ScanBist
A Multi-frequency Scan-Based BIST Method

Benoit Nadeau-Dostie, Dwayne Burek, Abu S. M. Hassan

Bell-Northern Research Ltd.
P.O. Box 3511 Station C
Ottawa, Ontario, Canada K1Y 4H7

Abstract

This paper presents a BIST technique that allows the synchronization of multiple scan chains clocked at different frequencies. The technique is used to improve performance testing of scannable circuits. A few new design rules and small modifications to the existing ATPG were necessary to implement the technique.

Introduction

Built-in Self-test (BIST) is gaining popularity as a means to address test issues at the different packaging levels of digital systems. One of the benefits of BIST is the fact that no patterns need to be stored in the test equipment, which is simply required to provide a clock and a few control signals. This is especially important when high-performance systems are being tested. BIST also makes the chip/board/system more independant of the specific test resources available at each manufacturing stage. BIST is also a convenient way of applying more test patterns to compensate for the weaknesses of the stuck-at fault model [Pan90].

Specialized self-test circuits for embedded blocks (memories, PLAs, etc...) have reached a good level of maturity and adequate quality-area-performance trade-offs can usually be made based on the existing methods. However, there is much room for innovation in the field of BIST for unstructured or random logic that needs to be tested for performance given the various design styles.

It is frequent, in telecommunications applications, to design circuits requiring to operate of different frequencies derived from one (or more) reference frequency. For example, communication protocols can be embedded in a hierarchical fashion in a single frame of data and several finite-state machines, working at appropriate sub-frequencies, can be used to extract embedded protocols by simple decimation of the input sequence. Using sub-frequencies derived from a master clock instead of the master clock itself can save power, reduce electromagnetic interference and reduce silicon area.

Scan techniques have been traditionally used to generate very high quality test for faults detectable at low speed. The speed of application of the scan patterns is typically limited to the lowest of the inverse of the longest propagation path delay found on the chip or the frequency of the test clock often not optimized for high speed operation. Delay faults are not adequately tested in such an environment.

In this paper we describe a BIST technique for scan-based designs, called ScanBist, that removes some of the barriers for performance testing in a scan environment. The first section provides an overview of the technique. The BIST hardware, the clocks synchronization, the fault coverage considerations and the software support associated with this technique are addressed in separate sections. Conclusions and results are also presented.

Overview of ScanBist

ScanBist is a BIST technique for scan-based designs that allows synchronization of multiple chains running at different frequencies. The frequencies can be generated by division of a reference (system) clock or driven from primary inputs or both. The technique is a significant improvement over the well-known STUMPS method [Bar82] developed by IBM. The synchronization of the various scan chains is such that the performance of the circuit can be tested in an environment that is as close as possible to the reality. An exhaustive stuck-at and transition fault coverage analysis can be performed using a fault simulator for combinational circuits.

0-8186-3167-8/92 $3.00 © 1992 IEEE

The sub-frequencies derived from the input clock do not need to be multiples of each other. 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 must be divisible by all frequency ratios. Another constraint is that the phase relationship between the sub-frequencies must be fixed during the test.

ScanBist is compatible with various clocking methodologies: edge-triggered, 2-edge, and a flavour of 2-phase developed at BNR. The complete circuit is tested in one pass i.e. there is no scheduling involved. Another advantage for designers already familiar with scan design is that there is little impact on the way the circuit is usually designed.

The BIST technique proposed relies on known methods to achieve extremely high stuck-at fault coverage in a reasonable time by using test point insertion [Sei91] and/or weighted random patterns [Mur90]. Additional precautions must be taken to augment the coverage of transition and delay faults due to the correlation of input patterns inherent to scan designs. More work is required to clearly define the requirements in terms of transition and delay fault coverage and in the synthesis of delay fault testable circuits in order to appreciate the effectiveness of ScanBist.

A block diagram of the ScanBist implementation is shown in figure 1. The IEEE 1149.1 Test Access Port (TAP) is connected to a macro circuit called BIST core. The BIST core is in turn connected to the various scan chains (1 to n). Each scan chain has its own scan-in and scan-out pins. In principle, each scan chain can operate from a separate clock and scan-mode signals. However, in practice, several chains are likely to share the same clock and scan-mode. One of the scan chains is the boundary scan chain that provides isolation of the chip under test from the tester or other chips of the system.

