FAST-BIST: Faster-than-At-Speed BIST Targeting Hidden Delay Defects

Hellebrand, Sybille; Indlekofer, Thomas; Kampmann, Matthias; Kochte, Michael A.; Liu, Chang; Wunderlich, Hans-Joachim

Proceedings of the IEEE International Test Conference (ITC'14) Seattle, Washington, USA, 20-23 October 2014

doi: http://dx.doi.org/10.1109/TEST.2014.7035360

Abstract: Small delay faults may be an indicator of a reliability threat, even if they do not affect the system functionality yet. In recent years, Faster-than-at-Speed-Test (FAST) has become a feasible method to detect faults, which are hidden by the timing slack or by long critical paths in the combinational logic. FAST poses severe challenges to the automatic test equipment with respect to timing, performance, and resolution. In this paper, it is shown how logic built-in self-test (BIST) or embedded deterministic test can be used for an efficient FAST application. Running BIST just at a higher frequency is not an option, as outputs of long paths will receive undefined values due to set time violations and destroy the content of the signature registers. Instead, for a given test pattern sequence, faults are classified according to the optimal detection frequency. For each class, a MISR-based compaction scheme is adapted, such that the critical bits to be observed can be determined by algebraic computations. Experiments show that rather a small number of inter-mediate signatures have to be evaluated to observe a large fraction of hidden delay faults testable by the given test sequence.

Preprint

General Copyright Notice

This article may be used for research, teaching and private study purposes. Any substantial or systematic reproduction, re-distribution, re-selling, loan or sub-licensing, systematic supply or distribution in any form to anyone is expressly forbidden.

This is the author's "personal copy" of the final, accepted version of the paper published by IEEE.[1]

[1] **IEEE COPYRIGHT NOTICE**

©2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

FAST-BIST: Faster-than-At-Speed BIST Targeting Hidden Delay Defects

Sybille Hellebrand[1], Thomas Indlekofer[1], Matthias Kampmann[1], Michael A. Kochte[2], Chang Liu[2], Hans-Joachim Wunderlich[2]
[1]University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
[2]University of Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart, Germany

Abstract

Small delay faults may be an indicator of a reliability threat, even if they do not affect the system functionality yet. In recent years, Faster-than-at-Speed-Test (FAST) has become a feasible method to detect faults, which are hidden by the timing slack or by long critical paths in the combinational logic. FAST poses severe challenges to the automatic test equipment with respect to timing, performance, and resolution.

In this paper, it is shown how logic built-in self-test (BIST) or embedded deterministic test can be used for an efficient FAST application. Running BIST just at a higher frequency is not an option, as outputs of long paths will receive undefined values due to set time violations and destroy the content of the signature registers. Instead, for a given test pattern sequence, faults are classified according to the optimal detection frequency. For each class, a MISR-based compaction scheme is adapted, such that the critical bits to be observed can be determined by algebraic computations. Experiments show that rather a small number of intermediate signatures have to be evaluated to observe a large fraction of hidden delay faults testable by the given test sequence.

1 Introduction

Certain delay defects are not detectable by standard transition delay and not even by timing-aware transition delay test patterns, since they are only propagated along short paths where the slack is larger than the defect size. Even if such *hidden delay defects* do not violate nominal timing, they point to marginal hardware and imperfections that can further degrade after a short operation period and cause early life failures (ELF). For example, gate-oxide defects can manifest themselves as delay faults before actual hard failures occur [Kim10, Malandruccolo11]. Detection of hidden delay defects can be achieved by testing with a test clock frequency higher than the nominal at-speed frequency, e.g. [Yan03]. This is called *Faster-than-At-Speed Testing* (FAST).

Usually, several different frequencies are used for FAST. In some sense this may look similar to speed binning [Cory03, Zeng04], but the problem here is fundamentally different. While speed binning addresses the critical paths to check the functionality of the circuit, here the short paths are addressed with frequencies above the intended system frequency. This may require a more costly high speed automatic test equipment (ATE). In addition to that, external FAST is particularly challenging, as the measurements are affected by tester skews, parasitic capacitances, and other electrical effects [Tayade08]. Built-in FAST using programmable schemes for on-chip clock generation overcomes these problems [Tayade08, Pei10]. However, if the test responses are captured at higher frequencies, the response values may not arrive in time at the outputs driven by longer paths. Because of potential hazards the old values may already be overwritten, such that test response evaluation has to deal with unknown values (X-values).

X-values corrupt the results of test response compaction, and a whole range of X-handling schemes have been developed to cope with this problem in BIST or embedded test [Mitra02, Naruse03, Rajski03, Sharma05, Tang06, Touba07 Wohl07]. Nevertheless, simply reusing standard solutions for X-handling will not work for built-in FAST or lead to suboptimal results for the following reasons. Firstly, the distribution of X-values depends on the distribution of long paths and changes with the test frequencies, and even for a fixed frequency the distribution may vary due to parameter variations. Secondly, even the set of hidden defects and the required test patterns may change because of parameter variations. The applied X-handling scheme must therefore be flexible and independent of the test set. Finally, the X-values will be clustered at outputs driven by many long paths, and only a few schemes take into account specific clustering of X-values [Czysz10].

