A Flexible and Scalable Methodology for GHz-Speed Structural Test

Vikram Iyengar
IBM Microelectronics
Essex Junction, VT 05452
vikrami@us.ibm.com

Gary Grise
IBM Microelectronics
Essex Junction, VT 05452
ggrise@us.ibm.com

Mark Taylor
IBM Microelectronics
Essex Junction, VT 05452
mark1@us.ibm.com

ABSTRACT

At-speed test of integrated circuits is becoming critical to detect subtle delay defects. Simulation-based functional test is difficult because low-cost testers are unable to supply multiple asynchronous clocks to the IC. Moreover, low-cost testers simply cannot operate at chip speed. Existing structural at-speed test methods are inadequate because they are unable to supply sufficiently-varied functional clock sequences to test complex sequential logic. Moreover, they require tight restrictions on the circuit design. In this paper, we present a new method for GHz-speed structural test of ASICs having no tight restrictions on the circuit design. In the present implementation, any complex at-speed functional clock waveform for 16 cycles can be applied. We also describe a method to test asynchronous clock domains simultaneously. Experimental results for two multi-million gate ASICs demonstrate high at-speed coverage.

Categories and Subject Descriptors

B.7.3 [**Integrated Circuits**]: Reliability and Testing

General Terms

Design, reliability

Keywords

ASICs, asynchronous clock domains, at-speed, deskewer, structural test, test waveform generator

1. INTRODUCTION

At-speed test for high-functionality integrated circuits (ICs) is important to reduce shipped product quality level (SPQL). SPQL is defined as the fraction of ICs shipped by the manufacturer to the customer that are identified as being defective by the customer during system test. SPQL directly impacts product quality, cost and the IC manufacturer's reputation. Therefore, achieving low SPQL is critical to the IC manufacturing business.

All ICs undergo manufacturing test prior to being shipped to customers. Scan-based test for stuck-at faults has been well studied in the literature [8] and is applied extensively in the industry with 99% fault coverage levels [1]. SPQL defects that escape manufacturing test are therefore more often related to design errors, process variations or subtle delay defects. Testing ICs at-speed to uncover small delay defects is therefore the fastest way to get to the root of SPQL.

Implementing at-speed functional test using simulation waveforms has been proposed to recreate the customers' environment on automatic test equipment (ATE) [11, 12]. However, high-functionality ICs having asynchronous clock domains require expensive ATE that can provide multiple asynchronous clocks during test. Additionally, functional test requires ATE to provide test data at-speed to IC pads. This is simply beyond the scope of low-cost ATE. Furthermore, no accurate functional fault coverage metrics exist.

At-speed manufacturing test based on automatic test pattern generation (ATPG) is therefore becoming popular as a means to exercise circuit logic at-speed with high fault coverage [2, 7]. The transition fault model is used to generate delay fault test patterns for the IC. These are applied at-speed to detect delay defects that cause slow transitions on internal logic. Since ATE cannot pulse capture clocks on internal flip-flops at functional speeds, the IC's functional clocks are used for test response capture [2]. However, existing methods for at-speed delay test are inadequate since they are unable to supply sufficiently varied functional clock sequences to test subtle delay defects in complex sequential logic. Moreover, existing at-speed test methods require tight restrictions on the circuit design.

In this paper, we present a new methodology for at-speed structural test (ASST) using on-chip clock generation in the GHz range. The novel contributions of this work are as follows.

1. The proposed methodology is highly scalable to any number of logic domains-under-test. Any number of on-chip phase-locked loops (PLLs) can be accomodated.
2. PLL frequency programming is separated from test waveform generation and pattern application; hence, PLLs can be easily reprogrammed to run faster than at-speed.
3. In the present implementation, *any* functional clock waveform for 16 clock cycles can be derived from the PLL clock to be applied to different clock domains. This provides tremendous flexibility in testing complex sequential logic.

DAC 2006, July 24–28, 2006, San Francisco, California, USA.