This view of the circuit is valid during random logic self-test mode. The BIST core is designed such that it is itself scannable and the complete circuit (including the BIST core) can be tested with a single scan chain controlled from the TAP. This option provides compatibility with the existing test method in use at BNR. This might be useful if, for example, it is not possible to use the system clock for chip re-testing during board manufacturing. Also, it provides better diagnotic capabilities at the chip manufacturing stage, if needed.

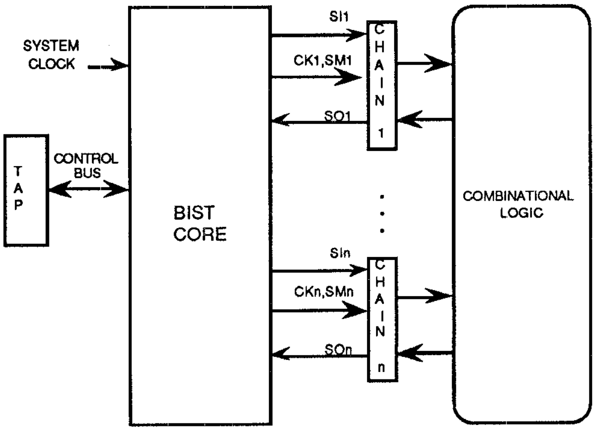

Figure 1: ScanBist implementation overview

The protocol used to invoke ScanBist is relatively simple;

1) Initialize BIST 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 2) if multiple signatures are necessary

Bist core

The BIST core circuit itself (figure 2) consists of a random pattern generator , a signature analyzer, a pattern counter, a scanmode signal generator and a clock generator that can be shared with the system clock generator. All the components of the BIST core work at the frequency imposed by the system clock input during self-test mode. The BIST circuit is synthesized to ensure better portability of the technique from one technology to an other. In the current implementation, the number of scan chains was limited to 16. It seems to be a good compromise between the size of the BIST core and the test length that might be required to test a circuit. This number could be changed in the future and would require the addition and verification of the new polynomials in the scan software.

The random pattern generator is a conventional 24-bit LFSR implementing a primitive polynomial known by the ATPG. 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 same patterns would be reapplied to the circuit.

The signature analyzer is also a conventional MISR based on a primitive polynomial. We decided not to hardwire the signatures because of the extra area required to implement the corresponding comparators, the difficulty of verifying the design and the fact that the design of the BIST core can only be finalized at the very end of the design cycle, thus possibly becoming a bottleneck. Instead, the final signatures are scanned out using the TAP.

The programmable pattern counter indicates when to stop the BIST test. It causes the BIST core to hold its state until the signature and other registers have been read from the TAP. Using more than one maximum count allows for tester time optimization. Aliasing is also reduced significantly. The appropriate times can be computed using the method proposed by [Lam91]. The programmable pattern counter can be eliminated by loading appropriate seeds in the pattern generator and using always the same state as the final one. However, the ATPG program needs an extra pre-processing step to compute the appropriate seeds using the reciprocal polynomial.

A mechanism to compute the number of patterns is absolutely needed if the BIST technique is to be used in a fully configured system where the system clock(s) will run asynchronously relative to the TAP clock. The TAP clock is also likely to be significantly slower than the system clock(s). It is therefore impossible to stop the BIST circuit at the right time from the TAP.

The scan mode signals generator is a counter enabled by the clock generator. The number of clock cycles for one scan pattern (including the capture cycle) must be a multiple of the lowest frequency in the circuit. The output of this counter is gated with the clock generator output to generate the scan mode signals associated with each sub-frequency. It is advisable to make the cycle length of this counter intentionally longer than required to make sure 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. Having the cycle length longer than required ensures that all scan chains have been flushed out with new data and the response from the previous scan pattern has been entered in the signature analyzer. Of course, the price to pay is slightly longer execution time.

The clock generator is preferably a single counter generating sub-frequencies that are powers of two. Non-power of two sub-frequencies require separate counters plus one extra to trigger the scan mode signals generator. The synchronization requirements are explained in more detail in the next section.