In this paper, a scheme for built-in FAST is presented which relies on a programmable solution for X-handling and is compatible with the widely accepted STUMPS architecture [Bardell82]. It combines the X-canceling MISR proposed in [Touba07] with small on-chip storage for intermediate signatures and off-chip post-processing. For test pattern generation a state of the art mixed-mode approach [Hakmi07] or embedded deterministic test can be used [Rajski04]. The scheme is supported by appropriate pre-processing routines to determine the hidden delay defects and their optimal detection frequencies as well as to minimize the information needed for fault detection. It

should be noted that the focus here is not on ATPG, and pre-processing can be performed with any given test set targeting delay faults. If a high quality timing aware test set is used [Lin06], the number of hidden delay defects will decrease and thus the efficiency of the developed scheme will increase.

The rest of this paper is organized as follows. After presenting related work in Section 2, an overview of the developed FAST-BIST scheme is given in Section 3. Subsequently, the algorithms for pre-processing are described in Section 4, and the details of test response compaction and X-handling follow in Section 5. Finally, the experimental results in Section 6 demonstrate that a high coverage of hidden delay defects can be achieved with only moderate hardware overhead.

2 Related Work

Small delay defects have become the focus of intensive research in recent years. As a complete review of the state of the art is beyond the scope of this paper, the reader is referred to [Tehranipoor12] as an introductory overview. Small delay defects, which are usually modeled as extra gate delays, can only be detected, if they can be observed at the end of long paths with a slack smaller than the delay size. Timing aware ATPG for small delay defects therefore favors long paths over short paths for fault propagation [Lin06]. Some recent approaches try to generate "robust" and hazard-free test patterns [Zolotov10, Eggersgluess11, Sauer13], while other approaches start with a low cost n-detect ATPG procedure and then select the best patterns based on specifically developed metrics [Yilmaz10].

In FAST, the test patterns are applied at an increased clock-rate to detect also small delay faults on short paths [Yan03]. The frequencies reported for practical applications are typically up to 3 times higher than the nominal clock rate [Amodeo05, Ahmed06, Lee08]. Special ATPG frameworks and procedures have been developed to support FAST, which choose for example test patterns with hazard-free fault propagation or select paths with switching times in certain clock intervals [Ahmed06, Krusemann04, Fu12]. In [Yoneda11], sensitized paths in a given test set are extracted to determine the maximum detectable defect size. Then, patterns are duplicated and reordered to increase defect coverage with as few as possible FAST timing groups. As already mentioned in the introduction, external FAST can be costly and is particularly challenging for the ATE. This can be overcome by a built-in solution with programmable on-chip clock generation [McLaurin00, Press06, Tayade08, Pei10].

To deal with an extremely high (potentially unbounded) number of X-values during FAST, Singh has proposed a special MUX-based compaction scheme [Singh10]. This solution, however, requires a significant amount of control data during test application and discards a large fraction of response data by the multiplexers. As explained above, standard X-handling schemes for BIST or embedded test cannot be simply re-used, however they can be the basis for test response compaction in FAST-BIST. Some typical examples for different X-handling strategies can be found in [Mitra02, Naruse03, Rajski03, Sharma05, Tang06, Touba07 Wohl07]. X-masking schemes mask out the X-values before they can enter the compactor [Naruse03, Tang06, Wohl07] while X-filtering or X-canceling strategies can extract uncorrupted information after compaction [Sharma05, Touba07]. X-tolerant compaction schemes like X-compact or convolutional compactors can tolerate a certain amount of X-values without additional measures [Mitra02, Rajski03, Mitra04].

3 FAST-BIST – Overview

This section gives an overview of the developed FAST-BIST strategy. It is based on a mixed-mode BIST combining pseudo-random patterns with deterministic patterns for the hard faults [Hakmi07]. The explanations in the sequel focus on the deterministic part and also apply to an embedded deterministic test [Rajski04]. The starting point for FAST-BIST is a given delay test set for embedded test or mixed-mode BIST. There is no special hardware required, as FAST-BIST works with a standard STUMPS architecture using launch-on-shift (LoS) or launch-on-capture (LoC) for the delay test. As shown in the flow chart of Figure 1, the first step is to identify the hidden delay faults for which FAST is necessary.

Definition 1: Let φ be a gate delay fault of size $\Delta(\varphi)$, let T be a test set, and let f be a frequency. If φ cannot be detected by T at frequency f, then φ is called a *hidden delay fault with respect to T and f*.

Here a gate delay is considered as faulty, if it is an outlier with respect to the delay distribution of the gate. This can be quantified in terms of the standard deviation σ of the mean delay. Standard timing aware fault simulation can be used to determine the set of all hidden delay faults Φ.

The next step is to determine the detection intervals for each hidden delay fault.

Definition 2: Let φ be a gate delay fault of size $\Delta(\varphi)$, let T be a test set, and let t be a point in time. Then t is called a *detecting observation time*, if φ is detected by capturing the test responses for T at time t. The set $I(\varphi)$ of all detecting observation times is an interval or a union of intervals which is referred to as the *detection range of φ with respect to T*.

