High Quality Testing of Grid Style Power Gating

Vasileios Tenentes*, Saqib Khursheed†, Bashir M. Al-Hashimi*, Shida Zhong*, Sheng Yang*
*ECS, University of Southampton, UK. Email: {V.Tenentes, bmah, sz3a13, sheng.yang}@ecs.soton.ac.uk
†Electrical Engineering & Electronics, University of Liverpool, UK. Email: S.Khursheed@liverpool.ac.uk

***Abstract*—This paper shows that existing delay-based testing techniques for power gating exhibit fault coverage loss due to unconsidered delays introduced by the structure of the virtual voltage power-distribution-network (VPDN). To restore this loss, which could reach up to 70.3% on stuck-open faults, we propose a design-for-testability (DFT) logic that considers the impact of VPDN on fault coverage in order to constitute the proper interface between the VPDN and the DFT. The proposed logic can be easily implemented on-top of existing DFT solutions and its overhead is optimized by an algorithm that offers trade-off flexibility between test-application-time and hardware overhead. Through physical layout SPICE simulations, we show complete fault coverage recovery on stuck-open faults and 43.2% test-application-time improvement compared to a previously proposed DFT technique. To the best of our knowledge, this paper presents the first analysis of the VPDN impact on test quality.**

***Index Terms*—power gating, dft, power-distribution-network, fault coverage, grid style power gating**

I. Introduction

Design-for-testability (DFT) is a design technique that increases the testability of Integrated Circuits (ICs) against fault models that mimic the behaviour of physical defects [12]. *Fault coverage* is used as a quantification of test quality and high quality testing is substantial to avoid yield loss [2]. Power gating is a low power design technique for ICs that assures the viability of high performance and energy efficient electronic devices at sub-100-nm CMOS technologies [14]. It utilizes transistors as power-switches of logic blocks supply voltage to reduce leakage power and power consumption during periods of inactivity. Power switches are susceptible to defects and their high quality testing is crucial for achieving low power and anticipated performance benefits of power-gated ICs [8].

Power switches are implemented as header or footer switches in either *fine-grain* or *coarse-grain* design styles. A *fine-grain* style incorporates a power switch within each logic cell simplifying power gating synthesis through existing EDA tools [4]. However, the *coarse-grain* design style is more popular choice and the focus of this paper because the power switches feed a block of logic with less area overhead and higher robustness against process variations. Coarse grain power gating is implemented in two different design styles by facilitating either a ring or a grid network of power switches. In *ring style*, power switches are placed at a ring externally to the power-gated block (Figure 1a). In *grid style* [4], [11], power switches are distributed throughout the power-gated region (Figure 1b) forming a grid between the power-distribution-networks (PDNs): the *supply voltage V_{dd} PDN* (SPDN) and the *virtual voltage V_{Vdd} PDN* (VPDN). When comparing these two styles [4], the ring is the only option for power gating IP blocks, while the grid style is scalable to large designs and the only option that supports *hybernation*, the ability to store the state of the power-gated IC in retention registers. For these reasons, the grid style is deployed very often in industrial designs and is the focus of this work.

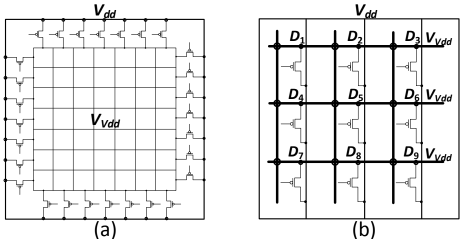

Fig. 1. (a) Ring style and (b) grid style power gating schemes

Although previous works have considerably advanced the DFT architectures for power switches, when considering the transistor fault models of stuck-opens and stuck-shorts [5]–[10], [13], [16], this is the first work that analyses the PDNs impact on fault coverage. This problem is described in Section II. In Section III the parameters that affect the problem are identified and fault coverage loss of up to 70.3% is shown. To this end, Section IV presents a DFT logic that can be automated designed on-top of existing DFT solutions through an algorithm that offers trade-off flexibility between test-application-time (TAT) and hardware overhead. Finally, Section V evaluates the performance and the trade-offs of the proposed method, while Section VI concludes the paper.