4. The method is applicable to a wide variety of application-specific integrated circuits (ASICs) having no restrictions on the circuit design.
5. The clock domain-under-test can be changed from test pattern to test pattern without the need to change any control setup signals, thus providing significant flexibility in test application.
6. We also present a method to test multiple asynchronous clock domains simultaneously to reduce test data volume and testing time.

This paper is organized as follows. In Section 2, we discuss prior work in transition fault test and on-chip clocking. In Section 3, we present the proposed test architecture for ASST. In Section 4, we describe test generation for ASST, including timing waveforms for pattern launch and capture. We also present a method to test multiple asynchronous clock domains in a single test pattern. Finally, in Section 5, we present preliminary experimental results on fault coverage for two multi-million gate ASICs.

2. PRIOR WORK

The transition fault model was created to model delays on internal nets. In this fault model, a slow-to-rise and slow-to-fall fault is modeled on every net in the circuit [9]. A test for a transition fault consists of i) a pattern that sets an initial value on the net-under-test, ii) a clock pulse that launches the opposite value onto the net, and iii) a clock pulse that captures the transition into a latch. Several methods to generate transition fault tests have been reported in the literature [6, 10].

Transition fault test application is being widely carried out in industry [2, 4, 5, 7, 13]. In [2], methods for at-speed built-in self test (BIST) and deterministic test are presented. The methods achieve high fault coverage, however, they require strict restrictions on the circuit design, including no multicycle paths and no X-states. Moreover, only 5 functional clock pulses can be applied.

In [4], the authors describe a method to perform delay test for general level-sensitive scan (LSSD) ASIC designs. Figure 1(a) illustrates an LSSD latch-pair; the A and B clocks are used for scan shift and the C clock is the system capture clock. As shown in Figure 1(b), a skewed-load is used to launch a transition off-scan and capture delay test responses in the L1 latches. Because LSSD clocks supplied by the ATE are used for launch and capture, the method is limited by the ATE's minimum clock pulse widths and signal accuracy. Moreover, functional clocking is not used; therefore, setup and hold design errors cannot be detected and correct functional operation of the tested circuit is not guaranteed.

In [5], the authors report 99% fault coverage using only at-speed BIST to test microprocessors. However, because of the test methodology, deterministic patterns (for hard-to-test faults) cannot be run. Moreover, strict design rules concerning multicycle paths and the number of asynchronous domains allowed must be followed. Such design restrictions are often unacceptable to general ASIC customers.

In [7], a method for slow-scan fast-capture clocking is described. The method is applied to a test chip having only two clock domains. Moreover, very tight timing on the scan enable pin is required. Finally, in [13], techniques to increase the efficiency of broadside transition test generation are investigated. While fault coverage is significantly increased, sufficiently-varied clock waveforms such as single edges cannot be produced.

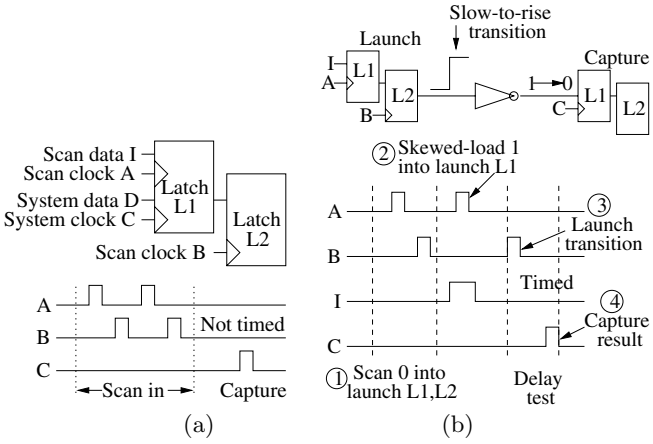

Figure 1: Illustration of (a) LSSD latch-pair, and (b) delay test using skewed-load sequence.

