

Test Cost Reduction for Multiple-Voltage Designs with Bridge Defects through Gate-Sizing

Saqib Khursheed, Bashir M. Al-Hashimi
School of ECS
University of Southampton, UK.
Email: {ssk06r, bmah}@ecs.soton.ac.uk

Peter Harrod
ARM Limited
Cambridge, UK.
Email: peter.harrod@arm.com

Abstract—Multiple-voltage is an effective dynamic power reduction design technique. Recent research has shown that testing for resistive bridging faults in such designs requires more than one voltage setting for 100% defect coverage; however switching between several supply voltage settings has a detrimental impact on the overall cost of test. This paper proposes an effective Gate Sizing technique for reducing test cost of multi-Vdd designs with bridge defects. Using synthesized ISCAS benchmarks and a parametric fault model, experimental results show that for all the circuits, the proposed technique achieves 100% defect coverage at a single Vdd setting; in addition it has a lower overhead than the recently proposed Test Point Insertion technique in terms of timing, area and power.

Index Terms—Gate Sizing, Test Cost, Resistive Bridging Faults, Multiple-Vdd designs, Design for Testability

I. INTRODUCTION

Resistive bridging faults (RBF) represent a major class of defects for deep submicron CMOS and have received increased attention on modeling, simulation and test generation [1]–[9]. A bridge is defined as an un-wanted metal connection between two lines of the circuit, which may deviate the circuit from its ideal behaviour. Typically, a multi-Vdd design has a set of discrete supply voltage settings it can switch between depending on the current workload and power saving mode. Manufacturing test needs to ensure that such a design operates correctly over the entire set of supply voltage settings, while keeping the overall cost of test low.

It has been shown in [3] and more recently in [9] that the fault coverage of a test set targeting resistive bridging faults can vary with the supply voltage used during test. This means that, depending on the operating Vdd setting, a given RBF may or may not affect correct operation of the design. Consequently, to ensure high fault coverage for a design that needs to operate at a number of different Vdds, it is necessary to perform testing at more than one Vdd to detect faults which manifest themselves only at particular Vdds. It was shown in [9] that the majority of circuits (8 out of 12) require testing at more than one voltage setting to achieve 100% defect coverage, which means that the ATE (Automatic Test Equipment) will have to switch between different voltage settings to apply the test. Switching between different Vdd settings during test is not a trivial task, and therefore a large number of Vdd

settings required during test can have a detrimental impact on the overall cost of test. Consequently it would be desirable to keep the number of Vdd settings required during test to a minimum. The only investigation that addresses test cost reduction through minimizing the number of test Vdds for multi-Vdd designs has been presented in [9]. It demonstrates that Test Point Insertion (TPI) can be used to reduce the number of Vdd settings during test, without affecting the defect coverage of the original test, thereby reducing test cost. One drawback with the TPI scheme [9] is that it does not guarantee a single Vdd test and usually results in more than one test Vdd setting. This paper proposes a new and more effective technique for reducing test cost of multi-Vdd designs with bridge defects. It targets resistive bridges that cause faulty logic behaviour to appear at a non-desired test Vdd setting and uses Gate Sizing (GS) to expose the same physical resistance at the preferred test Vdd. The number of test voltages is then reduced, minimizing test cost. We show that it is possible to achieve 100% defect coverage using a single test Vdd setting unlike the case with TPI. Furthermore, this paper evaluates the timing, area and power cost of the proposed technique and comparison with TPI scheme [9] shows that the proposed gate sizing algorithm achieves the same objective at lower cost in terms of timing, area and power.

The paper is organized as follows: Section II gives an overview of resistive bridge defects and their behaviour in the context of multi-Vdd design. The motivation for using gate-sizing is discussed in Section III. In Section IV we present the proposed gate-sizing algorithm. Experimental setup and results are reported in Section V, and finally Section VI concludes the paper.