II. Background & Motivation

Figure 2 presents the DFT architecture for delay-based testing against stuck-open faults on header power switches that was proposed in [6], [9], [10]. The basic idea is the segmentation of the power switches set at m segments under test (SUTs) of size L number of power switches [6]. A test is conducted by observing an ideal observation point V_{Vdd}, marked in Figure 2, without considering the delays of the VPDN. The test process is shown in Figure 3. During initialization phase, the control logic fully discharges the V_{Vdd} node by using the discharge transistors [10]. During application phase, a single SUT S_i is waken-up by the control logic by deasserting the $sleep_i$ signal. At capture moment, the NAND gate logic output is captured at the "Test Result" memory cell by the assertion of the test clock [9], the frequency of which depends on the segment size L. The captured value indicates if the observation point V_{Vdd} was sufficiently charged at the capture moment. Test clock frequency is selected based on the *observable charging delay* M of the V_{Vdd} point. That delay is the time elapsed from the

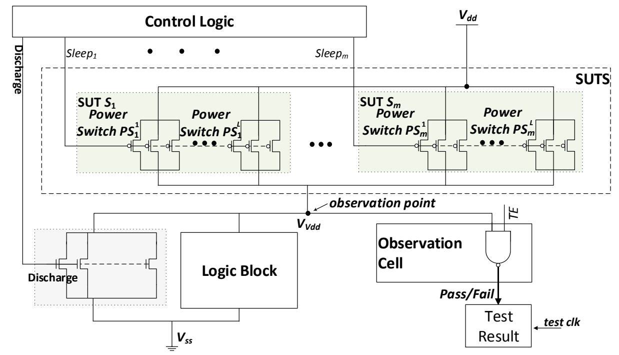

Fig. 2. DFT for ring style power gating with ideal observation point [9]

