Fault Detection and Diagnosis with Parity Trees for Space Compaction of Test Responses

Harald Vranken[1], Sandeep Kumar Goel[1], Andreas Glowatz[2], Juergen Schloeffel[2], Friedrich Hapke[2]

[1]Philips Research
5656 AE, Eindhoven, The Netherlands

[2]Philips Semiconductors
D-21147 Hamburg, Germany

Email: {sandeepkumar.goel, andreas.glowatz, juergen.schloeffel, friedrich.hapke}@philips.com

ABSTRACT

This paper shows that accurate fault detection and diagnosis are possible when applying a parity tree for space compaction of test responses and continuously observing the parity-tree output. We exploit that each set of faults results in a unique parity-tree output sequence that can be used as a unique fault signature for truly accurate fault detection and diagnosis. Our theoretical analysis shows that probabilities for fault cancellation and fault aliasing are extremely low and decrease with circuit size. Experimental results show that for a typical scan chain architecture in large industrial circuits, fault coverage is not affected by parity-tree space compaction for up to 256 scan chains, while diagnostic resolution is reduced by only 1% to 2%.

Categories and Subject Descriptors

B.8.1 [**Performance and Reliability**]: Reliability, testing and fault-tolerance.

General Terms: Performance.

Keywords: Test data compression, fault detection, fault diagnosis.

1. INTRODUCTION

The test data volume for IC manufacturing test has risen dramatically, due to a combination of growth in transistor count and new advanced test methods (such as delay-fault testing), which add significantly to the test set size. The increase in test data volumes causes critical problems related to ATE usage, since test data volumes may exceed the capacity of ATE memories and increasing test times imply increasing test costs.

Solutions are provided by test data compression and built-in self-test (BIST) techniques. In test data compression, the ATE stores the test stimuli and test responses in a compressed format. The decompression of the test stimuli is performed by on-chip decompression circuitry, and the test responses are compacted on-chip in space and/or time by compaction circuitry. The focus of this paper is on space compaction of test responses, where the space compactor consists of a basic XOR-tree implementing a parity function as shown in Figure 1. The n=4 scan chain outputs of the circuit-under-test (CUT) are compacted in each cycle into a single parity bit by the space compactor (SC). The parity bit has value 1 in case of an odd number of scan chains that output logic 1, while the parity bit has value 0 in case of an even number of scan chains that output logic 1.

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

In this paper, we propose an original approach using parity trees for accurate fault detection and fault diagnosis. The proposed approach allows detailed as well as statistical diagnosis to take place during the production test. This is of great value for fast yield ramp-up and fast diagnosis of systematic defects. Key idea is to observe the parity-tree output continuously during the entire test, and to consider that each fault causes multiple fault effects that usually can be observed in multiple cycles at the scan chain outputs in one or multiple patterns. Each fault therefore has a rather unique parity-tree output sequence, which we exploit for truly accurate fault detection and diagnosis.

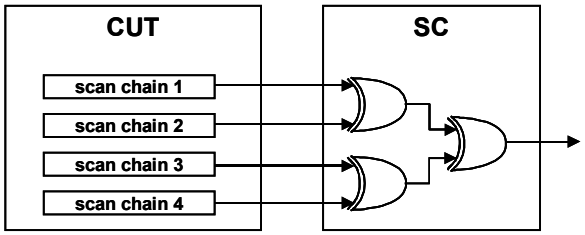

Figure 1: CUT with parity-tree space compactor

The remainder of this paper is organized as follows. Section 2 describes prior work on space compaction of test responses. Section 3 introduces basic principles of fault detection and fault diagnosis. Section 4 contains the theoretical analysis of fault cancellation and fault aliasing. Section 5 discusses bit masking to further improve fault detection and diagnosis. Section 6 presents experimental results, and Section 7 concludes the paper.