II. PRELIMINARIES

To explain the proposed gate sizing algorithm, it is necessary to discuss some concepts related to resistive bridging faults and their behaviour in the context of multi-Vdd designs; these concepts are briefly outlined in this section. A typical bridge fault behavior is illustrated in Fig. 1. Fig. 1-A shows a resistive bridge, D1 and D2 are the gates driving the bridged nets, while S1, S2 and S3 are successor gates; moreover, the output of D1 is driven high and the output of D2 is driven low. The

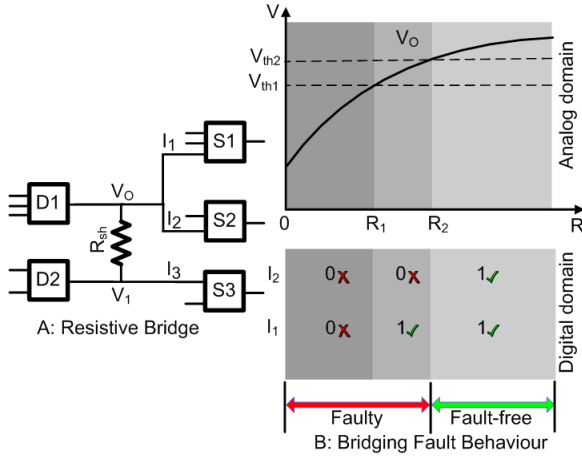


Fig. 1. Bridge fault example

dependence of the voltage level on the output of D1 (V_O) on the equivalent resistance of the physical bridge is shown in Fig. 1-B (based on Spice simulation with $0.12\mu\text{m}$ library). To translate this analog behavior into the digital domain, the input threshold voltage levels V_{th1} and V_{th2} of the successor gates S1 and S2 have been added to the V_O plot. The logic threshold of a gate input is defined as the input voltage at which the output reaches half of the supply voltage, while other inputs of the gate are at non-controlling value(s). For each value of the bridge resistance R_{sh} , the logic values read by inputs I_1 and I_2 can be determined by comparing V_O with the input threshold voltage of the corresponding input. These values are shown in the second part of Fig. 1-B (marked as “digital domain”). Crosses are used to mark the faulty logic values and ticks to mark the correct ones. It can be seen that, for bridges with $R_{sh} > R_2$, the logic behavior at the fault site is fault-free (all inputs read the correct value), while for bridges with R_{sh} between 0 and R_2 , one or more of the successor inputs are reading a faulty logic value. A number of bridge resistance intervals can be identified based on the corresponding logic behavior. For example, bridges with $R_{sh} \in [0, R_1]$ exhibit the same faulty behavior in the digital domain (all successor inputs read the faulty logic value), similarly, for bridges with $R_{sh} \in [R_1, R_2]$, successor gate S2 reads the faulty value, while S1 reads the correct value, and finally for $R_{sh} > R_2$ all the successor gates read the correct logic value. Consequently, each interval $[R_i, R_{i+1}]$ corresponds to a distinct logic behavior occurring at the bridge fault site.

Next, consider Fig. 2-A, which shows the relationship between the voltage on the output of gate D1 (Fig. 1-A) and the bridge resistance for two different supply voltages V_{ddA} and V_{ddB} [1], [3]. Fig. 2-A also shows how the analog behavior at the fault site translates into the digital domain. Using similar explanation (as for Fig. 1-B), we can see that two distinct Logic Faults LF1 and LF2 can be identified for each Vdd setting.

