Process Variation-Aware Test for Resistive Bridges

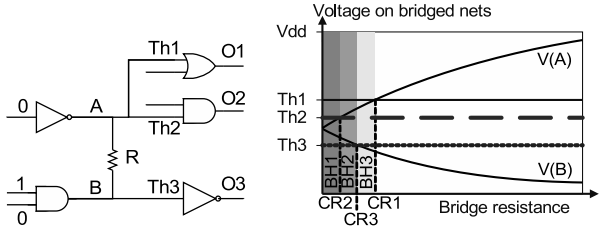
Urban Ingelsson, *Student Member, IEEE* Bashir M. Al-Hashimi, *Fellow, IEEE* Saqib Khursheed, *Student Member, IEEE* Sudhakar M. Reddy, *Life Fellow, IEEE* and Peter Harrod, *Senior Member, IEEE*

***Abstract*—This paper analyses the behaviour of resistive bridging faults under process variation and shows that process variation has a detrimental impact on test quality in the form of test escapes. To quantify this impact, a novel metric called test robustness is proposed and to mitigate test escapes, a new process variation-aware test generation method is presented. The method exploits the observation that logic faults that have high probability of occurrence and correspond to significant amounts of undetected bridge resistance have a high impact on test robustness and therefore should be targeted by test generation. Using synthesised ISCAS benchmarks with realistic bridge locations, results show that for all the benchmarks, the method achieves better results (less test escapes) than tests generated without consideration of process variation.**

***Index Terms*—Resistive Bridges, Process Variation, ATPG**

I. Introduction

It is widely recognised that process variation is emerging as a fundamental challenge to IC design with scaled CMOS technology [2]. Process variation has a negative effect on circuit design with respect to performance, power and memory stability [2]. Recent research has reported that process variation has an impact on manufacturing test quality. The work in [3]–[6] considered the effect of process variation on at-speed and delay test, addressing the issues of calculating delay as a function of process variables [5], identification of the longest path [3], [6] and calculation of a test response capture time that tolerates variation [4]. This paper addresses the impact of process variation on static defects with focus on resistive bridging faults. In [7], [8] it was shown that tests for bridging faults can be generated such that they are valid independent of process variation. However, these tests may include many test patterns that do not contribute to detecting unique bridge resistance ranges, leading to an unnecessarily large test set. This paper provides a test generation method that achieves high test quality in the presence of process variation while keeping the test set small. Process variation is considered either across different dies or within each die [2], [9]. In this paper, within-die variation is considered. Process variation is due to sub-wavelength photo-lithography, random dopant distribution and line edge roughness and affects threshold voltage (VT), oxide thickness (TOX) and transistor geometry (width W and length L) [2], [9]. Process variation affect bridge behaviour through two parameters: (1) gate drive strength and (2) logic threshold voltage [1]. Change in behaviour results in test escape and loss of test quality. The test robustness metric [1] quantify the test quality loss by considering the probability of test escape and undetected bridge resistances. To reduce the occurrence of test escapes a process variation-aware test generation method is proposed. The method is guided by the test robustness metric to target logic faults with high impact on test quality. Therefore each test pattern improves test quality, while avoiding unnecessary test patterns.