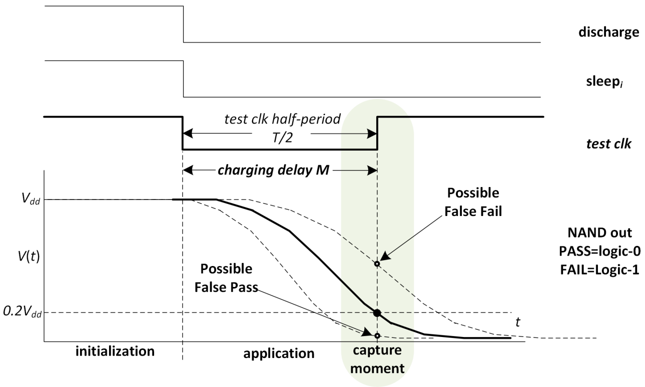

Fig. 3. Sensitivity of test process on charging delay M of V_{Vdd}

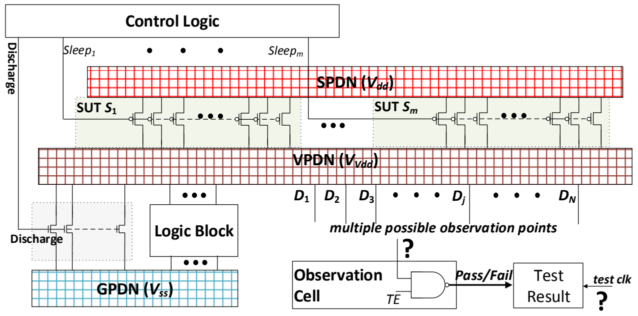

Fig. 4. Power gating DFT considering PDNs

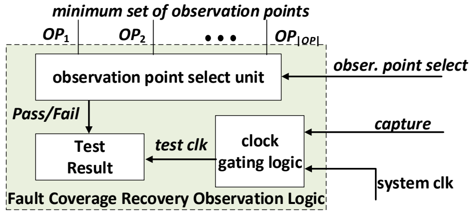

Fig. 5. Diagram of proposed fault coverage recovery block

start of the application phase to the *capture moment*, when the transient voltage at the NAND gate reaches logic-0 value at the fault free scenario testing of a SUT. For analog-to-digital conversion, the voltage level of $\leq 0.2V_{Vdd}$ is used as logic-0, and voltage $\geq 0.8V_{Vdd}$ as logic-1, because when considering process variation with $\pm 3\sigma$ variation effects, logic threshold voltage of a gate is within 20%-80% of V_{dd} [18].

In this paper we examine the test environment presented in Figure 4 that considers the RC components of the supply voltage (SPDN), the ground voltage (GPDN) and the virtual voltage (VPDN). The power gating style is the grid style shown in Figure 1b. In this environment, the observation NAND gate may observe any of the observation points D_j on the VPDN that are shown in Figure 1b and in Figure 4. Based on this setup, we show in Section III that contrary to the ideal V_{Vdd} (Figure 2), where the observable charging delay M is unique, at the VPDN consideration (Figure 4) the observable charging delay M_{ij} is affected by two *additional to the segment size* L factors that interact with the RC network during testing:

- the observation point D_j that observes the delay
- the SUT S_i that is waken-up

Note in Figure 3 how two hypothetical scenarios (dashed curves) with observable charging delay that deviates from the one used to calculate test frequency may affect fault coverage. We show in Section III that this fault coverage loss may reach up to 70.3% for a SUT when VPDN is considered. Since multiple test clocks, one for every SUT, is not a practical solution, this paper presents an enhanced DFT observation logic that restores the fault coverage loss with the usage of just the system clock. This observation logic, shown in Figure 5, enables the fault coverage restoration by skipping an appropriate number of clock cycles through clock-gating of the system clock before activating a suitable observation point. An algorithm is utilized to select the minimum number of observation points OP to achieve 100% fault coverage. This logic is combined with the DFT of Figure 4 to form the solution shown in Figure 7.

III. Analysis of VPDN Impact on Fault Coverage

We synthesize using a 90nm library, one of the IWLS benchmark circuits [1], the *ethernet* of 157.5K gate equivalents size (a gate equivalent corresponds to a two input NAND gate), with grid style coarse grain power gating using header power switches. The constraint during the physical synthesis of the PDNs was to achieve $\leq 5\%$ IR drop with 2048 power switches. Using Synopsys STAR-RCXT, we extracted two SPICE models: one for Figure 4 that includes the PDNs (MPDN) and the other for Figure 2 without the PDNs (MNOPDN). The operational voltage is $V_{dd} = 1.2$V. To speed-up SPICE simulations and focus on the effects of the PDNs on fault coverage, every logic gate not related to DFT is modeled with an RC network. The RC values are extracted by design's library as the median values of all input combinations. For large industrial designs parallel rail analysis techniques such as those reported in [17] can be deployed. Finally, the MPDN is monitored at 200 observation points D_j (shown as dots in Figure 6a), while the MNOPDN is only monitored at the V_{Vdd} node. The simulations showed that SPDN and GPDN do not affect power switches testing since they do not switch during that time. However, the VPDN impacts it considerably.