2. PRIOR ART

Parity trees have various advantages; they are easy to generate and to implement, and are independent of the CUT and the test pattern set. It has been shown that fault cancellation and fault aliasing can be reduced or eliminated completely by dedicated space compactors [1][3][6]. However, constructing such space compactors always requires knowledge of the test patterns and the CUT, and hence these compactors are not design independent. Parity trees are relatively insensitive to X's (i.e., test responses with unknown values). The output of the parity tree should be discarded in those cycles where X's show up at the inputs of the parity tree, while the output is valid in all other cycles. The use of X-masking logic is an increasingly popular approach to deal with compaction of responses that contain unknowns. Examples of this approach are channel masking [2], mask data encoding [8], or dedicated X-masking logic for BIST [4][7].

The vast majority of papers on parity trees consider probabilities for fault cancellation, fault aliasing, and fault diagnosis, solely on a cycle-by-cycle basis. In this paper, we take a different approach, in the sense that we consider the complete output sequence of the parity tree during the entire test. We also consider that all unknowns X's are masked before compaction.

3. FAULT DETECTION AND DIAGNOSIS

3.1 Definitions

A *response bit* contains (part of) the test response to a test stimulus. We define set R that contains all response bits (of all test patterns) for a given CUT. An *observation bit* is a response bit in which a fault effect is observed. Set $O \subseteq R$ contains all observations bits (of all test patterns) for a CUT.

A *fault signature* is a set of all observation bits $O_f \subseteq O$ in which the fault effects of a fault f are observed.

A *failure set* is a set of faults that all have the same fault signature. Failure set $FS_i = \{F_i, O_i\}$ consists of a set of faults F_i and fault signature O_i such that $\forall f \in F_i : O_f = O_i$. For a CUT with n failure sets and set F of all faults, holds that $\cup_{1 \le i \le n} F_i = F$, and $\cup_{1 \le i \le n} O_i = O$. The sets of faults in the failure sets are unique and disjoint, however the fault signatures in the failure sets are unique but usually not disjoint. FS is the set of all p failure sets of a CUT when observing fault effects at the n scan chain outputs, and FS^{SC} is the set of all q failure sets of the CUT+SC when observing fault effects at the parity-tree output only. It holds that $p \ge q$.

The *fault coverage FC* is the ratio of all faults that are detected for a given test pattern set when observing the CUT outputs. The *fault coverage* FC^{SC} is the ratio of all faults that are detected for a given test pattern set when observing the parity-tree output only. We define the *relative fault coverage* FC^{SCrel} as $100\% \cdot FC^{SC}/FC$.

We define the *diagnostic resolution DR* as the ratio of the number of failure sets observed at the parity-tree output (q) and to the number of failure sets observed at the CUT outputs (p). Hence, $DR = 100\% \cdot q/p$.

Fault cancellation of failure set $FS_i = \{F_i, O_i\}$ occurs when the number of observation bits in O_i at the parity-tree inputs is zero or even in all cycles. Fault cancellation implies that the fault effects of all faults $f \in F_i$ cannot be observed at the parity-tree output. Hence, fault cancellation causes loss of both fault coverage and diagnostic resolution.

Fault aliasing occurs when multiple failure sets FS_i at the CUT outputs result in the same failure set FS_j^{SC} at the parity-tree output. Fault aliasing does not cause fault coverage loss, but causes loss of diagnostic resolution.

3.2 Generating Failure Sets

The ATPG tool creates a fault dictionary of all target faults and generates test patterns to detect these faults. The failure sets can be derived by fault simulation of all test patterns without fault dropping. In this way, all observation bits in all test patterns are identified for each fault. The random-testable faults are usually detected by many test patterns, and hence the number of observation bits for such faults can grow very large. Therefore, the storage requirements for the failure sets for large circuits may easily take several Gigabytes.

A practical way for reducing this data volume is to introduce a *fault isolation limit*, such that the maximum number of observation bits in each failure set does not exceed this limit. During fault simulation, a fault is now dropped after it has been detected by the number of observation bits as specified by the fault isolation limit. The fault isolation limit should not be chosen too small, since this would reduce the number of failure sets and hence reduce the diagnostic resolution. Practical values for the fault isolation limit are in the range of a few hundreds. We generate the failure sets for stuck-at faults.