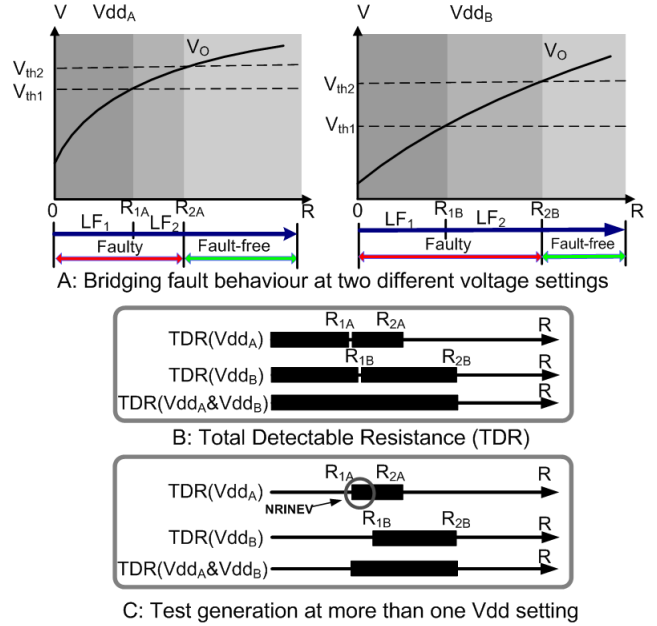


Fig. 2. Effect of supply voltage on bridge fault: Analog and Digital domain

However, because the voltage level on the output of D1 does not scale linearly with the input threshold voltages of S1 and S2 when changing the supply voltage, the resistance intervals corresponding to LF1 and LF2 differ from one supply voltage setting to another [9]. Fig. 2-B shows the Total Detectable Resistance (TDR) for the LFs detected at two voltage settings separately and combined as well. Furthermore, this means that in the case of Fig. 2-B, the complete range of physical defects can be covered alone at V_{ddB} .

Next let us consider a case where logic fault LF1 covering a resistance range ($0 \rightarrow R_{1A}$ at V_{ddA} , $0 \rightarrow R_{1B}$ at V_{ddB}) becomes undetectable, in which case Fig. 2-C shows the detectable resistance intervals at two voltage settings. For a certain bridge the *Essential Vdd* setting is the one at which the highest resistance interval is detected, which is V_{ddB} in this case. From a test generation point of view, essential Vdd(s) has to be included in test generation as the highest resistance interval, of certain bridge(s), exists at the essential voltage setting(s). On the other hand, non-essential voltage settings (V_{ddA} in this case) are included in test generation only because some non-redundant intervals are detectable at non-essential voltage setting(s), these intervals are referred to as NRINEV (Non-Redundant Intervals at Non-Essential Vdd) [9]. One such NRINEV interval is highlighted in Fig. 2-C. Therefore test generation tool uses extra Vdd settings to cover such NRINEV intervals in order to achieve 100% defect coverage.

The only investigation to reduce the number of test voltages for resistive bridging faults is presented in [9], which utilizes Test Point Insertion (TPI). Test points are used to provide additional controllability and observability at the fault-

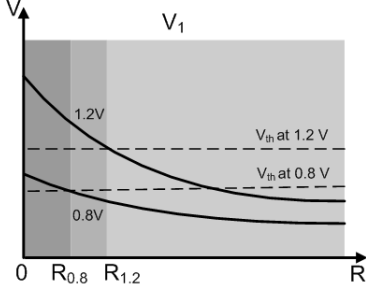


Fig. 3. Resistance range detection at different voltage settings

site to detect NRINEV intervals at essential Vdd, which are otherwise redundant (at essential Vdd) and therefore helps reducing the number of test Vdd(s). From Fig. 2-C it means that the resistance range marked as NRINEV is covered at essential Vdd (Vdd_B) by providing additional controllability and observability using test points. TPI has shown a reduction in the number of test Vdd(s) but it has some limitations. Experimental results presented in [9] show that TPI is unable to reduce to a single test Vdd for the majority of circuits (10 out of 13 circuits require more than one test Vdd). This is because TPI cannot reduce the number of test Vdd(s) below the number of essential Vdds. This can be understood from the following explanation. In Fig. 1-A, the gates used for driving the bridge (D1, D2) and the driven gates (S1, S2, S3), influence the number of essential Vdd(s) in a circuit. For the same circuit, assume that D1 is driving high and D2 is driving low, the output of D2 (V_1) on the equivalent resistance of the physical bridge is shown in Fig. 3, which shows that higher resistance range is covered at 1.2V (non-preferred test Vdd) than at 0.8V (preferred test Vdd). This means that 1.2V becomes the essential test Vdd and TPI has to include it for 100% defect coverage, as the resistance range covered at 1.2V cannot be covered at 0.8V. Other than that TPI has some well-accepted limitations (not limited to the scheme proposed in [9]) that to increase the fault coverage and to reduce test cost it may be necessary to introduce extra overhead on timing, area and power as is the case with [10]–[12]. For instance, a test point (control point) added to the critical path may violate the timing, which restricts its usage in critical paths. Furthermore, the overall area and power of the design may increase due to the large number of test points.