(a) Example bridge location (b) Nominal parameter behaviour of bridge

Fig. 1. Example bridge location and bridge behaviour

II. Bridge Behaviour Under Process Variation

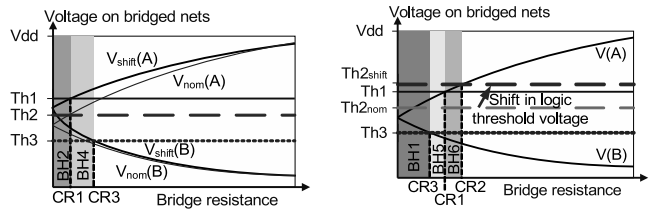
Bridges are unintended resistive connections between two nets. Nets A and B in Fig. 1(a) are bridged with resistance R. Fig. 1(b) shows how the voltages on the bridged nets depend on R, as has been discussed in [10]–[12]. From Fig. 1(b), the voltages on nets A and B can be read for a range of bridge resistances. The logic behaviour depends on how these voltages compare with logic threshold voltages Th1, Th2 and Th3 of driven gate inputs. This example results in three faulty logic behaviours, BH1, BH2 and BH3, corresponding to the resistance ranges between the critical resistances 0Ω, $CR1$, $CR2$ and $CR3$. In the rest of the paper, Fig. 1(b) is called the nominal scenario. For bridging faults, logic fault coverage is misleading as a test quality metric, since a bridge defect cause many different logic faults. Therefore, the bridging fault model proposed in [13] which takes into account process variation but does not use bridge resistance, is limited in its usefulness for the analysis carried out in Section II. It should be noted that the motivation for the model in [13] is fast fault simulation. Instead, this work employs the parametric bridging fault model [10], since it considers the defect resistance. The defect coverage DC (Eq. 1) of a test T on a parametric bridge b with IC parameters c is expressed in terms of Covered Analogue Detectability Interval ($CADI$) and Global Analogue Detectability Interval ($GADI$), representing the covered and

Manuscript received on October 10, 2008, revised on February 1, 2009. This work was supported by EPSRC(UK) under Grant EP/DO57663/1.

U. Ingelsson, B. M. Al-Hashimi and S. Khursheed are with the School of Electronics and Computer Science, University of Southampton, SO17 1BJ, UK. (fax: +44-23-8059-2901 email: {bui05r, bmah, ssk06r}@ecs.soton.ac.uk)

S. M. Reddy is with the University of Iowa, Iowa city, IA 52242, USA. (email: sudhakar-reddy@uiowa.edu)

P. Harrod is with ARM Limited, Cambridge, UK. (email: peter.harrod@arm.com)



(a) Shift in gate drive strength (b) Shift in logic threshold voltage Th2

Fig. 2. Examples of process variation induced behaviour

	Nominal case			GDSS		LTVS		
	BH1	BH2	BH3	BH2	BH4	BH1	BH5	BH6
Th1	$\underline{0\times}$	$\underline{0\times}$	$\underline{0\times}$	$0\times$	$1\surd$	$0\times$	$0\times$	$1\surd$
Th2	$\underline{0\times}$	$\underline{1\surd}$	$\underline{1\surd}$	$1\surd$	$1\surd$	$0\times$	$0\times$	$0\times$
Th3	$1\times$	$1\times$	$0\surd$	$1\times$	$1\times$	$1\times$	$0\surd$	$0\surd$
GDSS gate drive strength shift				LTVS logic threshold voltage shift				

TABLE I
LOGIC BEHAVIOURS FOR THE 3 SCENARIOS

detectable defect resistance respectively, [10], [11]. We investigated, using SPICE simulations, the behaviour of resistive bridges in the presence of process variation. In this section, we present a brief review of the findings. The interested reader is referred to [1] for more details. It was found that variation in parameters such as VT, W, L and TOX affect the following two gate parameters: gate drive strength and logic threshold voltage (Th). These two parameters influence the behaviour of resistive bridges and the subsequent logic values as can be seen from Fig 2. Fig. 2(a) shows the increased voltages on the bridged nets due to a process variation induced change in the gate drive strength of the gate that drives net A (Fig. 1(a)). A new logic behaviour BH4 is introduced, which did not occur in the nominal case (Fig. 1(b)). Similarly, we investigated the impact of logic threshold shift on resistive bridges. Fig. 2(b) shows how a process variation induced increase on Th2 induces two new logic behaviours, BH5 and BH6. This shows that a process variation changes the logic behaviour of a bridge. If the fanout increases, so does the number of variation induced logic faults. Process variation induced logic faults can become test escapes, i.e. undetected faults that correspond to loss of defect coverage. Table I illustrates this using the logic behaviours from Fig. 1(b), Fig. 2(a) and Fig. 2(b). Faulty values are marked '$\times$' and fault-free '$\surd$'. Consider the nominal scenario. Test generation without consideration of process variation would propagate the fault effect through input 1, because Th1 sees faulty Logic-0 for all the logic behaviours BH1, BH2 and BH3 of the nominal scenario (underlined). That means that logic behaviours BH4 and BH6 are test escapes, since they cannot be detected through input 1. This shows that process variation causes test escapes.