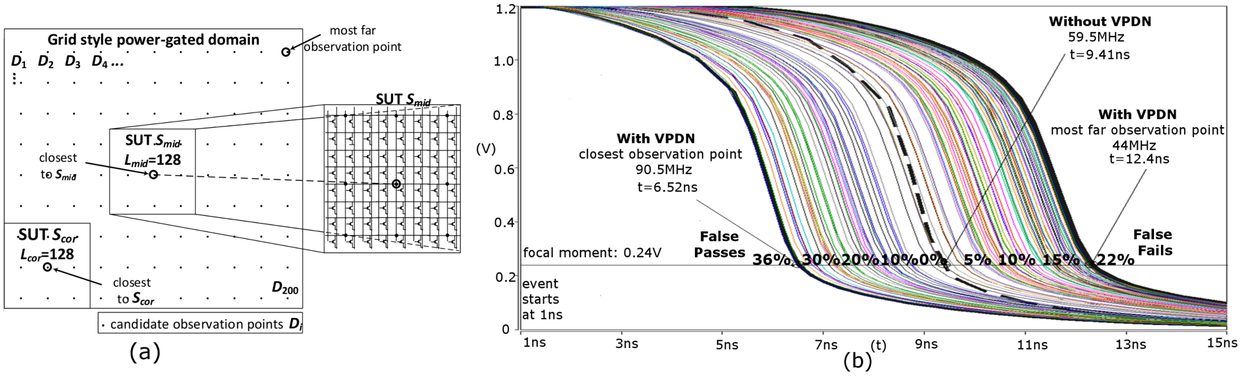

Fig. 6. (a) Setup of case study, (b) transient voltage level showing charging delay deviations and fault coverage loss for various observation points D_j

A. VPDN charging delay depends on observation point D_j

Firstly, we focus on a single SUT S_{mid} at the centre of the physical layout with size $L_{mid} = 128$ power switches, as shown in Figure 6a. We simulate the MPDN for every observation point D_j. These results are shown in Figure 6b. Each curve depicts the transient voltage level (Figure 3) of the output of a NAND gate (Figure 4), when observing one out of the 200 D_j observation points. We repeat this simulation for the MNOPDN. The voltage level of the V_{Vdd} node is shown as the dashed curve *"Without VPDN"* in Figure 6b. Note, how the curves of the MPDN, are deviating from *"Without VPDN"* curve. When it arrives early, it belongs to observation points close to the SUT S_{mid}. The earliest of them, belongs to the closest D_j (shown in Figure 6a) and arrives 52% sooner than the curve *Without VPDN*. When it arrives late, it belongs to observation points far from the SUT S_{mid}. The latest one of them, belongs to the most far observation point (shown in Figure 6a) and arrives 25% later than curve *Without VPDN*. As expected, the choice of the observation point D_j impacts considerably the observable charging delay of the VPDN.

Next, we compute the fault coverage loss of the observable charging delay per D_j because of the mechanism described in Figure 3. Fault coverage is negatively affected by *false passes* (FP), the percentage of devices that even if they are faulty they pass test, and *false fails* (FF), the percentage of fault-free devices that fail test. The *test frequency* that achieves 100% fault coverage on the MNOPDN is required. The half-period of that frequency (see Figure 3) is the charge delay of the MNOPDN simulation in Figure 6b. For the case at hand, it was found 59.5MHz. For every observation point D_j we measure the FPs and FFs by gradually injecting stuck-open faults at the power switches of the MPDN until a correct result of the test. The percentage of masked faults provides the fault coverage loss. The first injection characterizes the type of loss (FP or FF). The results are marked in Figure 6b: the selection of an observation point very close to the SUT S_{mid} (the highlighted curve to the left of the curve *Without VPDN*) leads to FP of 36%, while the selection of an observation point too far from the SUT S_{mid} leads to FF of 22% (highlighted curve to the right). An arbitary D_j selection leads to average FP of 19.6% and average FF of 7.8%. Note in Figure 6b the monotonic relationship between the fault coverage loss and the deviation of the charging delay for every observation point D_j from the capture moment of the test clock. In Section IV we exploit this relationship to select observation points that minimize these deviations and consequently the fault coverage loss.