III. IMPACT OF GATE SIZING ON TEST VDD REDUCTION

We investigate the effect of gate sizing on the behaviour of resistive bridging faults and how it can be used to propagate faulty behaviour such that a higher physical resistance is exposed at a single Vdd setting (thereby reducing the number of essential test Vdds to one). The limitations of TPI can be addressed by adjusting the driving gates (D1, D2) or driven gates (S1, S2, S3) at the fault-site. The driving/driven gates can be adjusted by two approaches, which include the following:

- Varying logic threshold of driven gates,
- Increasing drive strength of driving gates.

A. Varying logic threshold of driven gates

In the first case, the logic threshold of the driven gate is adjusted such that a higher resistance range is detectable at the lowest Vdd setting. This concept is further elaborated in Fig. 4, where the logic threshold of the same gate inputs (as for Fig. 3) is reduced by gate-resizing. Therefore, the highest resistance interval is exposed at the lowest Vdd setting, which facilitates test generation at the lowest Vdd setting.

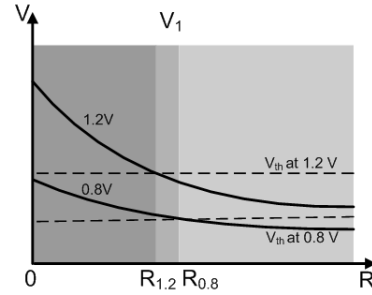


Fig. 4. Resistance range detection after adjusting logic thresholds of the driven gates

The logic threshold can be adjusted by altering the width/length of the PMOS/NMOS transistor connected to the particular gate input, or by using the body bias effect. For an inverter it is given by [13]:

$$V_{in} = \frac{V_{DD} + V_{tp} + V_{tn} \sqrt{\frac{\beta_n}{\beta_p}}}{1 + \sqrt{\frac{\beta_n}{\beta_p}}} \quad (1)$$

where, V_{in} is the voltage at the input of the gate, V_{DD} is the supply voltage, V_{tp} is the threshold voltage of the PMOS transistor, V_{tn} is the threshold voltage of the NMOS transistor.

$$\beta = \mu C_{ox} \left(\frac{W}{L} \right) \quad (2)$$

where, β is the MOS transistor gain factor, μ is the effective surface mobility of the carriers, C_{ox} is the gate oxide capacitance. From (1), it can be noticed that a variation in W_p and W_n can alter the logic thresholds of a given gate input. This concept was used to conduct some experiments using 0.12 μ m ST Microelectronics library. The transistor widths (connected to the gate input of interest) are varied to reduce the logic threshold, while operating at 0.8V Vdd. For all the considered cases, the targeted logic threshold was -80 mV or lesser to detect the fault at the lowest Vdd setting. The resultant widths for some of the transistors are shown in table I, where the first column shows the input of the particular gate for which the logic threshold is varied. The second main column shows the (W_p/W_n) ratios of the original design (as specified in the library) and that of the re-designed gates. The last column

TABLE I
TRANSISTOR WIDTH VARIATION FOR ALTERING LOGIC THRESHOLD

Gate (Input)	W_p/W_n *		Logic Th. Diff. @ 0.8V Vdd
	Original	Re-designed	
4 Input NAND (C)	0.46/0.64	0.24/3.09	-80 mV
4 Input NAND (B)	0.46/0.64	0.24/5.22	-100 mV
5 Input AND-NOR (B)	0.94/0.64	0.24/6.79	-140 mV