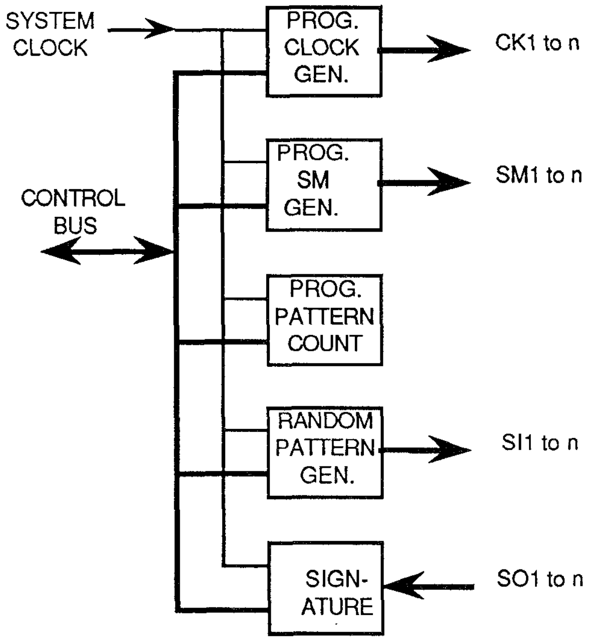

Figure 2: Bist core structure

Clocks synchronization

Two methods have been developed to generate sub-frequencies of the input clock: the "fixed pulse-width" method and the "50% duty cycle" method. Both methods offer different characteristics that make them useful in different contexts. Both methods can be used on the same chip.

The fixed pulse-width method is illustrated in Figure 3. The timing diagram shows the relationship between the control signals of 2 scan chains, one running at half the speed of the other. In this diagram, CK1 and CK2 are respectively the clocks applied to chain 1 and 2. Chain 1 is assumed to work at the system clock frequency. All flip-flops are assumed to update their output on the rising edge of the clock. SM1 and SM2 are the scan mode signals for chain 1 and 2 respectively. These signals are active high. That is, data is shifted in and out a chain when the corresponding scan mode signal is high. The output of the combinational circuit is sampled when the signal is low. This mode is called the scan sample mode.

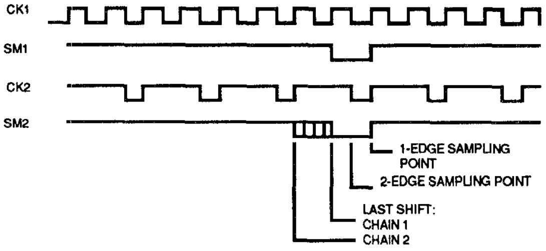

Figure 3: Fixed pulse-width timing

In this example, CK2 is obtained by gating the negative-going pulse of CK1 every 2 clock cycles. Therefore, the resulting clock is not a 50% duty cycle one. This way of generating the clock allows the use of a mixture of clocking techniques at the interface between the 2 frequency "domains", as shown by the position of the sampling points on the timing diagram. This will generally be necessary at the interface with the boundary scan chain, which uses a 2-edge clocking technique. Clock skew between the frequency domains is another reason to use 2-edge or 2-phase timing at the interface. On the other hand, if the system frequency is very high, it might be difficult to distribute the narrow clock pulses throughout the chip. Also, some circuitry on the chip might be sensitive to the non-50% duty cycle (e.g. memories).

The crucial point of this timing diagram is to show that the synchronization between the different chains must be such that during the scan sample mode cycle, all flip-flops sample after all scan chains have been loaded with new data that has had time to propagate through the combinational circuit, but before any flip-flop updates its ouput. This way, the circuit can still be analyzed as a strictly combinational circuit whose inputs have different arrival times for a given test pattern. It is also shown that the scan mode signal for the second chain (SM2) can change any time between the time of the last shift for chain 2 and the next rising edge of CK1. Therefore, SM1 could be used for both scan chains if this is more convenient.

Figure 4 shows a timing diagram for the case where the 50% duty cycle sub-frequency generation method is used for the same frequencies as before. The only difficulty with this method is in the case where a signal captured in a chain clocked by CK2 and using 2-edge timing has a signal originating at a chain clocked by CK1. Invalid data might be captured since the sampling operation on the chain clocked by CK2 coincides approximately (within clock skew limits) with the last shift on chain CK1.

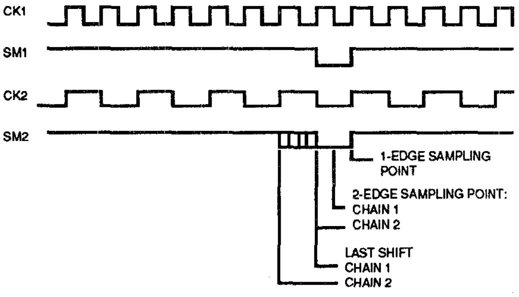

Figure 4: 50% duty cycle timing

In the discussion above, the conclusions concerning 2-edge clocking is also applicable to a flavor of 2-phase clocking developed at BNR and described in [Wil90]. Compatibility with other 2-phase clocking schemes requires more analysis.