B. VPDN charging delay depends on SUT S_i

The observable charging delay at a particular observation point depends on the location of the SUT S_i at the physical layout. To show that, we repeat the experiment of Section III-A for another SUT S_{cor} with the same segment size $s_{cor} = 128$, located at a corner of the power gated domain (Figure 6a). The observable charging delays for the S_{mid} SUTs from the closest and the most far observation points are $M_{mid} = 5.52$ns and $M_{mid} = 11.4$ns respectively. For the S_{cor} case, they were found to be $M_{cor} = 4.3$ns for the closest observation point, considerably different compared to the 5.52ns of the S_{mid}, and $M_{cor} = 11.1$ns for the most far observation point. The fault coverage loss FP and FF results for the S_{cor} are 48% and 20% (worst case) and 19.8% and 7% (average case) respectively. We showed that the observable VPDN charging delay M depends on the SUT S_i and the observation point D_j that observes it. Hereafter, it will be denoted as M_{ij}.

C. Fault coverage loss for various segment sizes L

Next, we consider a single observation point option at a corner of the design (the D_j at the top right corner in Figure 6a). For various SUT segmentations of the 2048 power switches, $L \times m = 32 \times 64, 64 \times 32, 128 \times 16$ and 256×8, the charging delay M is computed by using the MNOPDN (model without consider the VPDN) and the results are shown in column M of Table I. Then, the observable charging delay M_{ij} for each SUT is computed through the MPDN model. The deviations of M_{ij} from M are shown under columns '-' and '+' next to the column M. Finally, through fault injections we gather the FP and FF results also shown in Table I. The first two columns contain the SUT size L and the SUTs number m. The next three contain the charging delay M and its observable deviation '-' and '+'. The last four columns contain the average FP and FF results and the worst FP and FF results. For example, by ignoring the VPDN, for $m = 8$ segments of $L = 256$ power switches, the charging

TABLE I
OBSERVABLE CHARGING DELAY FOR VARIOUS SEGMENTATION SETUPS

L	m	$-$	M	$+$	FP %	FF %	worst FP %	worst FF %
32	64	3.9	33.3	3.7	2.3	2.8	50.3	19.3
64	32	3.7	17.8	3.5	6.9	5.7	43.7	28.1
128	16	2.9	8.4	3.0	8.6	9.6	46.9	21.9
256	8	1.9	4.2	1.0	13.4	12.6	70.3	29.6

delay without considering the VPDN is $M = 4.2$ns. Yet, when VPDN is considered the observable charging delay arrives 1.9ns sooner than M for a SUT that suffers from 70.3% FP and 1.0ns later than M for another SUT that suffers from 29.6% FF. The average fault coverage degradation is 13.4% FP and 12.6% FF for all SUTs. Note that this fault coverage loss increases while SUT size increases rendering previous DFT methods inapplicable for high speed testing of power switches. These results clearly motivate the importance of considering a VPDN interface between the SUT and the DFT logic.