* Width is in μm

shows the difference in logic thresholds as a result of gate re-sizing. It can be noticed that for all the cases the ratio between (W_p/W_n) is much lower than usually suggested design rule ratio of (W_p/W_n) $\approx 1.5 - 2.5$ [13]. The ratios (in table I) result in unbalanced charging/discharging time (t_{phl} and t_{plh}), require more power to switch the gate, and add to the overall area of the design. For these reasons, variation in transistor width is not considered for altering the logic thresholds of the driven gates. We also examined body biasing to vary the logic threshold but preliminary examination did not provide sufficient variations. For the cases considered, it resulted in ≈ 20 mV variation in logic threshold (operating at 0.8V Vdd) at the targeted gate input. Therefore it was not further pursued.

B. Increasing drive strength of driving gates

The drive strength of the gates driving the bridged nets can be adjusted to increase the voltage on the bridged nets (V_1 in Fig. 1-A). This increase in voltage level can help expose maximum resistance at the lowest Vdd setting thereby reducing the number of essential Vdd settings; additionally it can also be used to cover NRINEV intervals at the lowest Vdd setting. This concept is illustrated by Fig. 5, it should be noted that Fig. 5 shows the same pair of bridged nets as shown in Fig. 3 (derived from Fig. 1-A, where D1 is driving high and D2 is driving low), i.e., the logic thresholds of the driven gates remain the same. In Fig. 5 it can be noticed that the voltage level V_1 has increased such that $R_{0.8V} > R_{1.2V}$, by increasing the drive strength of the gates driving the bridge. This means that test generation will favor 0.8V over 1.2V, thereby reducing the number of test Vdd(s) and removing 1.2V as an essential Vdd.

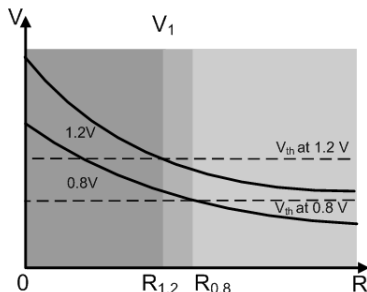


Fig. 5. Resistance range detection after adjusting the drive strength of the gates driving the bridge

The drive current of an NMOS transistor operating in *non-saturation* region is [13].

$$I_{ds} = \beta \left[(V_{gs} - V_t) V_{ds} - \frac{V_{ds}^2}{2} \right] \quad (3)$$

where, I_{ds} is the drain-source current, β is the gain factor expressed by (2), V_{gs} represents the gate-source voltage and V_t is the transistor threshold voltage.

From 3, it can be noticed that the drive current I_{ds} is directly proportional to the gain factor β (in saturation and non-saturation modes), which in turn is directly proportional to the W/L of the transistor. Thus replacing a gate with another having higher value of β (for transistors feeding the output) results in higher drive strength. This is feasible since, different versions of functionally equivalent gates are usually available in the gate library.

IV. GATE SIZING ALGORITHM FOR SINGLE TEST VDD

This section presents the gate sizing algorithm used to reduce the number of test Vdd setting(s). The proposed scheme consists of two phases: gate(s) identification and replacement, during which the method identifies the gates that should be replaced followed by test generation phase on the modified circuit to get a single Vdd test set.

The algorithm is shown in Fig. 6, it starts off by test generation (test generation follows the method presented in [9]) and marks all the bridges, which require test generation at higher than the lowest Vdd setting. All such bridges are placed in *TargetBridgeList* and all the driving gates of the respective bridges are marked as potential candidates for gate replacement. The algorithm then solves a minimum set covering problem that identifies the minimum number of driving gates such that all the bridges are covered, the selected gates are placed in *minGatesList*, this is shown in line 3. The algorithm then takes each selected gate in *minGatesList* and replaces it with another having higher drive strength from the gate library, this step is shown in lines 4-6. After updating the netlist, the algorithm generates a test set considering complete bridge list and finally returns with an updated netlist and a new test set.