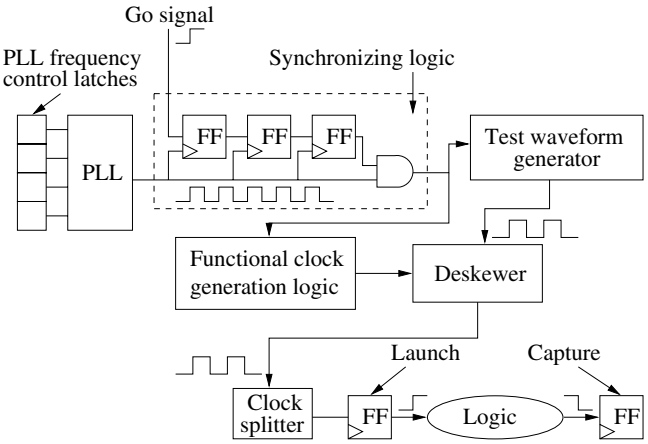

Figure 2: Illustration of ASST methodology.

3. AT-SPEED STRUCTURAL TEST

In this paper, we describe an at-speed structural test methodology for LSSD designs illustrated in Figure 2. The methodology is based on the use of transition test patterns created by an ATPG tool. These patterns are scanned into the latches of each clock domain using slow-speed tester clocks to reduce power consumption and eliminate clock races during scan. On-chip PLLs that generate high-speed functional clocks are up and running throughout the duration of the test. An asynchronous GO signal coming in from a tester pin is used to activate the on-chip clock generation logic to launch and capture each test pattern at-speed. A special test waveform generator (TWG) circuit fed by a PLL creates specific functional clock waveforms for each test pattern. The clock waveforms are applied to the functional clock tree by a deskewer circuit. Finally, clock splitters at the leaves of the functional clock tree receive the clock pulses and activate launch and capture latches to apply the test at-speed. Once the at-speed results have been captured, they can be scanned out at slow speed.

Figure 3: Functional clock waveforms.

3.1 Test modes

ASST test generation is performed in two test modes: parent mode and child mode. In the parent mode, every latch in the circuit-under-test (CUT) is scannable. In this mode, the PLLs are not locked and LSSD test clocks are used to control logic. The parent mode is used to scan in the values for the PLL frequency control latches and to set up the ASST control logic for correct operation. A small number of high-speed scan chain delay test patterns are also applied in the parent mode using the method presented in [4] to detect delay defects in scan chains that are not designed to be tested at functional speed. Once the control values have been scanned in to latches in the parent mode, the CUT enters the child mode, in which the PLL frequency control latches and other control logic is bypassed and not allowed to capture (illustrated later in Figure 6). This ensures that the PLLs are not disturbed and can continue to pulse while each ASST test pattern is scanned into the logic and applied at-speed. At the end of test application in the child mode, the CUT can be once again put into the parent mode for clean-up transition test patterns to be applied using tester clocks [4]. The clean-up patterns will test the remaining untested faults.

3.2 Test waveform generator

Once the CUT is in child mode and the PLLs are operational, a GO signal is asserted by the tester to begin an at-speed functional clock sequence. Since the GO signal is asynchronous and transitions at the slow tester speed, synchronizing latches (Figure 2) are used to ensure that the high-speed functional clock from the PLL to the TWG does not glitch. The present implementation of the TWG consists of two 16-bit shift registers (SRs) that can be programmed to create any arbitrary functional clock waveform for 16 clock cycles; see Figure 3. Note that any number of cycles can be produced by enhancing the TWG design to increase the SR length. Each SR controls a single phase of the functional clock as shown in Figure 4. When the functional clock is low the contents of SR1 determine the output of the TWG, and when the functional clock is high the contents of SR2 determine the output. These two output levels are then combined to create the final functional clock waveform using a special circuit known as a deskewer.