IV. PROPOSED VPDN-AWARE DFT ARCHITECTURE

The VPDN-aware DFT architecture provides on-chip control over the parameters that affect the deviations of the VPDN observable charging delay in order to restore fault coverage: the observation point D_j that observes that delay and the SUT S_i that charges the VPDN. Additional control over the system-clock clock-gating is required in order to generate the appropriate capture edge. The proposed architecture utilizes the block introduced in Figure 5 to restore fault coverage and is shown shaded in Figure 7. It consists of three major blocks:
Fault Coverage Recovery Observation Logic (FCR): This block is responsible for both the generation of the test clock and the activation of an observation point OP_j that achieve 100% fault coverage out of a set of minimum observation points OP. The basic idea is that one out of the multiple rising edges generated by clock gating the system clock, the c_{ij}^{th}, achieves high fault coverage when observes the result of a SUT S_i through the observation point OP_j. The high fault coverage is met, if the observable charging delay M_{ij}, exhibits negligible deviation with that rising edge. We call this condition *compatibility* and is described in Section IV-A. This unit latches system clock as long as the *capture* signal is zero and the multiplexer OP-MUX selects the appropriate observation point OP_j indicated by the *opselect* value. A shift register stores the test result, when *capture* is asserted.
FCR Controller (FCRC): This block is responsible for generating the control signals *opselect* and *capture* for the FCR unit. Firstly, the Observation Point Controller (OPC) generates on-chip the *opselect* signal to control the activation of a single observation point for a particular SUT S_i. The number of observation points OP that are integrated on-chip are selected to be minimum by the algorithm of Section IV-A. As a result, many SUTs S_i require the activation of the same OP_j and the data of this correspondance are suitable for Run-Length (RL) compression (which also requires minimum decompression logic). The compressed data are stored in a *registers file* (OP-REG), each register of which, stores the *opselect* value and its repetition number of SUTs. Secondly, the Capture Edge Controller (CEC), generates on-chip the *capture* signal that controls the clock gating of the system clock. A counter counts down c_{ij} clock rising edges before asserting *capture*. Skip cycles c_{ij} are also stored in a registers file (C-REG). They are also stored at an RL compressed form: each register stores a c_{ij} value and its repeatition number. That compression performs well for encoding long sequences that repeat the same value (many pairs (S_i, OP_j) require the same skip cycles value c_{ij}). Note that c_{ij} values are computed for a clock frequency f. When process variations affect f, the proposed fault coverage is also affected. To consider these effects, when *configure* is asserted, CEC unit loads at the C-REG a variations-aware configuration $C(f)$ generated by running the algorithm of Section IV-A on the final frequency.
Observation Cells (OCs): NAND observation cells, shown as an oval shape in Figure 7, that are attached on a minimum set of observation points OP selected by the algorithm of Section IV-A that achieve 100% fault coverage. Voltage monitoring alternatives like those reported in [15] can be deployed.

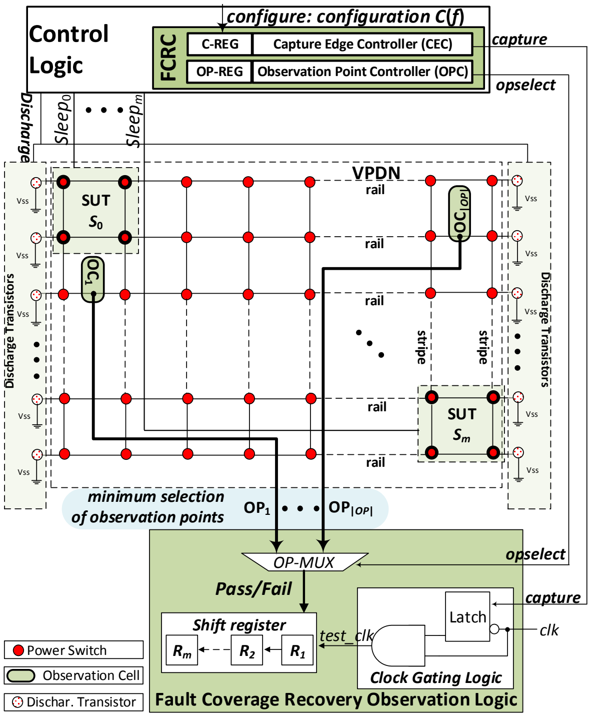

Fig. 7. Proposed DFT architecture

Figure 8 depicts the generation flow of this logic. The spice model MPDN generated after physical layout that includes PDNs is required. Next, power switches are segmented at segments of size L and observation points D_j are injected into the MPDN model. The algorithm, described below, is applied in order to select a minimum set OP of them.