4. THEORETICAL ANALYSIS

4.1 Fault Cancellation

When continuously observing the output of the parity tree, fault cancellation occurs only for those failure sets that have zero or an even number of observation bits in each cycle during the entire test. If we assume that a failure set has c observation bits, then there are $\binom{r}{c}$ possible ways in which these c observation bits can be distributed over r response bits, and the fraction out of these $\binom{r}{c}$ cases in which fault cancellation occurs, can be computed by means of counting. The test length consists of k cycles, with $k = r/n$ for a CUT with n scan chains. We can partition c into a sequence of d positive integers c_i such that $c = \sum_{i=1}^{d} c_i$. This corresponds to the situation in which c_i ($1 \le i \le d$) observation bits are compacted by the parity tree during d cycles.

We generate set P of all possible partitions of c, such that for each partition $p \in P$ holds that $d(p) \le k$, $c_i(p) \le n$, and $c_i(p)$ is even for $1 \le i \le d(p)$. For each partition $p \in P$, we compute $D(d(p))$, which is the number of ways that $d(p)$ cycles can be distributed over k cycles, and we compute $D(p)$, which is the number of ways that $d(p)$ groups with $c_i(p)$ observation bits can be distributed over $d(p)$ groups of n bits.

The total number of ways in which partition p can be distributed over the r response bits follows from $D(d(p)) \cdot D(p)$. This is repeated for all partitions in P, and the sum finally corresponds to the total number of cases in which the c observation bits are distributed over r response bits such that fault cancellation occurs.

Figure 2 shows the fault cancellation probability for even numbers of observation bits. For large industrial circuits, the number of observation bits per failure sets is around 100 on average, while the test length is in the order of 100k to 100M cycles. For such large circuits, fault cancellation probability is extremely low.

4.2 Fault Aliasing

We compute the fault aliasing probability in a similar way as in [5] assuming that all fault signatures are equally likely to occur and independent of each other. For a test of k cycles, the number of possible fault signatures that can be observed at the parity-tree output is 2^k-1, excluding the fault-free case. A necessary condition for minimizing fault aliasing is $m < 2^k$-1, where m is the number of failure sets.

We introduce a graph with m levels, where the nodes at level i represent states for i failure sets. Each node is labeled by (i,a,b), where i is the level, a is the number of fault signatures generated by at least two failure sets, and b is the number of unique fault signatures generated by exactly one failure set.

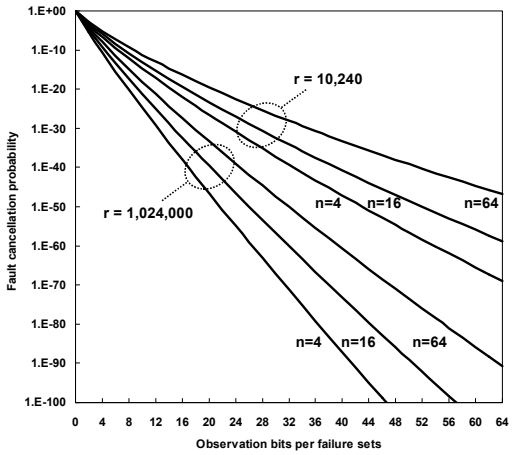

Figure 2: Fault cancellation probability

It holds that $a+b \leq i$ for each node (i,a,b). Each node has at most three outgoing edges leading to three different nodes in the next level. The possible successors of node (i,a,b) are the following:

1. Node $(i+1,a,b)$. In this case, the fault signature of failure set $i+1$ is similar to the fault signature of at least two other failure sets. Hence, fault aliasing occurs for failure set $i+1$, and the probability for this is $a/(2^k-1)$.
2. Node $(i+1,a+1,b-1)$. In this case, the fault signature of failure set $i+1$ is similar to the fault signature of exactly one other failure sets. Hence, fault aliasing occurs for failure set $i+1$, and the probability for this is $b/(2^k-1)$.
3. Node $(i+1,a,b+1)$. In this case, failure set $i+1$ has a unique fault signature, and hence no aliasing occurs. The probability that this occurs is $1-(a+b)/(2^k-1)$.