The cycle time of the scan mode signals is the same for all scan chains in both the fixed pulse-width and the 50% duty cycle methods. That is, the number of times the various scan chains will sample circuit responses is the same. Only the duty cycle of the scan mode signals might change. The cycle is related to the effective scan chain lengths. The effective scan chain length is defined 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 (system clock).

For the example, suppose that CK1 is clocking 4 flip-flops and that CK2 is clocking only 2. The effective length of the scan chains is L_{eff1}= 4+1 = 5 and L_{eff2}= (2+1) x 2 = 6. It then takes 6 clock cycles to perform a scan cycle (that is scan in/out and capture on all chains). If the number of flip-flops in the first chain is changed to 6, then the effective lengths are L_{eff1}= 7 and L_{eff2}=6. The effective length of the first chain is now larger than that of the second chain and should then impose the length of the scan cycle. However, the scan cycle must be a common multiple of all frequency ratios. Therefore, 8 clock cycles are necessary.

In the case where more than one system clock is present in the circuit, the clock with the highest frequency is selected to generate the sub-frequencies 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 that is as close as possible to the operating frequency without exceeding the constraints imposed by the longest paths in that frequency domain. It is clear that metastability issues due to phase uncertainty between the various system clocks are not accounted for with this technique. However, it should be noted that this constraint exists for most digital testers.

Fault coverage considerations

One of the main reasons to use the self-test approach is to provide a higher coverage of faults of all nature compared to the conventional approach of using stored patterns. However, in our environment, the only objective criteria we have to evaluate self-test fault coverage is the stuck-at and transition fault coverages reported by the ATPG. The strategy adopted here is to maximize the coverage of faults for which we have a metric and use some heuristics to roughly determine the coverage of other faults. For example it is claimed in [Wai88] that applying 30 times as many patterns as required for a stuck-at test gives an excellent coverage of other faults. Similar results were obtained in [Pan90] where each stuck-at fault was detected at least 2 times (most of them 710 times) to generate the reference test set that was detecting more faulty chips than were detected by the single stuck-at fault test set.

The effectiveness of the test might also be significantly increased when the patterns are applied "at speed". That is, at the normal frequency of operation of the circuit. This is another aspect that motivates the use of BIST since only the reference clock needs to be provided to the circuit. It greatly reduces the requirements for sophisticated pin electronics on testers.

A "near-perfect" stuck-at fault coverage is achievable using ScanBist given that the input patterns applied to the combinational part of the circuit are not too strongly correlated. For random pattern resistant circuits, it might be necessary to add test points to improve the controllability and/or observability of the circuit [Sei91]. Weighted-random patterns have also bee shown to be effective [Mur90] [Wai88].

It has been shown in the literature (Brg89] that random pattern generators based on Linear-Feedback Shift-Registers (LFSRs) might not provide good stuck-at fault coverage under certain combinations of polynomials and scan chain ordering. However, these results only hold for very small circuits. In practice, we did not observe a significant impact of this correlation on large circuits given that the cycle length of the LFSR is large compared to the size of the scan chain. When multiple chains are used, bits of different chains are correlated since they are all sharing the same source. This only has an impact if correlated bits are used as inputs to the same logic cone. In this case, several techniques are available to get around this problem: scan chain reordering, LFSR segmentation [Bar87], Cellular Automata random pattern source [Hor90], etc....

In scan-based designs, it is well-known that it is difficult to apply ordered pairs of vectors that lead to detection of stuck-open or delay faults. The correlation between consecutive bits of a scan chain reduces the number of these faults that can be exercised. In order to increase the coverage of transition faults, the only solution is to use an extra latch to hold the output of the flip-flops during shifting. However, this must be done in a way that will minimize the impact on performance and size of the circuit. See [Der91] for an example of an efficient scan cell design.