Based on the detection ranges of faults, the set of hidden delay faults is partitioned into a minimum number of groups $\Phi = \cup_{1 \le i \le m} \Phi_i$, such that all delay faults in a group Φ_i can be detected by applying T (or a subset of T) at the same frequency f_i. The frequencies f_i should be as low as possible. The groups Φ_i are called *FAST groups*.

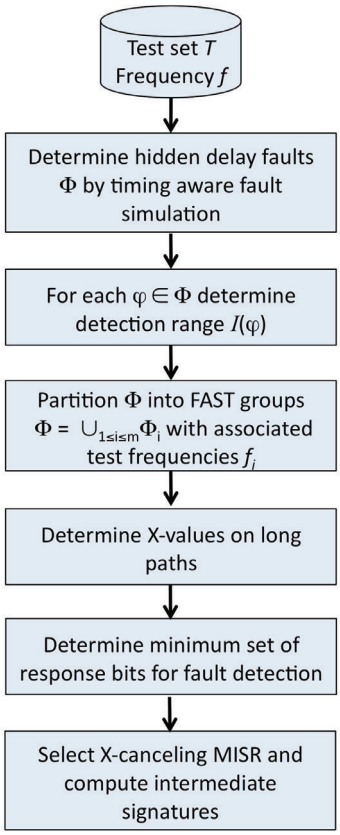

Fig. 1. FAST-BIST flow chart.

Then, the relevant information for test response compaction is collected by simulation. On the one hand, for each FAST group Φ_i and test frequency f_i the X-values on long paths must are determined. To take into account possible glitches, an X-value is assigned pessimistically, if the stable value is reached after the sampling time. On the other hand, the minimum set of response bits required for detection of Φ is derived. Finally, the proper dimensions of the X-canceling MISR are selected, and the necessary intermediate signatures are computed.

4 FAST-BIST – Pre-Processing

This section describes the computation of FAST groups in more detail. As pointed out in Section 3, standard timing aware fault simulation is used to derive the set of hidden delay faults Φ with respect to a given test. The detection range $I(\varphi)$ for a fault $\varphi \in \Phi$ is obtained by analyzing all paths $P(\varphi)$ along which φ is propagated and detected by the initial test set T. This step is feasible, because only the relatively small set of hidden delay faults and only short paths have to be considered. For each path $p \in P(\varphi)$ the detection interval $I_p(\varphi)$ is given by

$$I_p(\varphi) = [length(p), length(p) + \Delta(\varphi)], \quad (1)$$

and the detection range $I(\varphi)$ is obtained as

$$I(\varphi) = \cup_{p \in P(\varphi)} I_p(\varphi). \quad (2)$$

Partitioning the set of hidden delay faults Φ into FAST groups is an optimization problem with two objectives. Firstly, the number of FAST groups should be minimized to avoid expensive switching between test frequencies. Secondly, the frequencies associated with FAST groups should be as low as possible to avoid noise and unnecessary X-values on long paths. If priority is given to the first objective, then the following problem has to be solved.

Problem Minimize FAST-Groups: Given a set Φ of hidden delay faults and their detection ranges $I(\varphi)$ for all $\varphi \in \Phi$. Find a minimum set of observation times $\mathcal{T} = \{t_1, ..., t_n\}$, such that for each $\varphi \in \Phi$ the intersection $I(\varphi) \cap \mathcal{T}$ is not empty. If there are two or more solutions with the same cardinality, select the one with larger observation times.

The observation times t_i define the FAST groups $\Phi_i = \{\varphi \mid t_i \in I(\varphi)\}$ and the associated test frequencies $f_i = 1/t_i$. Finding a minimum set of observation times corresponds to a hitting set problem, which is NP-hard if discrete detection ranges $I(\varphi)$ are considered [Karp72]. The geometric version of the problem with continuous detection ranges is also known to be NP-hard, but here efficient approximations exist [Mustafa10].

In this work the simple greedy heuristic shown in Figure 2 has been implemented as a first prototype solution. In each iteration, the algorithm analyzes the detection ranges of faults not yet assigned to FAST groups. If a detection range $I(\varphi)$ is an interval, then its lower bound $I(\varphi)_{LB}$ is considered as a candidate for t_i. If it is a collection of intervals then $I(\varphi)_{LB}$ denotes the largest lower bound in $I(\varphi)$. Overall the maximum value of all candidate values $I(\varphi)_{LB}$ is selected as new observation time t_i and all faults detected at t_i are deleted from the fault set Φ.

```
// Partition PHI into FAST groups
Φ = {all hidden delay faults}
𝒯 = ∅, i = 0
while (Φ ≠ ∅) {
   i = i+1
   t_i = max_{φ∈Φ} I(φ)_LB
   𝒯 = 𝒯 ∪ {t_i}
   Φ = Φ \ {φ | t_i ∈ I(φ)}
}
```

Fig. 2. Pseudo-code for FAST grouping.