The probability for node $(i+1,a,b)$ is: $P(i+1,a,b) =$

$$P(i,a,b) \cdot a/(2^k-1) + P(i,a-1,b+1) \cdot b/(2^k-1) + P(i,a,b-1) \cdot (1-(a+-b)/(2^k-1)),$$

with initial probability $P(1,0,1) = 1$. The fault aliasing probability for m failure sets is obtained from the graph with m levels. No fault aliasing occurs for node $(m,0,m)$, and hence the fault aliasing probability is:

$$1 - P(m,0,m) = 1 - \prod_{i=2}^{m}\left(1-\frac{i}{2^k-1}\right) \quad (1)$$

The average number of unique fault signatures is obtained by $\bar{b} = \sum_{a,b=1}^{m} b \cdot P(m,a,b)$, and the diagnostic resolution (i.e., the fraction of failure sets with unique fault signatures) is $\bar{b}/m$.

4.3 Discussion

The analysis of fault cancellation and fault aliasing assumed that observation bits are uniformly distributed over the response bits. In practice, fault effects will be observed only in flip-flops that are close to the physical location of the defect. Faults may be detected in multiple patterns, but the fault effects will still be observed in the same flip-flops. Hence, in practice the observation bits for each fault will be clustered instead of being distributed uniformly. The number of observation bits per failure set is also not distributed uniformly in practice. Experimental results show that the number of observation bits per failure set is distributed geometrically. The theoretical analysis nevertheless is very useful since it illustrates the parameters that affect fault cancellation and fault aliasing.

5. BIT MASKING

Parity-tree space compactors are relatively insensitive to X's, although the parity-tree output sequence becomes less useful if X's show up in many cycles. We therefore apply X-masking logic (XML) in front of the space compactor (similar as in [2][4][7][8]) to mask all X's before the responses are compacted, as shown in Figure 3.

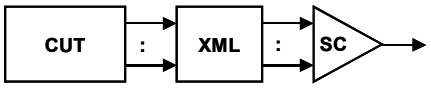

Figure 3: CUT, X-masking logic, and space compactor

The X-masking logic can also be used to reduce fault cancellation. By masking one of the observation bits, the number of observation bits can be transformed from even to odd, and the fault effect can be observed at the parity-tree output. Bit masking should be applied with care, since each observation bit is usually contained in multiple failure sets. Hence, repairing fault cancellation for one failure set may cause fault cancellation for other failure sets, while also fault aliasing may occur. For each failure set suffering from fault cancellation, we therefore mask only one observation bit that is used least often in other failure sets. Such failure set can now be observed in a single cycle at the parity-tree output.

Fault aliasing is more difficult to repair, since fault aliasing can only be analyzed after all patterns have been generated. Nevertheless, bit masking can also be applied to reduce fault aliasing. In case of aliasing of two failure sets, an observation bit can be masked that appears in only one of the failure sets. ATPG can be used to generate additional patterns to resolve fault aliasing, although this increases pattern count.

6. EXPERIMENTAL RESULTS

We performed experiments for several state-of-the-art Philips circuits. The Philips circuits are named *pN*, where *N* indicates the number of signal nets as reported by the ATPG tool. We performed experiments for each circuit with varying number of scan chains. We used the Philips ATPG tool AMSAL to generate the stuck-at test patterns and the failure sets.

We varied the number of scan chains for each circuit by partitioning the response bits for each pattern into n balanced parts. We next compacted the response bits using a parity tree with n inputs, and computed the failure sets as observed at the output of the parity tree. By comparing the failure sets before and after the space compactor, the relative fault coverage FC^{SCrel} and the diagnostic resolution DR is obtained.