Using ScanBist, further reduction of delay fault coverage is incurred along paths originating at flip-flops clocked at low frequency and ending at flip-flops clocked at high frequency. After the last shift is performed 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 the normal operation of the circuit, it is possible that the effect of signals originating at the slowest flip-flops must be propagated within the clock period associated with the fastest flip-flops. Paths originating at the fastest flip-flops and ending at the slowest flip-flops are correctly covered within the limitations imposed by the correlation effect explained above.

Another factor potentially reducing the fault coverage is the aliasing associated with the signature analyser. Exact fault simulation results obtained by [Raj91] indicate that, under the single stuck-at fault assumption, the fault coverage is reduced **on average** by 2^{-k} given that the signature analyser is in its steady state. Further analysis is required to determine 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. The impact on the transition fault coverage should also be investigated. On the other hand, the aliasing problem can be eliminated all together by using the multiple signature scheme proposed in [Lam91]. As mentioned before, the same method can be used to optimize tester time.

Clearly, there is more work required to define the requirements in terms of transition fault coverage leading to delay and stuck-open fault detection and in the synthesis of delay fault testable circuits in order to appreciate the effectiveness of ScanBist.

Software support

ScanBist is supported by the BNR scan design system, a modified version of that reported in [Nad89]. In addition to the standard full scan based rules, some additional rules are checked to ensure that

1) the control signals of the BIST core are hooked up correctly to the TAP
2) all flip-flops of a scan chain are driven from the same frequency clock
3) each scan mode signal used with a scan chain is compatible. One may use a scanmode signal from a higher frequency but not from a lower frequency.

The rules checker (Scanchek) also generates 3 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 etc... Most of this information is standard but can be customized if need be. 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 combinational logic is described in the tgfile_bist file. Based on the information contained in these files, the ScanBist core can be synthesized.

The existing ATPG was already able to generate pseudo-random patterns applied through a single scan chain controlled via the TAP, compute the fault coverage and the corresponding signature. This feature is especially useful for board and system level testing [Nad91]. For ScanBist, the sections of the ATPG emulating the random pattern generator and the signature analyzer had to be modified to handle several chains running at different frequencies. Also, some criteria have been defined to determine which input bits must be used by the fault simulator for the initialization pattern. There is, however, no change in the way the fault simulation is done or in the calculation of fault coverage. Those implementations are discussed in the following.

At the most, 16 scan chains are fed from a 24 bit LFSR through 16 taps on the LFSR. These chains are of different lengths in terms of the number of flip-flops connected together. In order to synchronize the operations of these different length scan chains 'dummy scan cells' are introduced in the data structure. Let us take an example of two scan chains, C1 and C2 of the same frequency. Suppose, C1 has 6 FFs and C2 has 2 FFS. Both the chains have to finish loading bits from the LFSR before going into scan sample mode cycle. But, C1 needs 6 clock cycles and C2 needs 2 clock cyles to load the bits. C2 has to wait 4 extra clock cycles before going to scan sample mode cycle. For C2, first 4 bits will go directly into the signature analyzer and the remaining 2 bits will be loaded in the FFs of the chain. For this example, 4 dummy scan cells will be appended at the beginning of the data structure of the input list of C2 as shown in Figure 5. This ensures that 'proper' bits are loaded into the FFs. D1, D2, D3 and D4 are the four dummy cells in the figure. For chains of different lengths, the longest length chain determines the length of the input list of each chain. Dummy bits are inserted 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 scan sample mode.

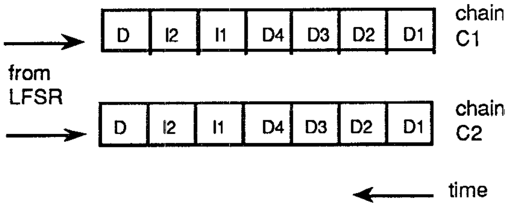

Figure 5: Scan chains alignment (single frequency case)

Similarly, dummy scan cells are inserted between input cells of slower frequency chains in a multi frequency environment. Again, let us take an example of two scan chains C1 and C2. This is shown in Figure 6. C1 has a frequency F1 and C2 has a frequency F2, where F1 is twice as fast as F2. C1 has 6 FFs I1 through I6 and C2 has 2 FFs I and I2. According to the calculation shown in section entitled 'Clock Synchronization', the length of the scan cycle for these two chains is 8. Therefore, chain C1 will have one dummy bit at the beginning, 6 FFs and one dummy bit at the end to represent scan sample mode. Notice that the LFSR always runs at the fastest frequency (in this case, F1). C2 will be loading every second bit from the LFSR (because its frequency is half the frequency of LFSR). This requires insertion of a dummy scan cell after every flip-flop in the chain C2. Chain C2 will have two dummy bits at the beginning. Each of the two FFs will be followed by a dummy cell because its frequency is half of that of the LFSR. Finally, there are two dummy cells at the end of this chain to represent the scan sample mode.