Figure 3 illustrates the grouping procedure for a small example with seven hidden delay faults $\Phi = \{\varphi_1, ..., \varphi_7\}$. The detection ranges are shown as shaded rectangles in the timing diagram. The algorithm slides a line over the timing diagram and selects $I(\varphi_3)_{LB}$, $I(\varphi_5)_{LB}$, and $I(\varphi_2)_{LB}$ as observation times. The resulting FAST groups are $\Phi_1 = \{\varphi_3, \varphi_4\}$, $\Phi_2 = \{\varphi_5, \varphi_7\}$, and $\Phi_3 = \{\varphi_1, \varphi_2, \varphi_6\}$.

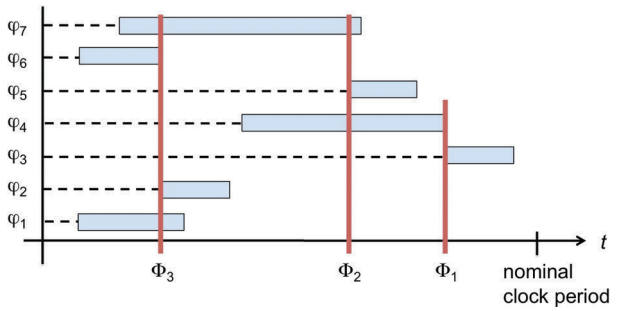

Fig. 3. Example for FAST grouping.

5 FAST-BIST – Compaction

The distribution of X-values is different for each FAST group, since the paths with negative slack depend on the clock rate and the test patterns for each group. For higher clock rates, the number of X-values increases. Test response compaction for FAST-BIST must therefore be extremely flexible and adaptable to changing X-distributions. For this reason a programmable solution with off-line post-processing has been chosen in this work. Among the known X-handling compaction schemes, the X-canceling MISR with deterministic observation has been identified as the most suitable base scheme [Touba07, Garg08]. The main idea of the X-canceling MISR is to analyze the MISR states by symbolic simulation and derive X-free information by linear combinations of MISR bits.

For a better understanding, this is briefly summarized for the small example of Figure 4. The X-bits represent unknown values, and the D-bits are deterministic bits necessary for detecting specific faults. The other bits in the test response are assumed to be don't care for fault detection. After shifting the first scan slice into the MISR, the MISR bits are $m_0 = X_0$, $m_1 = 0$, and $m_2 = 0$.

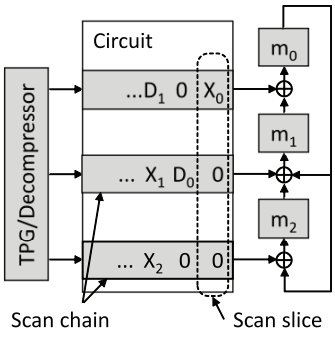

Fig. 4. Example for X-canceling MISR

With the next scan slice entering the MISR, the following equations are obtained for the state bits:

$$m_0 = 0,$$
$$m_1 = X_0 \oplus D_0,$$
$$m_2 = X_0.$$

Appropriate EXOR combinations of the MISR state bits provide:

$$m_0 = 0,$$
$$m_1 \oplus m_2 = X_0 \oplus D_0 \oplus X_0 = D_0,$$
$$m_2 = X_0.$$

This allows observing two X-free combinations of MISR bits, and in particular, the deterministic response bit D_0 can be observed as required. However, with three scan slices compacted in the MISR, the equations

$$m_0 = X_0 \oplus D_0 \oplus D_1,$$
$$m_1 = X_0 \oplus X_1,$$
$$m_2 = X_2$$

cannot be converted into a representation with X-free combinations of MISR bits. Thus, the intermediate signature obtained after the second scan slice must be analyzed, and the MISR must be reset. For the general case, this analysis can be efficiently implemented with the help of matrix representations and Gauss-Jordan elimination [Touba07, Garg08].

To obtain the required flexibility for FAST-BIST, a small memory for storing intermediate signatures is added to the MISR as illustrated in Figure 5. An additional counter controls when intermediate signatures are written to the memory and when the MISR is reset. During test, the intermediate signatures are stored in the memory. In embedded test, these signatures can be shifted out while the subsequent patterns are shifted in. The signatures are evaluated with an offline post-processing routine after test. The size of the memory depends on the number of intermediate signature to be stored, and thus on the number and distribution of X-values, as well as on the number and distribution of D-bits. While the X-values can be limited to a certain extent by a proper selection of FAST groups, the D-bits are selected by the set covering procedure explained in the sequel.

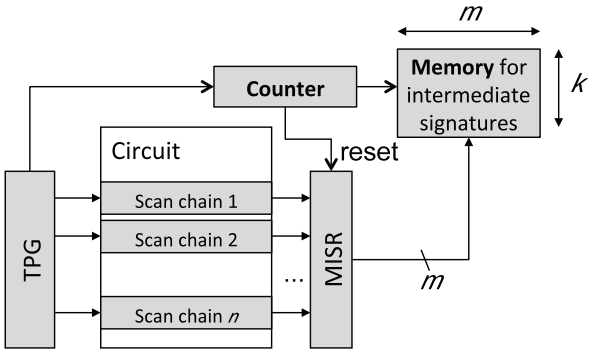

Fig. 5. Architecture for FAST-BIST.