3.3 Deskewer

The structure of the deskewer is illustrated in Figure 4. The functional clock generation logic block shown is a finite-state machine used in mission mode. This logic is bypassed for ASST to prevent adding delay on the mission mode clock path. In child mode, the *ASST* signal is used to select the TWG waveform values for test. The *ASST* signal is a special test pin that goes high in the child mode and is used to put the CUT into the at-speed test state. As soon as the GO signal is asserted in child mode, the synchronizing latches (of Figure 2) will go to 1 after 3 PLL clock cycles. The PLL clock is then observed at the TWG latches. Values that were scanned into the TWG latches during scan in of the test

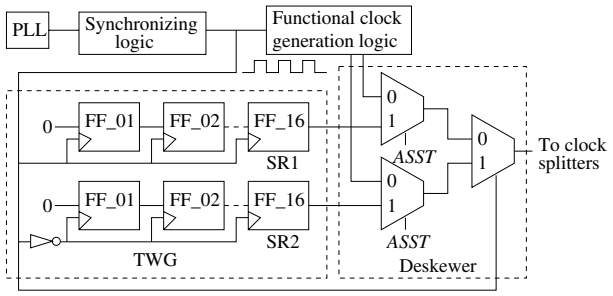

Figure 4: Illustration of TWG and deskewer

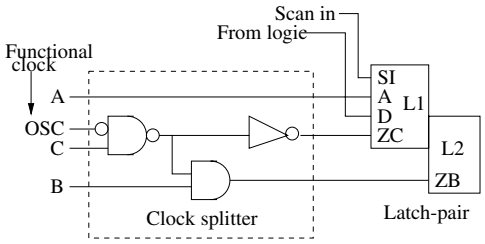

Figure 5: Illustration of clock splitter.

pattern now shift towards the deskewer. At the deskewer, they are used to create test waveforms on the functional clock tree to downstream clock splitters. After 16 PLL clock cycles,the TWG latches zero out, and the GO signal can be de-asserted.

3.4 Clock splitter

In LSSD designs, a clock splitter (see Figure 5) is used to create two non-overlapping clock pulses at the ZC and ZB clock inputs of the L1 and L2 latches of a flip-flop. During scan, the C clock is held low, and alternate A and B pulses are used (Figure 1(a)). In mission mode, the B and C inputs to the clock splitter are held high, thereby transforming the L1–L2 latch-pair from a level-sensitive latch-pair to an edge sensitive flip-flop controlled by OSC. In the ASST child mode, the B and C inputs to each splitter are therefore turned high to allow the at-speed test pulses on the functional OSC clock to launch and capture at latches.

As mentioned earlier, ASST control latches, such as PLL frequency control bits, are scanned with control values in the parent test mode, and then bypassed in the child mode. This is done so that subsequent test patterns when scanned in will not disturb the PLL. This is achieved using muxes controlled by *ASST* as shown in Figure 6. Furthermore, certain latches are not allowed to capture functionally during child mode. This can be due to several reasons: (i) PLL and ASST control latches are not allowed to be disturbed, (ii) slow logic such as boundary scan is not tested at-speed, and (iii) certain areas of the chip that receive their functional clocks from high-speed cores instead of PLLs are shut off. Therefore, capture to these latches is shut-off, but they can still be scanned in the child mode as shown in Figure 7. The *ASST* signal is used to gate off the C clock to the splitters. A combination of *ASST* and the scan enable (SE) signal is used to gate off the B clock to the same splitters.

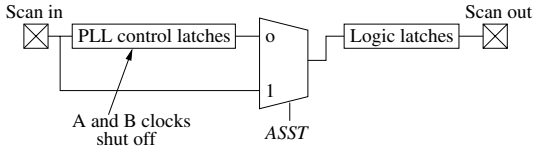

Figure 6: Illustration of PLL control latches.

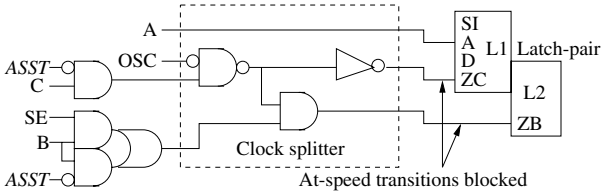

Figure 7: Latches that are shut off for ASST.

4. TEST GENERATION

In this section, we describe clocking methods for ASST. We also present the test generation procedure used.