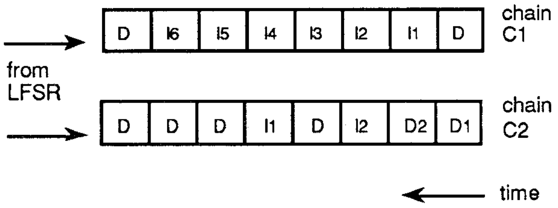

Figure 6: Scan chains alignment (multi-frequency case)

At the output side, the output bits are loaded in parallel from the 16 scan chains into a 16 input MISR. The MISR is also running at the fastest frequency like the LFSR. Dummy scan cells are also introduced in the output data structure of the scan chains to select the 'proper' bits going into the MISR. For multiple chains with the same frequency, the longest length chain determines the length of 'scan shift cycle'. Dummy scan cells are inserted in the shorter scan chains to make all the chains equal. For the multiple frequency environment, frequency and number of flip-flops together determine the effective scan shift cycle. (see section on clocks synchronization). Based on this effective length, dummy scan cells are introduced where necessary. Dummy cells are also introduced between outputs of slower frequency chains and to represent scan sample mode.

Figure 7 shows the sequence of bits going into the MISR for chains C1 and C2 of Figure 6. In Chain C1, output bits O1 through O6 are going into the MISR followed by the dummy input bit. The last input bit I6 of the next vector goes into the MISR during scan sample mode. Chain C2 has half the frequency of C1. Each output bit of C2 should be loaded twice into the MISR. One dummy bit after each output bit takes care of that requirement. These bits are followed by two dummy bits D1 and D2. Finally, I2, the last input bit of next vector is loaded twice into the MISR.

Some of the 16 scan chains can be totally empty. For the empty chains, LFSR taps are directly connected to the MISR taps. Dummy scan cells form these empty scan chains in the data structure.

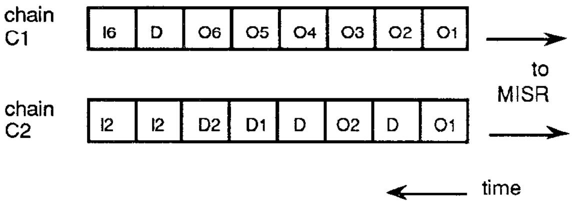

Figure 7: Output sequence for different frequencies

As discussed above, the software implementation makes sure that the proper bits are loaded from the LFSR into the circuit and from the circuit into the MISR. It also ensures that the operations of the different frequency scan chains are synchronized.

Transition fault testing requires so-called 'two-pattern' test. First pattern initializes the node under test to '0'/'1'. Second pattern applies the opposite value ('1'/'0') to test that a '0'-'1' / '1'-'0' transition occurs at the node for the desired clock frequency. Notice that clock speed is very important for transition test. Thus, for multi-frequency ScanBist environment, it is important to determine which node is tested for what frequency. In this implementation, each node fed by multiple frequencies is tested for the fastest frequency only. Example circuit in Figure 8 has 3 different frequencies F, F/2 and F/4. Node D is fed by two inputs B (of frequency F/2) and C (of frequency F/4). D is tested for frequency F/2. Thus, C remains stable and transition values are propagated along B-D. Similarly, node E is fed by two inputs A (frequency F) and D (frequency F/2). E is tested for frequency F only.

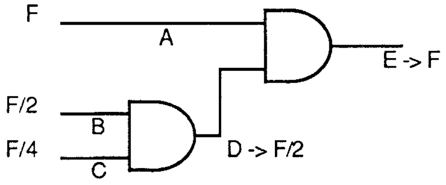

Figure 8: Transition fault testing (multi-frequency case)

Two different types of transition faults are taken into consideration. For simple transition fault model, transition is checked only at the faulty node. Transition is not required to propagate beyond the faulty node. In the second type of fault model, transition is also propagated to the output of the faulty node. This ensures better stuck-open fault detection.