$$DC(b,c,T) = \frac{\|CADI(b,c,T)\|}{\|GADI(b,c)\|} \quad (1)$$

III. TEST ROBUSTNESS

This section is a review of test robustness [1], a metric for quantifying the impact of process variation on test quality. The influence of process variation is modelled by a Parameter Value Configuration (PVC), which is a set of parameter values that are influenced by process variation, i.e. values for the gate drive strengths and the logic threshold voltages. A PVC c is used to consider the influence of process variation on a particular unit-under-test, by defect coverage Eq. 1 and by the probability $P(c)$ for the PVC to occur, which varies with the values for the gate drive strengths and logic threshold voltages. To calculate the robustness of a test set T, Eq. 2, a Monte-Carlo simulation is performed for a set of PVCs. Here PP is the set of PVCs and while considering a PVC $c \in PP$, $DC(b,c,T)$ is the defect coverage for bridge b and $P(c)$ is the probability for PVC c. The denominator in Eq. 2 adjusts so that full robustness has the value of one. Experiments have shown that Eq 2 is a useful metric for test quality in the presence of process variation [1]. For a set of bridges B, we calculate the weighted average test robustness WA using Eq. 3 to account for the probability of getting a bridge defect for each bridge location. Here $w(b)$ corresponds to the defect probability for bridge b, defined so that $\sum_{b\in B} w(b) = 1$.

$$Robustness(b,T) = \frac{\sum_{c\in PP} P(c)\cdot DC(b,c,T))}{\sum_{d\in PP} P(d)} \quad (2)$$

$$WA(T) = \sum_{b\in B} Robustness(b,T)\cdot w(b) \quad (3)$$

	μ	σ		μ	σ		μ	σ
L	45nm	5nm	TOXN	17.5$\mathring{A}$	1.5$\mathring{A}$	VTHN	0.471V	0.045V
W	*	5nm	TOXP	18.5$\mathring{A}$	1.5$\mathring{A}$	VTHP	-0.423V	0.045V
* depends on gate						Vdd	0.878V	0.022V

TABLE II
VARIED PROCESS PARAMETERS

As a pre-step to test robustness calculation, a Monte-Carlo simulation is performed. In this process, we compute the logic threshold voltage for each gate input, while varying IC parameters with mean (μ) and standard deviation (σ) values according to Table II. From simulation, we get the mean (μ_{Th}) and standard deviation (σ_{Th}) of the logic threshold voltage for each gate input in the gate library. The μ_{Th} and σ_{Th} values are used to generate PVCs. 11% to 15% standard deviation (σ(Th)) was observed for the logic threshold voltages. Table II is based on data for relevant parameters based on [14]–[16] and 45nm transistor models from [17]. To account for voltage drop in practise, Vdd is varied by 2.5% (0.022V) around 0.878V for a 0.9V nominal Vdd. A standard deviation of 5nm is assumed for the transistor length (L) and width (W) due to line edge roughness affecting the geometry of fabricated transistors [15]. For the thickness of the gate oxide, TOX, 1.5 standard deviation reflects the thickness of one atom layer. For the transistor threshold voltage, VT, 0.045V standard deviation was chosen for random dopant fluctuations based on [16]. Gaussian distribution was used to approximate these variations, supported by observations in [16] and practice in [15]. In this study, the variation on VT, W, L and TOX is assumed to be independent between transistors for the following reasons. The effect of random dopant fluctuations on VT is independent between transistors because of the random number and location of dopant atoms [16]. With respect to W and L, it was observed in [18] that there is no correlation on W and L across transistors due to line edge roughness. Further, we have performed Monte-Carlo simulation with and without W and L correlation and it was found that the simulation

