ordre des Ingénieurs du Québec.

Dwayne Burek is a member of the scientific staff at Bell-Northern Research Ltd., Ottawa, Canada, where he has been involved in test tools for printed circuit packs, and is currently with the Design for Testability Department. Burek's responsibilities include the development, support, and maintenance of integrated circuit test tools. His interests include built-in self-test of logic circuits, self-test of embedded memories, board and system self-test, and diagnostics. He received an MS in electrical engineering from the University of Manitoba. He is a member of the IEEE Computer Society.



Abu S.M. Hassan is a member of the scientific staff of the Design for Testability Department at Bell-Northern Research Ltd., Ottawa, Canada. His research interests include built-in self-test, automatic test pattern generation of digital circuits, and boundary-scan based testing. He received the BS degree in electrical engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, and the ME and PhD degrees in electrical engineering from McGill University, Montreal, Canada.

Address questions and comments about this article to Benoit Nadeau-Dostie at Bell-Northern Research, MS 049, 185 Corkstown Road, Nepean, Ontario, Canada, K2H 8V4; bnados@bnr.ca.