For a given test frequency each deterministic response bit d detects a subset of hidden delay faults $\Phi(d)$. Then minimizing the number of intermediate signatures corresponds to the following covering problem.

Problem D-Bit Selection: Given a set of faults Φ and a collection of subsets $\Phi(d)$ associated with the deterministic response bits of a test T. Find a subset $\mathcal{D}$ of response bits, such that

$$\Phi = \cup_{d \in \mathcal{D}} \Phi(d),$$

and the number of intermediate signatures is minimal.

However, evaluating the candidate solutions for this problem would require a full symbolic analysis of the MISR state sequence. This is highlighted by the small example of Figure 6.

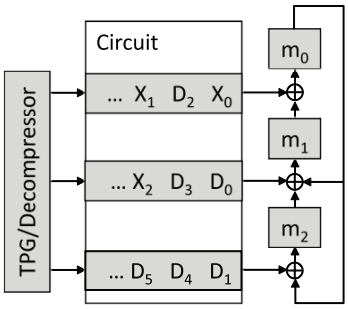

Fig. 6. Example for D-bit selection.

Before D-bit selection all deterministic bits are considered. To demonstrate D-bit selection the following detection profile is assumed: $\Phi = \{\varphi_1, \varphi_2, \varphi_3, \varphi_4, \varphi_5, \varphi_6\}$, $\Phi(D_0) = \{\varphi_1, \varphi_2, \varphi_5\}$, $\Phi(D_1) = \{\varphi_1, \varphi_4, \varphi_5, \varphi_6\}$, $\Phi(D_2) = \{\varphi_3, \varphi_4, \varphi_6\}$, $\Phi(D_3) = \{\varphi_2, \varphi_3\}$, $\Phi(D_4) = \{\varphi_3, \varphi_4\}$, $\Phi(D_5) = \{\varphi_6\}$. Combining the first scan slice with the MISR state provides $m_0 = X_0$, $m_1 = D_0$, and $m_2 = D_1$. With the next scan slice entering the MISR the state bits are:

$$m_0 = D_0 \oplus D_2,$$
$$m_1 = X_0 \oplus D_1 \oplus D_3,$$
$$m_2 = X_0 \oplus D_4.$$

After X-canceling row operations this results in

$$m_0 = D_0 \oplus D_2,$$
$$m_1 \oplus m_2 = D_1 \oplus D_3 \oplus D_4,$$
$$m_2 = X_0 \oplus D_4.$$

At this stage both $\mathcal{D} = \{D_1, D_3\}$ and $\mathcal{D}^* = \{D_0, D_2\}$ would ensure the observation of all faults in Φ with a minimum number of D-bits. Yet, executing one more compaction step yields

$$m_0 = X_0 \oplus X_1 \oplus D_1 \oplus D_3,$$
$$m_1 = X_0 \oplus X_2 \oplus D_0 \oplus D_2 \oplus D_4,$$
$$m_2 = D_0 \oplus D_2 \oplus D_5.$$

Now only $\mathcal{D}^* = \{D_0, D_2\}$ still ensures fault detection. Selecting $\mathcal{D} = \{D_1, D_3\}$ would require to store the second MISR-state as an intermediate signature. Thus $\mathcal{D}^* = \{D_0, D_2\}$ is the optimal solution of the D-selection problem.

To avoid complex symbolic simulation for all considered candidate solutions, the classical set covering problem minimizing the number of D-bits is solved as a first approximation. Instead of minimizing the number of intermediate signatures, the number of deterministic bits entering the MISR is minimized. In the small example for Figure 6, both $\mathcal{D} = \{D_1, D_3\}$ and $\mathcal{D}^* = \{D_0, D_2\}$ are valid solutions for the set covering problem. As this problem is known to be NP-hard as well [Karp72], a greedy heuristic has been implemented, which is interleaved with fault simulation and avoids building large covering tables. Whenever a new fault $\varphi \in \Phi$ is detected at an additional output bit d, the bit is added to $\mathcal{D}$.

It should be noted that aliasing is possible, if the fault sets $\Phi(d)$ for the deterministic bits are not disjoint. Then a fault effect visible at an even number of deterministic bits could be canceled out, if all bits appear in the observed combination of MISR bits.

6 Experimental Results

In order to validate the developed FAST-BIST scheme, several experiments for ITC'99 and NXP benchmark circuits have been performed [ITC99]. The relevant circuits characteristics are shown in Table 1. The first column shows the circuit name, and columns two to four indicate the number of gates, the number of (pseudo)-primary inputs, as wells as the number of (pseudo)-primary outputs. For each circuit 64 scan chains are used. The lengths of the longest scan chains are shown in column 5. Column 6 lists the propagation times for the longest sensitizable paths by the initial test set, which correspond to the nominal clock periods. As both, required slack and the outlier definition of a delay fault are technology dependent, additional margins are not considered. The minimum faults size has been set to 6σ. Increasing the fault size and introducing a larger slack would work in favor of the presented method. In column 7, the minimum clock period for FAST is given, which is 30 % of the nominal clock period. This value is in line with [Amodeo05, Ahmed06, Lee08]. Column 8 presents the number of faults for each circuit. Some of these faults are not detectable at all by FAST, as they are located on short paths with a combined propagation time less than 30 % of the clock period. Finally, the numbers of the remaining, relevant faults are given in column 9.