results do not change and therefore W and L are assumed to be independent. For TOX variation, there is lack of published data and the TOX variations are in terms of one atom-layer across different transistors, which is why we have assumed the effect to be independent between transistors. Monte-Carlo simulation was performed using Table II to see the impact of process variation on gate drive strength. Two gates drive the bridged nets, so each pair of gates in the gate library were simulated. In this work, the gate drive strength is represented by the resistance RH (RL) between the Vdd (ground) node and the gate output for the gate that is driving high (low). The mean μ_{RH} (μ_{RL}) and standard deviation σ_{RH} (σ_{RL}) for the resistance are calculated and used to generate PVCs. The standard deviation for RH and RL was found to be in the range 29% to 38% (measured for R=0Ω). The distributions of logic threshold voltage and gate drive strength both show a bell-shaped distribution around the mean. This is in-line with observations of the transistor threshold voltage (VT) [16], which is almost Gaussian. Based on such observations and the central limit theorem (a sum of a large number of similarly distributed and independent random variables will have an approximately Gaussian distribution) we assume that both the logic threshold voltage and the gate drive strength have Gaussian distribution. This assumption is used when PVCs are generated. PVCs are generated by assigning a random number to each logic threshold voltage parameter and gate drive strength parameter. The random number is taken from the Gaussian distribution using the mean and standard deviation values for the considered parameters, RL, RH and Th. These mean and standard deviation values are obtained by Monte-Carlo simulation as discussed above. The probability $P(c)$ of a PVC c is the product of the probability $p(x)$ (Eq. 4) for each parameter value x. This probability $p(x)$ is taken from the Gaussian probability density function according to Eq. 4. The robustness metric (Eq. 2) improves in accuracy with increasing number of PVCs considered in PP. In the experiments presented, PP is limited to 500 PVCs for each bridge. Experiments with smaller and larger sets of PVCs showed that 500 PVCs identify the majority of process variation induced logic faults for the benchmarks considered in the experiments (Section V). The set of considered PVCs is limited to 500 PVCs to limit test robustness calculation time.

$$p(x) = \frac{1}{\sqrt{2 \cdot \pi}} \cdot e^{-\frac{(x-\mu)^2/\sigma^2}{2}} \tag{4}$$

IV. Process Variation-Aware ATPG Method

The proposed process variation-aware test generation PVAA (Process Variation-Aware ATPG) for bridge defects employs a key observation from [1] that logic faults with high probability of occurrence and with a large amount of undetected bridge resistance should be prioritised in test generation. Employing the observation leads to small yet effective test size for bridging faults in the presence of process variation. The test generation flow is shown in Fig. 3 and includes the following two steps: Step 1 gathers all information required to enable PVAA to target logic faults according to the key observation from [1]. This information includes the set of process variation induced

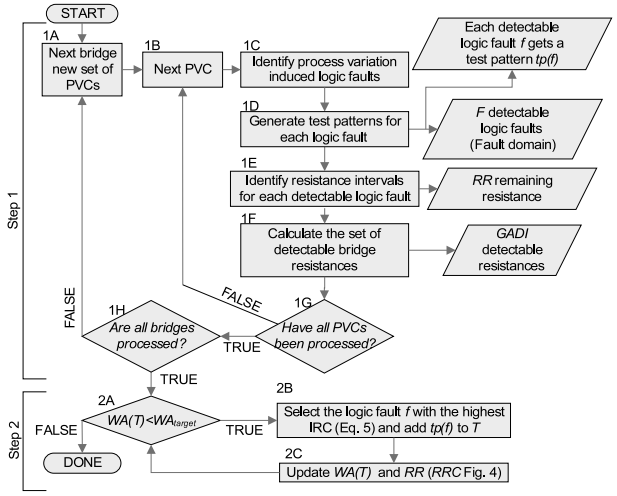


Fig. 3. Flow of the process variation-aware ATPG method (PVAA)