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

It is difficult to compare the performance overhead due to implementation of ScanBist features. This is mainly because of the fact that the circuit under test is not exactly the same when configured as a single scan chain and as a multiple scan chain. 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.

Conclusions and results

The BIST technique was tested on one BiCMOS test chip. The chip works from a 100 MHz clock that is divided down to 50 MHz internally and both clocks are distributed across the chip. About 60,000 patterns are required to achieve 100 % stuck-at fault coverage and 88 % of transition fault coverage. The remaining transition faults are probably due to the stong input correlation. No special precautions were taken to improve the application of ordered pairs of patterns. The BIST core circuit accounts for about 2 % of the chip active area and the test points add an extra 0.5%. More work is required to determine the actual effectiveness of the method.

This paper presents a BIST technique that allows the synchronization of multiple scan chains clocked at different frequencies. The technique is used to improve performance testing of scannable circuits. A few new design rules and small modifications to the existing ATPG were necessary to implement the technique.

Acknowledgements

Stephen Sunter provided the initial motivation to develop this BIST technique. Patricia Birks helped with the *implementation on the first chip.*

References

[Bar82] Bardell P.H., McAnney W. H., "Self-Testing of Multichip Logic Modules", International Test Conference '82", Philadelphia PA, November 1982, pp. 200-204

[Bar87] Bardell P.H., McAnney W. H., Savir J., "Built-In Test for VLSI: pseudorandom techniques", Wiley Inter-Science series, New York, August 1987

[Brg89] Brglez F., Gloster C., Kedem G., "Hardware-Based Weighted Random Pattern Generation for Boundary Scan", International Test Conference '89, Washington D.C., September 1989, pp.264-273

[Der91] Dervisoglu B., Stong G., "Application of scan-based design-for-test methodology for static and timing failures in VLSI components", VLSI '91, Edinburg, Scotland, August 1991, pp. 10.3.1-10

[Hor90] Hortensius P. D., McCleod R. D., Pries W., Miller D.M., Card H.C., "Cellular Automata-Based Pseudorandom Number Generators for Built-In Self-Test", IEEE Transactions on ICCAD, August 1989, vol. 8, pp. 842-859

[Lam91] Lambidonis D., Agarwal V. K., Ivanov A, Xavier D., "Computation of Exact Fault Coverage for Compact Signature Testing Scheme", IEEE International Symposium on Circuits and Systems, Singapore, June 1991, vol. 3, pp. 1873-1876

[Mur90] Muradali F., Agarwal V.K., Nadeau-Dostie B., "A New Procedure for Weighted Random Built-In Self-Test", International Test Conference '90, Washington D.C., 10-12 September 1990, pp. 660-669

[Nad89] Nadeau-Dostie B., McNaughton L., Ermarkaryan A., Burek D., "Scan Design Software for ASICs", Canadian Conference on VLSI '89, Vancouver B.C., 22-24 October 1989, pp.3-8

[Nad91] Nadeau-Dostie B., Wilcox P.S., Agarwal V.K., "A Scan-Based BIST Technique Using Pair-Wise Compare of Identical Components", Fourth CSI/IEEE International Symposium on VLSI Design, New Delhi, India, January 4-8, 1991, pp. 225-230

[Pan90] Pancholy A, Rajski J., McNaughton L. J., "Empirical Failure Analysis and Validation of Fault Models in CMOS VLSI", International Test Conference '90, Washington D.C., 10-12 September 1990, pp. 938-947

[Raj91] Rajski J., Tyszer J., "Fault detection and diagnosis based on signature analysis", IEEE International Symposium on Circuits and Systems '91, Singapore, June 1991, vol. 3, pp.1877-1880

[Sch87] Schulz M., Brglez F., "Accelerated Transition Fault Simulation", 24th ACM/IEEE Design Automation Conference, Miami Fl., June 87, pp. 237-243

[Sei91] Seiss B. H., Trouborst P. M. , Schulz M.H., "Test point Insertion for Scan-based BIST", IEEE European Test Conference, Munich, Germany, 1991, April 10-12, pp.253-262

[Wai88] Waicukauski J. A., Gupta V. P., Patel S.T., "Fault detection Effectiveness of Weighted Random Patterns", International Test Conference '88, Washington D.C., September 1988, pp. 245-255

[Wil90] Wilcox P., Sunter S., Mehta N., "Circuit for generating non-overlapping two-phase clocks", U.S. patent No. 4,912,340 (also filed in Canada), April 1990