For each circuit, 1000 deterministic patterns for transition faults have been generated by a commercial ATPG tool. The coverage of all relevant faults by this test set is listed in column 2 of Table 2. The coverage is in the range of very few percent. As mentioned in the introduction, the coverage can be increased with a high quality timing aware test set, but the focus of this work is not on ATPG but on the BIST implementation starting from a given test set. For all the circuits a fault of size 6σ can introduce a malfunction for some patterns. Column 3 of Table 2 gives the number of hidden relevant faults, which are the target of this paper. The fourth column shows the number of FAST groups, which corresponds to the number of different frequencies to be applied from nominal frequency up 10/3 of the nominal frequency (corresponding to 30 % of the nominal clock period). In column 5, the coverage of the hidden relevant faults is presented. As already mentioned, these numbers are test set dependent and not objective of this paper.

Table 1. Circuit Characteristics

Circuit	# Gates	# PPI	# PPO	Max. Scan Chain Length	Nominal Clock Period [ps]	Minimum Clock Period [ps]	# Faults	# Relevant Faults
b14_1	12,438	260	214	4	4,124	1237	23,686	22,150
b17_1	21,858	1,827	1348	22	3,576	1073	95,404	72,094
b18_1	75,618	4,116	3,085	49	4,484	1345	255,960	180,952
b20_1	25,547	533	450	8	4,269	1281	56,948	53,992
b21_1	25,561	534	450	8	4,276	1283	57,432	54,426
b22_1	38,568	786	664	11	4,445	1334	90,231	85,609
p35k	22,803	2,861	2,229	35	3,159	948	95,942	55,310
p45k	22,414	3,739	2,550	40	3,499	1050	127,295	79,083
p78k	46,504	3,148	3,484	58	1,475	443	268,989	256,099
p81k	78,665	4,029	3,952	62	1,586	476	434,514	414,005

Table 2. Hidden Delay Faults and FAST Groups

Circuit	Coverage of Relevant Faults	# Hidden Relevant Faults	# Fast Groups	Detected Hidden Relevant Faults (%)
b14_1	0.02%	22,146	3	24.11%
b17_1	2.02%	70,637	4	22.20%
b18_1	0.04%	180,885	5	41.51%
b20_1	0.09%	53,939	4	33.34%
b21_1	0.22%	54,304	4	33.69%
b22_1	0.02%	85,590	7	34.69%
p35k	0.09%	55,259	10	3.44%
p45k	0.02%	79,079	2	11.67%
p78k	1.85%	251,367	8	82.08%
p81k	1.29%	408,676	5	76.88%

None of the circuits needs more than 10 different frequencies to detect all relevant faults testable by the given test set.

Finally, the number of intermediate signatures to be stored on chip has been determined. Here, a basic solution without considering fault detection information has been compared to the results of D-minimization. In the basic solution the intermediate signatures have been distributed in the state sequence of the MISR, such that at least seven X-free combinations of MISR bits could be observed as recommended in [Touba07]. For each case, the MISR sizes m have been varied using $m = 64$, $m = 128$, $m = 256$, *and* $m = 512$.

Table 3 shows the number of X-values to be cancelled and the required number of intermediate signatures to be stored. As expected the number of signatures decreases reciprocally with the MISR length. While shorter MISRs have a slight advantage in terms of memory size, they may lead to higher aliasing. In case of a MISR width of 64 bits, it is not guaranteed to find a solution with seven X-independent combinations. These situations are marked with n.a. in the table. Table 4 applies the D-minimization, and here the second column denotes the number of bits to be evaluated. Especially for the larger circuits and smaller MISRs, up to 30 % storage can be saved with respect to the basic solution, but the remarks about aliasing still apply.

Table 3. Intermediate Signatures for 7 X-Free Combinations

Circuit	# X	# Intermediate Signatures for m =			
		64	128	256	512
b14_1	5,003	n.a.	37	19	9
b17_1	12,174	180	93	47	23
b18_1	130,015	n.a.	2,272	1,150	579
b20_1	25,725	374	192	99	50
b21_1	34,240	506	259	131	66
b22_1	65,116	943	487	249	125
p35k	2,008	31	15	10	3
p45k	278,211	4,399	2,185	1,090	544
p78k	185,044	n.a.	1,514	739	366
p81k	408,855	n.a.	3,143	1,584	795

Table 4. Intermediate Signatures with D-Minimization

Circuit	# D	# Intermediate Signatures for m =			
		64	128	256	512
b14_1	1,673	48	29	17	9
b17_1	1,637	118	71	41	22
b18_1	36,297	2,874	1,748	1,009	543
b20_1	6,041	247	151	89	47
b21_1	5,761	317	195	115	62
b22_1	9,816	598	375	215	118
p35k	315	27	14	7	3
p45k	9,652	2,358	1,505	891	491
p78k	45,769	1,650	1,045	601	333
p81k	48,252	3,446	2,223	1,310	723

Even for the largest circuit, just 220 K of signature memory are required. In embedded testing, signatures can still be read out at nominal frequency even during FAST, and a smaller buffer is sufficient.