4.1 Launch-off-capture clocking

The clock splitter used in conjunction with a PLL and deskewer is used to provide at-speed launch-off capture (LOC) clock pulses to latches. Data is first scanned into latches, and the C and B clocks are asserted so that the CUT enters the at-speed state; see Figure 8. As the C clock is asserted, L1 latches in the logic receive a broadside load that sets up the CUT for a LOC sequence. The at-speed pulses from the deskewer then launch and capture test data.

4.2 Test sequence generation

Here, we describe the procedure required to generate an ASST test suite. The procedure is presented in the form of pseudocode in Figure 9. Since timing relationships between clock edges belonging to asynchronous PLLs cannot be predicted, test of multiple clock domains becomes an issue. To address this issue, at first we test only one clock domain per test pattern. Later, we describe a method to test multiple asynchronous clock domains simultaneously.

The procedure begins in the parent mode in Line 1. LSSD clocks are set to 0 to allow for circuit stability. Test control pins are set to required values to begin high-speed scan chain delay test [4].

After scan chain delay test patterns using LSSD clocks have been generated in Line 2, the CUT is scanned with

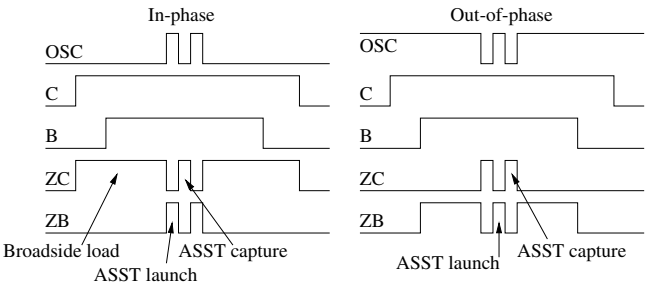

Figure 8: Launch-off-capture test sequences for in-phase and out-of-phase latches.

Procedure *ASST_test_generation*()

Parent mode
1 Set A, B, and C clocks to 0 for stability; set delay test control pins to required values;
2 Generate high-speed scan chain delay test patterns using LSSD clocks;
3 Scan in PLL frequency control values;
Child mode
4 Set *ASST* to 1; /*Enter child mode*/
5 Pulse reference clocks to bring up PLLs to required frequencies;
6 **For** each clock domain in the CUT
7 **While** test pattern budget has not been reached
8 Set SE to 1; scan in functional clock control values into TWGs;
9 Scan in test data values into CUT;
9 Set SE to 0, C to 1, and B to 1;/*Enter at-speed state*/
10 Assert GO signal; /*Begin at-speed test*/
/*Synchronizing latches, TWG and deskewer pulse*/
/*At-speed transitions in domain-under-test*/
11 De-assert GO signal; /*End at-speed test*/
12 Set B to 0, C to 0, and SE to 1;/*Exit at-speed state*/
13 Scan out test responses;
14 Generate clean-up delay tests [4] and stuck-at fault patterns;
15 **Return**.

Figure 9: Test generation procedure for ASST.

PLL frequency control values in Line 3. The CUT is now ready to enter the child mode.

In Line 4, the *ASST* signal is asserted to put the CUT in child test mode and to bypass the PLL latches (Figure 6). From Line 6, the procedure loops through all the clock domains in the CUT. For each clock domain, a separate clocking sequence is used to activate only the deskewer feeding that particular domain. The proposed method thus provides tremendous flexibility, since the domain-under-test can be changed from test pattern to test pattern without the need to change any control set up.

In Line 8, we begin the test generation procedure for a single test pattern. SE is asserted and the test pattern, including TWG values (for the domain-under-test) is scanned into the CUT. All 0s are scanned into the remaining TWGs to hold their deskewers at stability during the at-speed sequence. In Line 9, we de-assert SE, assert C, and assert B to enter the at-speed state. The GO signal is then asserted to begin on-chip clocking. As the synchronizing latches go high, the PLL pulses are allowed through to the TWG. At this point, functional clock pulses appear at the output of the deskewer and begin going through clock splitters to latches. Transitions are launched and captured at latches in the domain-under-test until the TWG zeroes out. The GO is then de-asserted in Line 11. Finally, B and C are de-asserted and SE is asserted to begin scan out of the test responses. In Line 14, delay tests [4] and stuck-at fault patterns are generated for any remaining untested faults.