Fig. 4. Algorithm RRC - Robustness ReCalculation

Input: Test pattern tp, Fault domain F
Bridge locations B and weights $w(b)$, $b \in B$
Set of PVCs PP and probability $P(c)$, $c \in PP$
Remaining (not yet covered) resistance ranges $RR(b, f, c)$ for $b \in B$, $f \in F$ and $c \in PP$
Set of detectable defect resistance $GADI(b, c)$ for $b \in B$ and $c \in PP$
Weighted average test robustness $WA(T)$
Output: Updated $WA(T)$, $RR(b, f, c)$

for all faults $f \in F$ detected by tp **do**
  // *b is the bridge location of fault f*
  for all PVCs $c \in PP$ **do**
    $WA(T) := WA(T) + \frac{w(b) \cdot P(c) \cdot \frac{\|RR(b,f,c)\|}{\|GADI(b,c)\|}}{\sum_{d \in PP} P(d)}$
    $covered\_resistance := RR(b, f, c)$
    for all faults g for bridge b **do**
      $RR(b, g, c) := RR(b, g, c) \backslash covered\_resistance$
    end for
  end for
end for

faults that can be detected, the probability for each such logic fault to occur and the range of bridge resistance that would be covered by detecting it. Furthermore, step 1 generates a test pattern for each detectable logic fault, as if it was the only bridge fault in the entire circuit. In step 2, the gathered information is used to select the logic fault to target by test generation, and as a test pattern has already been generated for it, this test pattern is added to the final test set. Through this procedure, the test set is built to efficiently achieve a user defined weighted average test robustness target while keeping the test set small. Step 1 of PVAA begins with a bridge location $b \in B$ with bridge defect probability $w(b)$ which is estimated from the layout of the design (see Section V for more details). Each bridge location b is considered with respect to a set of PVCs called PP, where each PVC $c \in PP$, has an occurrence probability $P(c)$ calculated as discussed in Section III. PVAA processes the complete set of PVCs for each bridge location, as shown by the loop between steps 1B and 1G. In step 1C logic faults are identified for bridge b and PVC c using the fault simulation method in [11]. To determine if the identified logic faults are detectable, deterministic test pattern generation is performed in step 1D, using an ATPG-engine based on a solver for the Boolean satisfiability problem [19]. This procedure

will generate a test if one exists and otherwise the fault is undetectable and subsequently ignored. The set of detectable logic faults form the fault domain F. Through this process, each detectable logic fault f becomes associated with a test pattern $tp(f)$. Step 1E determines the bridge resistance range that would be covered by detecting each logic fault and logs this resistance range in the set RR (Remaining Resistance). This is followed by step 1F that determines the $GADI(b,c)$-value for defect coverage calculation using Eq. 1. The above steps are repeated for all the bridges and their respective sets of PVCs as shown by the loop between 1A and 1H. Step 2 is then initiated by PVAA after processing all the bridges. The aim of step 2 is to identify and select a small number of test patterns such that the user-specified weighted average robustness WA_{target} is achieved. In step 2B, PVAA selects a logic fault f which has the highest incremental robustness contribution (IRC, Eq. 5). Targeting logic fault with a high IRC value is the recommendation of the observation that a fault f that has a high probability of occurrence ($\sum_{c\in PP} P(c)$) and correspond to a large amount of undetected bridge resistance ($\sum_{c\in PP} RR(b,f,c)$) has a high impact on test robustness. The bridge defect probabilities $w(b)$ are used in Eq. 5 to compare logic faults for different bridge locations. The logic fault f with the highest IRC marks the preferred logic fault to target and therefore, the test pattern $tp(f)$ (generated in step 1D) that detects that logic fault is added to the final test set T. This is followed by step 2C that updates $WA(T)$ and RR after a test pattern has been added to the test set T. $WA(T)$ is the weighted average test robustness of test set T and RR is the set of remaining resistances, those that are not covered by T. Fig. 4 shows the algorithm RRC (Robustness ReCalculation) for step 2C. A newly selected test pattern tp is fault simulated to determine the detected set of logic faults as shown in line 1. The detection of logic fault f implies that the weighted average test robustness $WA(T)$ is increased by $IRC(f,b)$ (Eq. 5), which is implemented by the loop starting on line 3 and the summation on line 4. $RR(b,f,c)$ is updated on lines 5-8 so that resistances that are covered for the first time by tp are counted as covered only once, because logic faults for the same bridge can have overlapping resistance ranges.