A. Minimum set of observation points selection

The Algorithm I in Figure 8 selects a minimum observation points set OP. Firstly, the *observable charging delay* M_{ij} is computed for every pair of SUT S_i and candidate observation point D_j through simulations of the MPDN. Next, the most

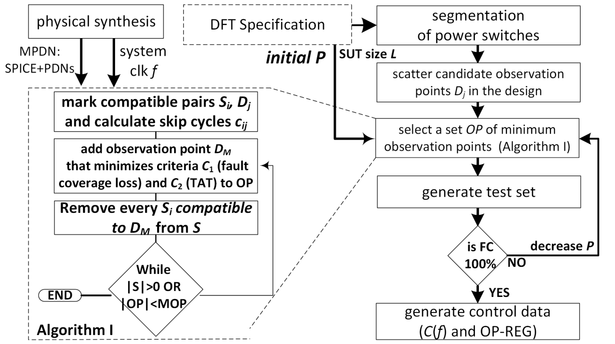
physical synthesis
MPDN:
SPICE+PDNs
system
clk f
DFT Specification
initial P
SUT size L
segmentation
of power switches
scatter candidate observation
points D_j in the design
select a set OP of minimum
observation points (Algorithm I)
generate test set
is FC
100%
NO
decrease P
YES
generate control data
(C(f) and OP-REG)
mark compatible pairs S_i, D_j
and calculate skip cycles c_ij
add observation point D_M
that minimizes criteria C_1 (fault
coverage loss) and C_2 (TAT) to OP
Remove every S_i compatible
to D_M from S
While
|S|>0 OR
|OP|<MOP
END
Algorithm I

Fig. 8. Simulation flow of proposed method

close rising edge of the system clock to M_{ij} is identified and evaluated for its fault coverage loss using its deviation from M_{ij}. For a pair (S_i, D_j) we define as *charging delay deviation* of the N^{th} rising edge from the focal moment M_{ij} as: $d(N, M_{ij}) = |N \cdot T - M_{ij}|$, where T is the period of the system clock. Next, we define for a pair (S_i, D_j) to be *compatible* at the N^{th} rising edge of the system clock when that deviation is $d(N, M_{ij}) < P \cdot M_{ij}$ (**compatibility condition**), where $0 \leq P \leq 1$ is the *maximum charging delay deviation value*, a parameter given by the designer that serves as an input constraint: for P values close to '0', pairs with low deviation are characterized as *compatible*. Even if results of a P value do not offer 100% fault coverage (FC), the monotonic relationship between charging delay deviation and fault coverage loss assures that a smaller P value will achieve lower FC loss. When the above *compatibility condition* is met, we set *skip cycles* $c_{ij} = N$, the number of system-clock clock-gated cycles beforing capturing the test response.

A selection algorithm is deployed to assure that for every SUT S_i there will be an observation point in OP set that will be compatible and also minimize the number of observation points $|OP|$ (and consequently hardware overhead), the fault coverage loss and TAT. The algorithm selects every element of OP set based on the critera below:
C_1: Select a set $MIN(D)$ with the observation points D_j that exhibit the minimum *average charging delay deviation* for all their compatible SUTs still remaining in S.
C_2: Among those D_j selected by criterion C_1, select the one that requires the minimum *average number of skip cycles* for all its compatible SUTs (even those droped from S).
Every new observation point selection follows these criteria. After the selection of an observation point, its compatible SUTs are droped from set S. The algorithm terminates when the set S is empty. If the designer has set the *more observation points* parameter, MOP, to a value greater than the minimum number of observation points $|OP|$, the algorithm selects MOP number of observation points. This property offers trade-off between hardware overhead and TAT. After the selection of the set OP the *test generation* process assigns every SUT S_i to be tested through its *compatible* observation point OP_j that requires the minimum skip cycles c_{ij}. Finally, the OP-REG and $C(f)$ memory data are generated by compressing the tests (triplets of S_i, OP_j and c_{ij}) using RL compression [3]. Note that the above algorithm requires the system clock frequency in order to evaluate the observation points quality. To restore possible fault coverage loss caused by process variations that affect system clock, the configuration $C(f)$ must be computed after the system clock frequency is known. For systems with high speed clocks this approach can completely restore the fault coverage loss even when the observation points have been pre-selected with a slightly different system clock frequency. However, for systems with slow system clock frequency the pre-selected observation points will be incompatible with the final system clock. An initial selection of more observation points (MOP parameter) and the expansion of configuration $C(f)$ to include the assignment of SUTs to observation points (data of OP-REG) solve this issue.