It should be noted that the minimum set covering technique is useful for area minimization and has shown positive results for almost all the cases considered. However in a few cases increasing the drive strength of a gate may make the fault redundant (un-detectable) at all Vdd settings. Next, we explain the scenario where gate resizing by minimum set cover may make the fault redundant. This concept is explained using Fig. 7, which shows a fault-site with driving gates D1 (driving high), D2 (driving low) and S1, S2 are successor gates. Consider Fig. 7-A and assume that the output of D1 is a “Weak one” and the output of D2 is a “Strong zero”. This results in S1 reading a faulty logic value at its input (shown as 1/0), while S2 reads the correct logic value in both good/faulty circuits. Furthermore, assume that the fault effect is propagated

- 1: Run test generation using the netlist
- 2: Compute *TargetBridgeList*
// Mark the bridges that require test at additional
// voltage setting(s)
- 3: Compute minimum number of driving gates *minGatesList* across complete *TargetBridgeList* by solving a minimum set cover
- 4: **for all** *minGatesList* **do**
- 5: Replace the selected gate with another having higher drive strength.
- 6: **end for**
- 7: Generate Test Set for the modified netlist using complete bridge list.
- 8: **return** (*netlist*, *Test Sets*)

Fig. 6. Gate Sizing Algorithm

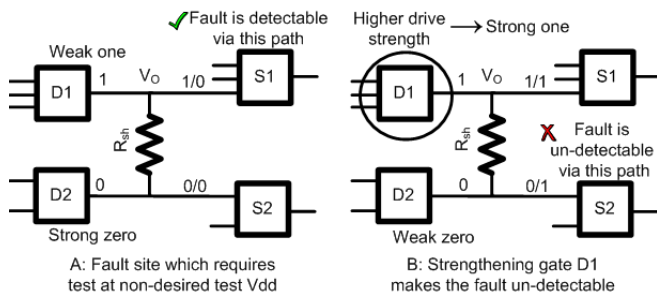


Fig. 7. Fault Redundancy due to gate selection by minimum set cover

to the primary output via S1 and results in test generation at a non-desired voltage setting. Next consider Fig. 7-B, which shows that gate D1 is selected by the minimum set cover and is replaced by a gate with higher drive strength. Due to this change in drive strength, D1 outputs a “Strong one” and D2 outputs a “Weak zero”, which results in S2 reading a faulty logic value (shown as 0/1) but this faulty logic value does not reach to the primary output and therefore the fault becomes un-detectable. In such cases, the drive strength of both the driving gates (D1 and D2) is adjusted such that higher amount of physical resistance is exposed at the lowest Vdd setting (as shown in Fig. 5) while ensuring that the fault is detectable. Therefore it is worth mentioning that for a few bridges, gate replacement and test generation may be repeated for fault detection at the lowest Vdd setting.

V. EXPERIMENTAL RESULTS

The proposed technique for reducing test Vdds is validated using ISCAS’85 and ’89 full scan circuits. We have conducted two experiments. The first experiment applies the proposed Gate Sizing (GS) technique to reduce Vdd setting(s) during test. The second experiment evaluates the cost and compares the Timing, Area and Power performance of the proposed GS technique with the only available test cost reduction technique, i.e., TPI [9]. The benchmark circuits are synthesized using

TABLE II
RESULTS OF PROPOSED GATE SIZING TECHNIQUE (GS) AND COMPARISON WITH TPI [9]

CKT.	No. of		Test Vdd(s)			Gates	
	Gates	Bridges	Orig.	TPI	GS	GS	TPI
c432	93	1,094	All*	All	0.8v	2	0
c1355	226	6,563	All	0.8v	0.8v	4	10
c1908	205	7,986	1.2v, 0.8v	1.2v, 0.8v	0.8v	3	0
c2670	269	10,000	All	1.2v, 0.8v	0.8v	6	19
c3540	439	10,000	All	1.0v, 0.8v	0.8v	7	7
c7552	731	9,998	All	0.8v	0.8v	1	1
s344	62	469	1.2v, 0.8v	0.8v	0.8v	1	1
s382	74	1,146	1.2v, 0.8v	0.8v	0.8v	2	5
s386	63	1,625	All	1.2v, 0.8v	0.8v	7	4
s838	149	5,737	All	0.8v	0.8v	14	28
s5378	578	9,933	All	1.0v, 0.8v	0.8v	9	9
s9234	434	10,000	All	1.0v, 0.8v	0.8v	6	2
s15850	1578	10,000	All	0.8v	0.8v	8	3