We used two different scenarios for partitioning the scan chains, as illustrated in Figure 4. Figure 4a corresponds to a CUT with a single scan chain of 8 flip-flops. (This also corresponds to a CUT with 8 flip-flops in multiple scan chains of arbitrary length that are concatenated.). The scan chain is partitioned into two balanced scan chains; In Figure 4b (*Configuration 1*) the scan chain is simply cut in two, while in Figure 4c (*Configuration 2*) the flip-flops are assigned alternately to the two scan chains. Assume that scan chain in Figure 4a is created according to the actual layout positions of the flip-flops. A defect near flip-flop 2 will then most likely be observed only in flip-flop 2 and its

neighbors, while it is unlikely that the defect will be observed in flip-flops 6 to 8. Fault cancellation is therefore more unlikely to occur in Figure 4b than in Figure 4c. The typical behavior of scan insertion tools is according to Figure 4b, while Figure 4c is less common. We therefore consider Figure 4b as typical, and Figure 4c as worst-case scan chain configurations.

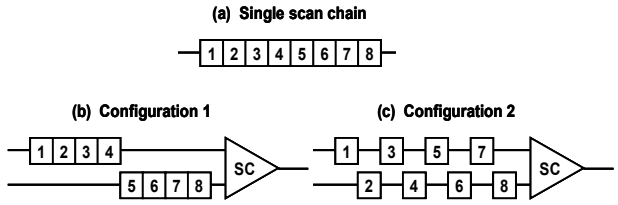

Figure 4: Examples of scan chain partitioning

Table 1 shows the experimental results for the Philips circuits for varying number of scan chains between 2 to 1024. For most circuits the scan chains are optimized considering the layout positions of the flip-flops. It is therefore expected that fault cancellation will cause larger reduction of both diagnostic resolution and fault coverage in configuration 2. This can indeed be seen from the table: for 1024 scan chains, the fault coverage is 99.59% on average for configuration 1 versus 97.24% for configuration 2, while the diagnostic resolution is 96.72% on average for configuration 1 versus 88.38% for configuration 2.

Table 1: Experimental results for the industrial circuits.

Circuit	Diagnostic Resolution (%)					Relative Fault Coverage (%)				
#chains	2	8	32	256	1024	2	8	32	256	1024
Scan Chain Configuration 1 (see Figure 4b)										
p133146	100	99.95	99.82	98.57	95.01	100	100	100	99.91	99.71
p288091	100	99.97	99.86	99.20	97.81	100	100	100	100	99.94
p349343	100	99.95	99.78	98.71	96.82	100	100	100	100	99.98
p423503	99.99	99.97	99.86	99.15	98.41	100	100	100	100	99.99
p482917	99.99	99.94	99.70	98.18	95.23	100	100	100	100	98.76
p532673	100	99.97	99.84	98.81	96.49	100	100	100	99.98	99.54
Average	**100**	**99.96**	**99.81**	**98.76**	**96.72**	**100**	**100**	**100**	**99.99**	**99.59**
Scan Chain Configuration 2 (see Figure 4c)										
p133146	98.26	94.95	91.89	88.02	84.98	99.92	99.75	99.24	98.33	97.35
p288091	98.96	96.92	94.74	91.71	90.00	99.88	99.67	99.31	98.56	97.70
p349343	98.15	95.12	92.60	89.41	87.24	99.76	99.06	98.31	97.21	96.34
p423503	99.67	98.47	96.70	92.66	90.58	99.94	99.69	99.28	98.09	97.49
p482917	97.85	94.89	92.57	89.62	87.37	99.77	99.42	99.16	98.55	97.40
p532673	97.96	95.22	93.02	89.97	88.27	99.76	99.39	99.08	98.23	97.22
Average	**98.44**	**95.96**	**93.72**	**90.43**	**88.38**	**99.82**	**99.46**	**99.05**	**98.16**	**97.24**
Scan Chain Configuration 2 (see Figure 4c) with Bit Masking										
p133146	98.42	95.40	92.92	89.88	87.12	99.99	99.96	99.87	99.69	99.35
p288091	99.12	97.34	95.53	93.24	91.70	99.96	99.89	99.79	99.61	99.29
p349343	98.51	95.98	93.99	91.33	89.52	99.88	99.55	99.29	99.06	98.93
p423503	99.75	98.83	97.40	94.03	92.16	99.97	99.88	99.74	99.34	99.12
p482917	98.44	96.11	94.21	92.03	90.73	99.91	99.74	99.58	99.43	99.36
p532673	98.47	96.30	94.51	92.41	91.23	99.90	99.69	99.50	99.31	99.22
Average	**98.80**	**96.77**	**94.95**	**90.84**	**90.84**	**99.93**	**99.76**	**99.59**	**99.36**	**99.20**