$$IRC(f,b) = \sum_{c\in PP} \frac{w(b)\cdot P(c)\cdot \frac{\|RR(b,f,c)\|}{\|GADI(b,c)\|}}{\sum_{d\in PP} P(d)} \qquad (5)$$

V. Experimental Results

Experiments were performed on ISCAS85 and -89 benchmark circuits, synthesised with the default options of Synopsys Design Compiler to a 45nm gate library [20] consisting of 33 different characterised gates, including AOI, OAI, FA, MUX and XOR, with transistor models from [17]. The ISCAS89 benchmarks are sequential circuits with full scan-chains. Cadence Encounter was used to identify bridge locations and defect probabilities from layout. Layout information is not required in PVAA, but used to conduct realistic experiments. Bridge defect probabilities were estimated from the coupling capacitance between nets. A pair of nets with a high coupling capacitance has high defect probability and vice versa. To show the impact of process variation on test sets that are generated unaware of process variation, we have employed a single-Vdd version of the test generator presented in [12].

A. Analysis: Process variation impact on test quality

To demonstrate the impact of process variation on test quality, we calculated the robustness for test sets of 12 benchmarks in two scenarios: (1) Nominal and (2) with process variation. Table III, columns 2-7 show the results. Column 2 and 3 show the number of gates and bridges and column 4, marked TPs, show the test set size. In the nominal case the test sets have full defect coverage. Column 5 shows the fault domain, i.e. the number of detectable faults identified in the nominal case. Column 6 and 7 show how the test sets perform under process variation. The fault domain of all benchmarks has increased. Some of the faults escape the tests leading to low weighted average robustness WA as demonstrated in column 7. For example, 31 test patterns for design C432 achieve 100% defect coverage in the nominal scenario. Under process variation, there are 1719 detectable logic faults, but some of them escape the test, which reduces WA to 0.916. From Table III, it can be seen that process variation has a negative impact on test quality, because the weighted average test robustness is <1, indicating escapes for all the considered tests.

B. Test quality gain through process variation-aware ATPG

To mitigate the test escapes and to improve test robustness, we employ PVAA. Two experiments were performed. In the first experiment (Table III columns 8-13) we kept the original test set of each circuit and augmented it with additional test patterns to reduce test escapes for three weighted average test robustness targets (WA_{target} 0.96, 0.98 and 0.995). Columns marked "WA" show the weighted average robustness of the test sets, evaluated with other PVCs than those that were used in test generation. The PVCs for evaluation were generated with another sequence of random numbers regarding the parameters (logic threshold voltages and gate drive strengths, see Section III). The evaluated WA value varies around the intended WA_{target}. For example, the test generated for design C432 and WA_{target} 0.995 achieved $>$0.995 for PVCs used in test generation, but when evaluated with another set of PVCs, it achieved WA 0.993 (column 13). The columns marked "TPs" represents the total number of test patterns (original test set plus added test pattern). As can be seen from column 13, it is possible to achieve higher WA in comparison to column 7 by including additional test patterns. For example, WA has increased from 0.899 to 0.995 for design S838 at the expense of 17 additional test patterns. Table III shows that significant increase in test patterns is required to achieve the WA_{target}=0.995 when compared with WA_{target}=0.96. When the test robustness is high and only faults with low impact on robustness remain, the benefit of adding a single test patterns is low. To increase the test robustness further, many test patterns are required. In the second experiment, we started with empty test sets and used PVAA to achieve the required WA. The results are shown in Table III, column 14-19. The required WA for a given design can be met using a smaller test set (TPs) than the test sets shown in column 9-13. For example, design S9234 needs 84 test patterns (column 17) to achieve test robustness of 0.98 compared to 101 test