*All = 0.8v, 1.0v, 1.2v

ST Microelectronics 0.12 μ m cell library. The setup uses non-feedback bridges only and an exhaustive bridge list is generated by considering all possible pairs of nets in the netlist, up to a maximum of 10,000 pairs. This is done to increase the total number of bridges and, therefore, create more challenging test cases for all the circuits. The benchmarks used, total number of gates and extracted bridges for each circuit are tabulated in the first two main columns of Table II.

The outcome of the first experiment is also shown in Table II, where the third main column (labeled as Test Vdd(s)) tabulates total number of test Vdd setting(s) for each of the original design (labeled as Orig.), by TPI [9] (labeled, TPI)¹ and by the proposed gate sizing technique (labeled, GS). As can be seen the proposed GS technique is able to achieve 100% defect coverage at a single Vdd. This is unlike TPI, which requires two or more Vdd setting for most of the circuits to achieve the same defect coverage. Moreover, TPI is unable to reduce any test Vdd in case of c432 and c1908. The last main column of Table II (labeled as Gates) shows the number of gates replaced by gate sizing (GS) technique and the number of test points (control/observation points) added by TPI². The number of gates replaced by the GS technique ranges from 1-14, while TPI has added up to 28 test points. The computation time of the proposed GS technique is lesser than TPI as it uses a straightforward minimum set cover for gates replacement, while TPI uses a complex control point minimization algorithm to reduce the total number of test points.

The second experiment compares the timing, area and power performance of the original design (labeled as Orig), proposed Gate Sizing (labeled as GS) and the TPI (labeled as TPI).

¹It should be noted that TPI results may vary from those reported in [9] because of using different logic threshold values