7 Conclusions

Faster than at speed test (FAST) makes small delay faults observable even before they alter the system functionality. This work shows how embedded deterministic test and mixed-mode BIST schemes, which support LoC, LoS or enhanced scan, can be modified with little hardware overhead to support FAST as well.

For a given test pattern set, all delay faults are determined which exceed a given size and require a fast test up to a given frequency. This fault set is partitioned into a minimum number of groups such that all the faults of a group can be tested by the same frequency. With increasing frequency, the number of undefined values, X-values, captured in the MISR grows as well. The method presented here provides a minimum set of intermediate signatures to reconstruct all the necessary error indicating bits by X-cancelling. Just a few hundred intermediate signatures have to be stored in an on-chip memory to cover all the detectable hidden delay faults.

8 Acknowledgements

Parts of this work have been supported by the German Research Foundation (DFG) under grant WU 245/11-1, OASIS.

9 References

[Ahmed06] N. Ahmed, M. Tehranipoor, and V. Jayaram, "A Novel Framework for Faster-than-at-Speed Delay Test Considering IR-drop Effects," Proc. IEEE/ACM Int. Conf. on Computer Aided Design (ICCAD'06), San Jose, CA, USA, Nov. 2006, pp. 198–203.

[Amodeo05] M. Amodeo and B. Cory, "Defining faster-than-at-speed delay tests," Cadence Nanometer Test Quarterly eNewsletter, Vol. 2, No. 2, May 2005.

[Bardell82] P. H. Bardell and W. H. McAnney, "Self-Testing of Multichip Logic Modules," Proc. IEEE Int. Test Conf. (ITC'82), Philadelphia, PA, USA, Nov. 1982, pp. 200-204.

[Cory03] B. D. Cory, R. Kapur, and B. Underwood, "Speed Binning with Path Delay Test in 150nm Technology," IEEE Design & Test of Computers, Vol. 20, No. 5, 2003, pp. 41-45.

[Czysz10] D. Czysz, et al., "On compaction utilizing inter and intra-correlation of unknown states," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 29, No. 1, Jan. 2010, pp. 117-126.

[Eggersgluess11] S. Eggersglüss, R. Drechsler, "As-Robust-As-Possible Test Generation in the Presence of Small Delay Defects using Pseudo-Boolean Optimization," Proc. Design, Automation and Test in Europe (DATE'11), Grenoble, France, April 2011, pp. 1291-1296.

[Fu12] X. Fu, H. Li, and X. Li, "Testable path selection and grouping for faster than at-speed testing," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol. 20, No. 2, 2012, pp. 236-247.

[Garg08] R. Garg, R. Putman, and N. A. Touba, "Increasing Output Compaction in Presence of Unknowns using an X-Canceling MISR with Deterministic Observation," Proc. IEEE VLSI Test Symposium (VTS'08), San Diego, CA, USA, April – May 2008, pp. 35-42.

[Hakmi07] A.-W. Hakmi et al., "Programmable Deterministic Built-In Self-Test," Proc. IEEE Int. Test Conf. (ITC'07), Santa Clara, CA, USA, October 2007, pp. 1-9.

[ITC99] Benchmark information and circuits available at http://www.cerc.utexas.edu/itc99-benchmarks/bench.html

[Karp72] R. M. Karp, "Reducibility Among Combinatorial Problems," in R. E. Miller and J. W. Thatcher (editors), "Complexity of Computer Computations," New York: Plenum, 1972, pp. 85-103.

[Kim10] Y. M. Kim, et al., "Low-cost gate-oxide early-life failure detection in robust systems," Proc. IEEE Symposium on VLSI Circuits (VLSIC'10), Honolulu, HI, USA, June 2010, pp.125-126.

[Krusemann04] B. Kruseman, et al., "On hazard-free patterns for fine-delay fault testing," Proc. IEEE Int. Test Conf. (ITC'04), Charlotte, NC, USA, Oct. 2004, pp. 213–222.

[Lee08] J. Lee and E. J. McCluskey, "Failing Frequency Signature Analysis," Proc. IEEE Int. Test Conf. (ITC'08), Santa Clara, CA, USA, Oct. 2008, pp. 1–8.

[Lin06] X. Lin, et al., "Timing-Aware ATPG for High Quality At-speed Testing of Small Delay Defects," Proc. 15th Asian Test Symposium (ATS'06), Fukuoka, Japan, Nov. 2006, pp.139-146.

[Malandruccolo11] V. Malandruccolo, et al., "Design and Experimental Characterization of a New Built-In Defect-Based Testing Technique to Achieve Zero Defects in the Automotive Environment", IEEE Trans. on Device and Materials Reliability, Vol. 11, No. 2, June 2011, pp. 349-357.