				Section V-A Evaluating original test set			First experiment of Section V-B Original test set augmented by PVAA						Second experiment of Section V-B Test set generated by PVAA					
				Nominal	With process variation		target 0.96		target 0.98		target 0.995		target 0.96		target 0.98		target 0.995	
Design	Gates	Bridges	TPs	Domain	Domain	WA	WA	TPs	WA	TPs	WA	TPs	WA	TPs	WA	TPs	WA	TPs
C432	175	36	31	363	1719	0.916	0.958	32	0.978	39	0.993	53	0.964	27	0.977	33	0.992	49
C499	211	107	37	1560	9990	0.958	0.971	38	0.981	40	0.994	54	0.960	26	0.977	37	0.994	53
C880	297	96	45	1981	10654	0.978	0.978	45	0.980	47	0.995	73	0.956	43	0.980	59	0.995	97
C1355	307	111	46	2912	24008	0.965	0.965	46	0.979	52	0.993	74	0.961	34	0.977	44	0.994	71
C1908	278	154	55	2042	10691	0.971	0.971	55	0.980	59	0.994	76	0.961	42	0.981	58	0.995	78
C2670	481	154	68	2434	9499	0.943	0.961	74	0.981	92	0.994	136	0.960	62	0.980	85	0.994	134
C3540	1001	695	138	9317	50586	0.992	0.992	138	0.992	138	0.995	156	0.958	62	0.980	89	0.995	147
S641	175	44	20	649	4976	0.938	0.964	22	0.979	27	0.994	43	0.959	27	0.980	36	0.994	50
S838	265	28	11	343	1357	0.899	0.958	14	0.980	19	0.995	28	0.959	13	0.979	18	0.994	27
S1488	704	873	124	9613	53925	0.991	0.991	124	0.991	124	0.995	129	0.958	56	0.980	76	0.995	119
S5378	1365	727	168	10024	50619	0.984	0.984	168	0.984	168	0.994	228	0.965	82	0.979	103	0.995	191
S9234	1015	318	89	4593	28354	0.956	0.959	91	0.978	101	0.994	143	0.959	56	0.980	84	0.994	131

TABLE III
EXPERIMENTAL RESULTS

patterns (column 11). The smaller number of test patterns is due to the fact that PVAA targets logic faults that (in being detected) have the largest contribution to robustness. Since the original test set used as a starting point in the first experiment is not generated with this objective, it is likely that the original test patterns are less effective in achieving high robustness, leading to a higher number of test patterns. This is true for a large majority of cases, except for example C880, WA_{target}=0.98 and WA_{target}=0.995, which are due to accidental detection. With accidental detection we mean that a test pattern is generated for a particular logic fault, but is found to be effective at detecting other logic faults as well. The computational time for PVAA range from 30 minutes for design C432 to 22 hours for design S5378. Over 50% of the time is spent using the solver for the Boolean Satisfiability problem [19] for ATPG. We believe that the computation time can be significantly reduced if a more efficient commercial ATPG-engine is available. To give an insight into how PVAA improves test quality, compare the original test set in column 4 with the test set in columns 14-19 (Table III). For all designs but C880, PVAA achieves higher WA than the original test sets with a smaller test set size. For example, the original test for S9234 achieved WA=0.956 with 89 test patterns and PVAA achieved WA=0.959 with 56 test patterns (improvement of 36%). For C880, the test set size is increased by PVAA, but higher WA is still achieved. The results show that test sets generated without consideration of process variation are compromised in terms of test quality in the presence of process variation. To regain the lost test quality, additional test patterns can be generated (Table III column 8-13). Alternatively, high test quality in the presence of process variation can be obtained by generating a new test set (Table III column 14-19). The latter option leads to a smaller test set size.