For Configuration 1, the fault coverage is not affected by the space compaction for up to 256 scan chains, while the diagnostic resolution is reduced by only 1% to 2%. The third part of the table (*Configuration 2 with bit-masking*) reports results for configuration 2 with bit masking to reduce fault cancellation. It can be seen that bit masking is very effective, since even for 1024 scan chains the fault coverage is 99.20% on average, compared to 97.24% without bit masking. The diagnostic resolution improves also. Only very few observation bits were masked, typically less than 0.1% of the response bits. The bit-masking allows to limit the loss in fault coverage by 0.5% and the loss in diagnostic resolution by 6% for up to 32 scan chains.

7. CONCLUSION

We showed that accurate fault detection and fault diagnosis is possible when applying a parity tree for space compaction of test responses and continuously observing the parity-tree output. We exploit that each failure set results in a unique parity-tree output sequence (i.e., fault signature), which allows truly accurate fault detection and fault diagnosis. Our theoretical analysis shows that probabilities for fault cancellation and fault aliasing are extremely low. It also reveals that both probabilities decrease drastically with the number of test responses.

Our experimental results show that for a worst-case scan chain architecture, bit masking allows to limit the loss in fault coverage and diagnostic resolution. For a typical scan chain architecture in large industrial circuits, the fault coverage is not affected by parity-tree space compaction for up to 256 scan chains, while the diagnostic resolution is reduced by only 1% to 2%. Even for 1024 scan chains, the loss in fault coverage and diagnostic resolution is only 0.4% and 3.3% on average, while also in this case results can be improved by applying bit masking.

8. ACKNOWLEDGMENTS

This research work was funded partially by the European MEDEA+ Program ('NanoTEST' project) and the German Federal Ministry for Education and Research in the project AZTEKE under contract number 01M3063C.

9. REFERENCES

[1] K. Chakrabarty, *Zero-aliasing space compaction using linear compactors with bounded overhead*, IEEE Trans. CAD of Integrated Circuits and Systems, Vol. 17, No. 5, pp. 452-457, 1998.

[2] V. Chickermane, B. Foutz, B. Keller, *Channel masking synthesis for efficient on-chip test compression*, IEEE Proc. Int. Test Conf., pp. 452-461, 2004.

[3] A. Morosov et al., Design of parameterizable error-propagating space compactors for response observation, IEEE Proc. VLSI Test Symp., pp. 48-53, 2001.

[4] I. Pomeranz, S. Kundu, S.M. Reddy, *On output response compression in the presence of unknown output Values*, ACM/IEEE Proc. DAC, pp. 255-258, 2002.

[5] J. Rajski, J. Tyszer, *On the diagnostic properties of linear feedback shift registers*, IEEE, Trans. CAD of Integrated Circuits and Systems, Vol. 10, No. 10, pp. 1316-1322, 1991.

[6] O. Sinanoglu, A. Orailoglu, *Parity-based output compaction for core-based SOCs*, IEEE Proc. European Test Workshop, pp. 15-20, 2003.

[7] Y. Tang et al., *X-Masking during logic BIST and its impact on defect coverage*, IEEE Proc. Int. Test Conf., pp. 442-451, 2004.

[8] M. Narsue et al., *On-chip compression of output responses with unknown values using LFSR reseeding,* IEEE Proc. Int. Test Conf. pp. 1060-1068, 2003.

The average numbers in Table 1 are weighted averages, where the weights are the number of signal nets in the circuits.