Here, only the latches belonging to the domain-under-test have launched and captured test data at-speed. The data scanned into latches of other domains are scanned out as is. Hence, at most one clock domain is tested per test pattern.

4.3 Wrapper for asynchronous clock domains

Here, we propose a wrapper for asynchronous domain crossings that allows us to test multiple asynchronous domains in the same test pattern. This significantly reduces test data volume and testing time. Prior work on wrapper design for asynchronous clock domains was presented in [14].

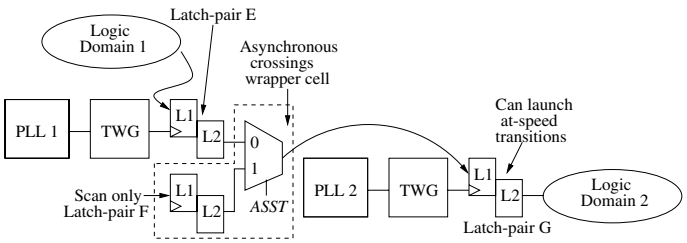

Figure 10: Wrapper for asynchronous domain crossings

Circuit information for Chip K			
No. of primary inputs	77	No. of latches	668K
No. of primary outputs	38	No. of clock domains	7
No. of bidirectional IOs	573	No. of stuck-at faults	15.2 M
No. of library cells	2785K	No. of transition faults	16.2 M
No. of RAMs	161	No. of PLLs	2
No. of scan chains	32		
Circuit information for Chip S			
No. of primary inputs	46	No. of latches	1M
No. of primary outputs	1	No. of clock domains	6
No. of bidirectional IOs	475	No. of stuck-at faults	24.2 M
No. of library cells	4716K	No. of transition faults	26.6 M
No. of RAMs	8.8K	No. of PLLs	3
No. of scan chains	64		

Table 1: Description of ASICs Chip K and Chip S.

The proposed wrapper for asynchronous domain crossings is illustrated in Figure 10. A wrapper cell consists of a multiplexer and a single flip-flop (Latch-pair F). Test responses for Domain 1 are captured at-speed in Latch-pair E. Test data for Domain 2 can be simultaneously launched at-speed from Latch-pair G. A wrapper cell is situated at the data input of every latch-pair in the design that has a latch belonging to a different clock domain in its input cone. The wrapper cells prevent unpredictable test responses from being captured at-speed in asynchronous domains.

5. EXPERIMENTAL RESULTS

In this section, we present experimental results for two multi-million gate ASICs called Chip K and Chip S. A description of Chip K and Chip S are provided in Table 1. Chip K has 10 asynchronous clock domains; of these, 7 domains are tested using ASST. The other domains are either not designed to operate at GHz-speeds or driven by cores that are shut down for ASST. Chip S has 19 asynchronous clock domains; of these, only 6 are tested using ASST. The other domains are either too small to consider or not designed for GHz-speeds, e.g., boundary scan logic. The test architecture for Chip S includes an on-chip MISR for response compression [3]. Hence, 64 scan chains are implemented.

In Figure 11, we present results on fault coverage for Chip K using the test generation method presented in Figure 9. Only one asynchronous domain is tested at a time. Before the at-speed test patterns are applied, we generate scan chain delay test patterns that achieve 49.29% stuck-at and 26.12% transition fault coverage. All fault coverage numbers in this paper are reported for the entire chip, i,e., for Chip K, 16.2 million faults are used as the denominator in the fault coverage expression for transition faults.