VI. CONCLUSION

This paper investigated the impact of process variation on test quality of bridging faults. It has been shown that process variation influences two parameters (logic threshold voltage and gate drive strength) which affects the logic behaviour of bridging faults leading to test escape. To quantify the impact of process variation on test quality, a metric called test robustness has been presented. The metric guides a novel process variation-aware ATPG method, which target the logic faults that have the largest impact on test quality. Experimental results show that for all considered benchmarks, the proposed method achieves better results (less test escapes) than tests generated without consideration of process variation.

REFERENCES

[1] Ingelsson *et al.*, "Variation aware analysis of bridging fault testing," in *ATS*, Nov. 2008, pp. 206–211.

[2] Bhunia *et al.*, "Process variations and process-tolerant design," in *VLSID*, Jan. 2007, pp. 699–704.

[3] Iyengar *et al.*, "Variation-aware performance verification using at-speed structural test and statistical timing," in *ICCAD*, Nov. 2007, pp. 405–412.

[4] Devanathan *et al.*, "Variation-tolerant, power-safe pattern generation," *D&TC*, vol. 24, no. 4, pp. 374–384, Jul. 2007.

[5] Lu *et al.*, "PARADE: parametric delay evaluation under process variation," in *ISQED*, Mar. 2004, pp. 276–280.

[6] Lu *et al.*, "Longest-path selection for delay test under process variation," *TCAD*, vol. 24, no. 12, pp. 1924–1929, Dec. 2005.

[7] Chen *et al.*, "An unified fault model and test generation procedure for interconnect open and bridges," in *ETS*, May 2005, pp. 22–27.

[8] Favalli *et al.*, "High quality test vectors for bridging faults in the presences of IC's parameters variations," in *DFT*, Sep. 2007, pp. 448–456.

[9] Nassif, "Modeling and analysis of manufacturing variations," in *CICC*, May 2001, pp. 223–228.

[10] Renovell *et al.*, "The concept of resistance interval: a new parametric model for realistic resistive bridging fault," in *VTS*, Apr. 1995, pp. 184–189.

[11] Engelke *et al.*, "Simulating resistive bridging and stuck-at faults," *TCAD*, vol. 25, pp. 2181–2192, Oct. 2006.

[12] Khursheed *et al.*, "Bridging fault test method with adaptive power management awareness," *TCAD*, vol. 27, no. 6, pp. 1117–1127, Jun. 2008.

[13] Favalli *et al.*, "Bridging fault modeling and simulation for deep submicron CMOS ICs," *TCAD*, vol. 21, no. 8, pp. 941–953, Aug. 2002.

[14] Kuhn, "Reducing variation in advanced logic technologies: approaches to process and design for manufacturability of nanoscale CMOS," in *IEDM*, Dec. 2007, pp. 471–474.

[15] Oldiges *et al.*, "Modeling line edge roughness effects in sub 100 nanometer gate length devices," in *SISPAD*, Sep. 2002, pp. 131–134.

[16] Asenov, "Random dopant induced threshold voltage lowering and fluctuations in sub-0.1 μm MOSFETs," *TED*, vol. 45, no. 12, pp. 2505–2513, Dec. 1998.

[17] "Predictive technology model," Released by Arizona State University, Apr. 2008. Url: http://www.eas.asu.edu/˜ptm

[18] Leunissen *et al.*, "Line edge roughness: experimental results related to a two-parameter model," *Microelectronic Engineering*, vol. 73–74, pp. 265–270, 2004.

[19] "zChaff Boolean Satisfiability problem solver," Mar. 2007. Url: http://www.princeton.edu/˜chaff/zchaff.html

[20] "OSU FreePDK," Oklahoma State University, 2008. Url: http://avatar.ecen.okstate.edu/projects/scells/OSUFreePDK.php