[McLaurin00] T. L. McLaurin and F. Frederick, "The Testability Features of the MCF5407 Containing the 4th Generation ColdFire® Microprocessor Core," Proc. IEEE Int. Test Conf. (ITC'00), Atlantic City, NJ, USA, Oct. 2000, pp. 151-159.

[Mitra02] S. Mitra and K. S. Kim, "X-Compact: An efficient response compaction technique for test cost reduction," Proc. IEEE Int. Test Conf. (ITC'02), Baltimore, MD, USA, Oct. 2002, pp. 311-320.

[Mitra04] S. Mitra, S. S. Lumetta, M. Mitzenmacher, "X-tolerant signature analysis," Proc. IEEE Int. Test Conf. (ITC'04), Charlotte, NC, USA, Oct. 2004, pp. 432-441.

[Mustafa10] N. H. Mustafa, S. Ray, "Improved Results on Geometric Hitting Set Problems," Discrete & Computational Geometry, Vol. 44, No. 4, December 2010, pp 883-895.

[Naruse03] M. Naruse, et al., "On-chip compression of output responses with unknown values using LFSR reseeding," Proc. IEEE Int. Test Conf. (ITC'03), Charlotte, NC, USA, Sep. – Oct. 2003, pp. 1060-1068.

[Pei10] S. Pei, H. Li, and X. Li, "An On-Chip Clock Generation Scheme for Faster-than-at-Speed Delay Testing," Proc. Design and Test in Europe (DATE'10), Dresden, Germany, March 2010, pp. 1353-1356.

[Press06] R. Press and J. Boyer, "Easily Implement PLL Clock Switching for At-Speed Test," Chip Design Magazine, Feb./March 2006.

[Rajski03] J. Rajski, et al., "Convolutional compaction of test responses," Proc. IEEE Int. Test Conf. (ITC'03), Charlotte, NC, USA, Sep. – Oct. 2003, pp. 745-754.

[Rajski04] J. Rajski, et al., "Embedded Deterministic Test," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 23, No. 5, 2004, pp. 776-792.

[Sauer13] M. Sauer, et al., "Early-life-failure detection using SAT-based ATPG," Proc. IEEE Int. Test Conf. (ITC'13), Anaheim, CA, USA, Sep. 2013, pp. 1-10.

[Sharma05] M. Sharma and W.-T. Cheng, "X-Filter: Filtering Unknowns from Compacted Test Responses," Proc. IEEE Int. Test Conf. (ITC'05), Austin, TX, USA, Nov. 2005, pp. 1-9.

[Singh10] A. Singh, C. Han, X. Qian, "An Output Compression Scheme for Handling X-States from Over-Clocked Delay

Test," Proc. 28th IEEE VLSI Test Symposium, Santa Cruz, CA, USA, April 2010, pp. 57-62.

[Tang06] Y. Tang et al., "X-Masking During Logic BIST and its Impact on Defect Coverage," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol. 14, No. 2, Feb. 2006, pp. 193-202

[Tayade08] R. Tayade and J. A. Abraham, "On-chip Programmable Capture for Accurate Path Delay Test and Characterization," Proc. IEEE Int. Test Conf., Santa Clara, CA, USA, Oct. 2008, pp. 1-10.

[Tehranipoor12] M. Tehranipoor, K. Peng, K. Chakrabarty, "Test and Diagnosis of Small-Delay Defects," Springer, 2012, ISBN 978-1-4419-8296-4

[Touba07] N. A. Touba, "X-canceling MISR - An X-tolerant Methodology for Compacting Output Responses with Unknowns Using a MISR," Proc. IEEE Int. Test Conf. (ITC'07), Santa Clara, CA, USA, Oct. 2007, pp. 1-10.

[Yan03] H. Yan, A. D. Singh, "Experiments at Detecting Delay Faults using Multiple Higher Frequency Clocks and Results from Neighboring Die," Proc. IEEE Int. Test Conf. (ITC'03), Charlotte, NC, USA, Sep. - Oct. 2003, pp. 105-111.

[Yilmaz10] M. Yilmaz, K. Chakrabarty, and M. Tehranipoor, "Test-pattern selection for screening small-delay defects in very-deep submicron integrated circuits," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 29, No. 5, May 2010, pp. 760-773.

[Yoneda11] T. Yoneda, et al., "Faster- Than-At-Speed Test for Increased Test Quality and In-Field Reliability," Proc. IEEE Int. Test Conf. (ITC'11), Anaheim, CA, USA, Sep. 2011, pp. 1-9.

[Wohl07] P. Wohl, J. A. Waicukauski, and S. Ramnath, "Fully X-Tolerant Combinational Scan Compression," Proc. IEEE Int. Test Conf. (ITC'07), Santa Clara, CA, USA, Oct. 2007, pp. 1-10.

[Zeng04] J. Zeng, et al., "On correlating structural tests with functional tests for speed binning of high performance design," Proc. IEEE Int. Test Conf. (ITC'04), Charlotte, NC, USA, October 2004, pp. 31-37.

[Zolotov10] V. Zolotov, et al., "Statistical Path Selection for At-Speed Test," IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems, Vol. 29, No. 5, May 2010, pp. 749-759.