After the scan chain test, 1024 at-speed test patterns are applied per clock domain. The total fault coverage at the

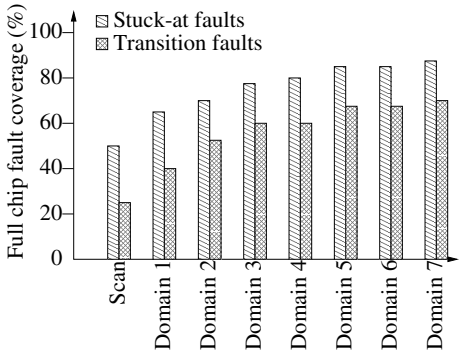

Figure 11: Chip K testing one domain at a time.

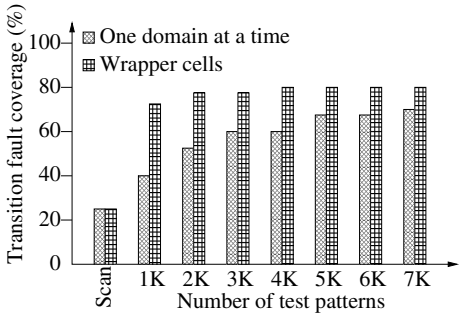

Figure 12: Chip K testing multiple domains simulataneously.

end of the 7K at-speed test patterns is 87.11% for stuck-at faults and 70.22% for transition faults. The stuck-at fault coverage rises much faster than transition fault coverage. This is because the broadside load helps to detect stuck-at faults even in domains in which no at-speed test patterns are applied.

The ATPG experiments in this paper are for the purpose of validating our methodology, and not for production test. Hence, only 1024 test patterns are applied per domain.

In Figure 12, we report fault coverage results testing more than one asynchronous domain per test pattern using the wrapper presented in Section 4.3. These results were obtained by ATPG simulation; the wrapper has not been implemented in hardware. As the number of test patterns applied increases, the fault coverage rises significantly faster than in Figure 11. Most of the fault coverage is reached after 3K at-speed test patterns; after this, only hard-to-test faults remain. After 7K test patterns have been generated, the total fault coverage is 92.59% for stuck-at faults and 81.07% for transition faults. A significant reduction in test data volume and testing time has been realized by testing more than one domain per test pattern.

In Figure 13, we present results on fault coverage for Chip S testing only one asynchronous domain at a time. First, we generate 27 scan chain delay test patterns that achieve 45.22% stuck-at and 26.69% transition fault coverage. After the scan chain test, 1024 at-speed test patterns are applied per clock domain. The total fault coverage at the end of the 6K at-speed test patterns is 91.24% for stuck-at faults and 67.18% for transition faults.

The fault coverage here is low because these experiments are only for the purpose of validating our methodology. In

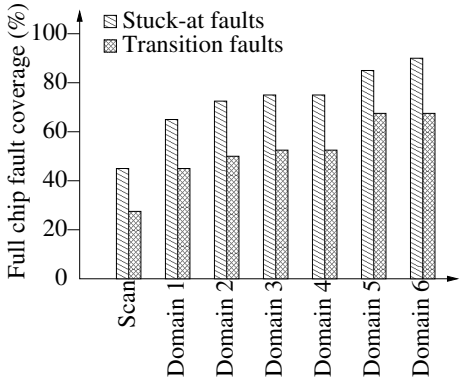

Figure 13: Chip S testing one domain at a time.

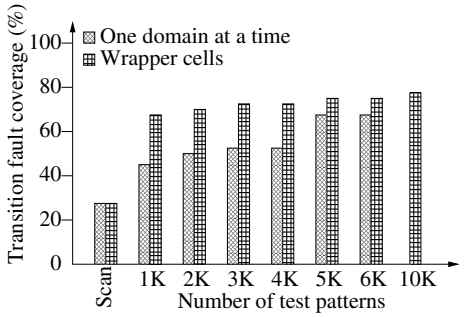

Figure 14: Chip S testing all domains simultaneously.

a complete ASST test suite, 25K test patterns each would be generated for ASICs as large as Chip K and Chip S.