²The number of test points is the sum of control and observation points

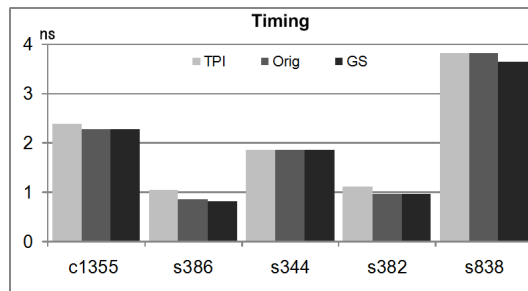


Fig. 8. Timing performance comparison

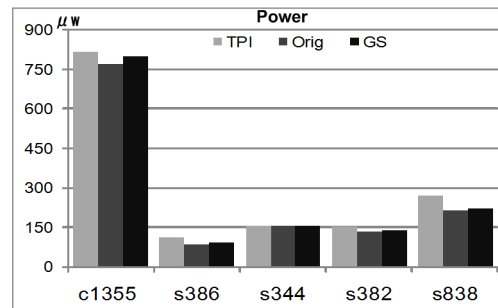


Fig. 10. Power performance comparison

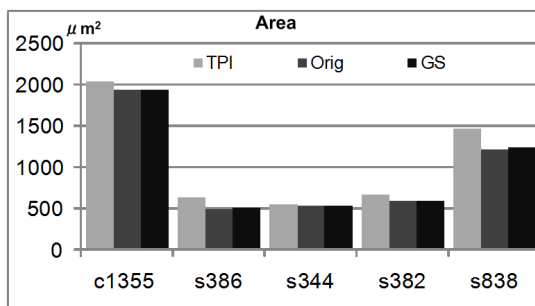


Fig. 9. Area performance comparison

Synopsys design compiler is used to calculate these parameters for the three designs. Fig. 8 shows the timing performance, as can be seen the proposed GS technique has little effect on the timing performance when compared to the original design, this is because it replaces small number of gates. For majority of circuits shown in Table II, it has replaced less than 2% of the total number of gates. It should be noted that for some circuits the proposed GS technique has reduced timing compared to the original design due to larger and faster gates. This is unlike the case with TPI, where the timing has increased because of test points in the critical path. Similarly, comparison of area overhead is shown in Fig. 9 for the three designs. It can be noticed that the proposed Gate Sizing (GS) technique results in a slight area increase in comparison to original designs, however it is less than TPI for all the circuits. Finally, power comparison is shown in Fig. 10, which shows the sum of leakage and dynamic power consumed by each design. As can be seen the proposed gate sizing technique slightly increases the power budget in comparison to the original design, however it is less than the TPI scheme for all designs other than s344 which results in equal power consumption. High power consumption of TPI is because of additional switching activity and leakage power of added test points. In the case of GS, switching activity does not change (in comparison to original design) but load capacitance and leakage power increases due to bigger and faster gates.

VI. CONCLUSION

This paper has proposed gate sizing to reduce test cost of multi-Vdd designs with bridge defects, by reducing the number of test voltage settings. It has been shown that it is possible to achieve 100% defect coverage using a single Vdd test setting. This represents an improvement on the recently proposed TPI scheme [9] which mostly requires two or more test Vdd settings to achieve complete defect coverage. Furthermore the proposed gate sizing technique has little effect on timing, area and power when compared with the original design (prior to gate sizing) and outperforms the TPI scheme in terms of these three parameters. Our future work will focus on identifying gates for resizing during design time to reduce its impact on timing closure.

ACKNOWLEDGMENTS

This work is supported by EPSRC (UK) under grant no. EP/DO57663/1.

REFERENCES

- [1] M. Renovell, P. Huc, and Y. Bertrand, "Bridging fault coverage improvement by power supply control," in *VTS*, Apr. 1996, pp. 338–343.
- [2] V. R. Sar-Dessai and D. M. H. Walker, "Resistive bridge fault modeling, simulation and test generation," in *ITC*, Sep. 1999, pp. 596–605.
- [3] P. Engelke, I. Polian, M. Renovell, B. Seshadri, and B. Becker, "The pros and cons of very-low-voltage testing: an analysis based on resistive bridging faults," in *VTS*, Apr. 2004, pp. 171–178.
- [4] P. Engelke, I. Polian, M. Renovell, and B. Becker, "Simulating resistive bridging and stuck-at faults," *IEEE TCAD*, Oct. 2006.
- [5] M. Renovell, F. Azais, and Y. Bertrand, "Detection of defects using fault model oriented sequences," *JETTA*, vol. 14, pp. 13–22, Feb. 1999.
- [6] T. Maeda and K. Kinoshita, "Precise test generation for resistive bridging faults of CMOS combinational circuits," in *ITC*, Oct. 2000, pp. 510–519.
- [7] G. Chen *et al.*, "An unified fault model and test generation procedure for interconnect open and bridges," in *ETS*, May 2005, pp. 22–27.
- [8] P. Engelke, I. Polian, M. Renovell, and B. Becker, "Automatic test pattern generation for resistive bridging faults," *JETTA*, pp. 61–69, Feb. 2006.
- [9] S. Khurshid *et al.*, "Bridging fault test method with adaptive power management awareness," in *IEEE TCAD*, Jun. 2008, pp. 1117–1127.
- [10] M. Abramovici, M. Breuer, and A. Friedman, *Digital Systems Testing and Testable Designs*. IEEE Press, 1998.
- [11] N. A. Touba and E. J. McCluskey, "Test point insertion based on path tracing," in *VTS*, Apr. 1996, pp. 2–8.
- [12] I. Pomeranz and S. Reddy, "Design-for-testability for path delay faults in large combinational circuits using test points," *IEEE TCAD*, vol. 17, no. 4, pp. 333–343, Apr. 1998.
- [13] N. H. E. Weste and K. Eshraghian, *Principles of CMOS VLSI design: a systems perspective*. Addison-Wesley Publishing Co., 1994.