Finally, in Figure 14, we report results for Chip S testing more than one asynchronous domain per test pattern. Again, these results were obtained by ATPG simulation; the wrapper has not been implemented in hardware. After 6K test patterns have been generated, the transition fault coverage is 74.54%. We continued ATPG until a total of 10K ASST patterns are generated. After 10K test patterns have been generated, the transition fault coverage is 76.47%.

Note that the fault coverage numbers reported here are for preliminary experiments. In a complete ASST test suite, 25K test patterns would be generated for ASICs as large as Chip K and Chip S. In future work, we are investigating methods to reduce test pattern count for ASST using improved scan architectures. We are also enhancing our clocking methodology by adding launch-off-scan capability.

6. CONCLUSIONS

We have described a novel GHz-speed structural test methodology for LSSD designs. This methodology is targeted at subtle delay defects, especially on functional clock trees. Functional design errors and manufacturing process variations that cannot be detected using tester clocks have a greater chance of being detected using the proposed method. We have presented DFT logic to generate varied functional clock waveforms to test complex sequential logic. Finally, a new method to test multiple asynchronous clock domains simultaneously has been described. This achieves a significant reduction in test data volume over testing one domain at a time.

7. ACKNOWLEDGEMENTS

We are grateful to Steve Oakland, David Lackey, Phil Nigh, Theo Anemikos, Mark Johnson, Phil Stevens, Frank Woytowich, and Mike Degregorio for their help with the ASST methodology. We thank Chandu Visweswariah for his help with the final version of this paper.

8. ADDITIONAL AUTHORS

Additional authors: Bob Bassett, Rudy Farmer.

9. REFERENCES

[1] ASICs Test Methodology. IBM Microelectronics. Essex Junction, VT 05452.

[2] B. Bailey et al. Test methodology for Motorola's high-performance e500 core based on PowerPC instruction set architecture. *Proc. Int. Test Conf.*, pp. 574–583, 2002.

[3] C. Barnhart et al. OPMISR: The foundation for compressed ATPG vectors. *Proc. Int. Test Conf.*, pp. 748–757, 2001.

[4] P. Gillis et al. Low overhead delay testing of ASICs. *Proc. Int. Test Conf.*, pp. 534–542, 2004.

[5] M. Kusko et al. 99% AC test coverage using only LBIST on the 1GHz IBM S/390 zSeries 900 microprocessor. *Proc. Int. Test Conf.*, pp. 586–592, 2001.

[6] X. Liu et al. Novel ATPG algorithms for transition faults. *Proc. European Test Workshop*, pp. 47–52, 2002.

[7] T.L. McLaurin and F. Frederick. The testability features of the MCF5407 containing the 4th generation ColdFire microprocessor core. *Proc. Int. Test Conf.*, pp. 151–159, 2000.

[8] J.H. Patel. Stuck-at fault: A fault model for the next millennium. *Proc. IEEE Int. Test Conf.*, pp. 1166, 1998.

[9] J. Savir and S. Patil. Scan-based transition test. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 12, pp. 1232–1241, Aug 1993.

[10] Y. Shao, I. Pomeranz, and S.M. Reddy. On generating high quality tests for transition faults. *Proc. Asian Test Symposium*, pp. 1–8, 2002.

[11] C. Stolicny et al. Manufacturing pattern development for the Alpha 21164 microprocessor. *Proc. IEEE Int. Test Conf.*, pp. 278–285, 1997.

[12] T. Taylor. Tools and techniques for converting simulation models into test patterns. *Proc. IEEE Int. Test Conf.*, pp. 133–138, 1993.

[13] N. Tendolkar et al. Novel techniques for achieving high at-speed transition fault test coverage for Motorola's microprocessors based on PowerPC^{TM} instruction set architecture. *Proc. IEEE VLSI Test Symp.*, pp. 3–8, 2002.

[14] Q. Xu and N. Nicolici. Wrapper design for testing IP cores with multiple clock domains. *Proc. Design, Automation and Test in Europe Conf.*, pp. 416–421, 2004.