V. Simulation Results

In this section we validate, through SPICE simulation, the performance of the proposed DFT architecture (Figure 7) when VPDN is considered for various power switches segmentation setups. Also, for various parameters of the flow (Figure 8), we show the available trade-offs on FC, TAT and hardware overhead. Finally, we compare the results of a recent work that does not consider the VPDN for testing power switches [9] with and without the proposed fault coverage recovery logic. The simulation setup is the same as that of Section II and the operational frequency of the benchmark is f = 1GHz.

Firstly, we examine various segmentation setups $L \times m = 32 \times 64, 64 \times 32, 128 \times 16$ and 256×8 by varying the maximum charging delay deviation parameter $P = 0.2, 0.1, 0.08, 0.06, 0.04, 0.02,$ and 0.01, starting from the largest towards the smallest value. The parameter MOP (More Observation Points) is set to zero in order to trigger the selection of a minimum set of observation points. Results for the first P value that achieves 100% FC are shown in Table II, and include the number of selected observation points $|OP|$, the size in bits of the OP-REG and the configuration $|C(f)|$ (that is stored in C-REG, Figure 7), the TAT and the FC (FC=100% - "False Passes %" - "False Fails %", Figure 6b) In every setup there is at least a P value that restores FC at 100% with very low hardware overhead. The selected observation points number was in the range of [1 5] and the register files requirements (OP-REG + $|C(f)|$) are very low, in the range of [38 95] flip flops.

Next, for one case from Table II, the $L \times m = 256 \times 8$, the selected observation points number $|OP|$, the achieved FC, the configuration size $|C(f)|$ and the OP-REG size are shown in Figure 9. As expected, while P values decrease more observation points are selected and the FC increases. At the same time the size of OP-REG increases from 3 to 20 flip flops, while the size of the configuration $|C(f)|$ remains small in the range of [12 20] bits. For $P = 0.06$ the algorithm achieves 100% FC with $|OP| = 5$ observation points. Next, Figure 10 presents a trade-off between hardware overhead and TAT for more observation points $MOP = 6, 7, 8$. These values trigger the selection of more than the minimum $|OP| = 5$

TABLE II
RESULTS FOR MINIMUM OBSERVATION POINTS SELECTION

Basic Info		Hardware Overhead			Performance	
L	m	$\|OP\|$	OP-REG	$\|C(f)\|$	TAT (ns)	FC (%)
32	64	1	6	80	7.76E+03	100
64	32	3	18	77	1.93E+03	
128	16	4	16	30	4.3E+02	
256	8	5	18	20	1.38E+02	

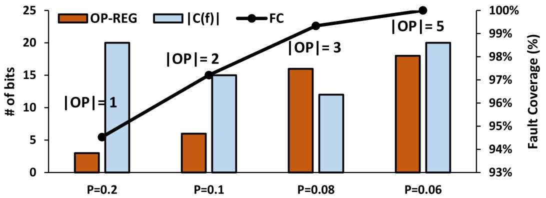

Fig. 9. Fault coverage vs. hardware overhead trade-off

observation points. While the $|OP|$ increases from 5 to 8 the bit requirements remain almost unaffected (OP-REG increases, but $|C(f)|$ decreases). The same time TAT decreases 26.6% compared to the case of $|OP| = 5$, clearly indicating that more observation points can be spared for less TAT, a trade-off observed at all the simulations.

Finally, we evaluate the proposed method when it is applied on top of [9], a technique that does not consider the PDNs for testing power switches. The results in Table III show that the proposed method (labelled as " [9]+Prop.") restores fault coverage to 100%. Note that the proposed architecture is able to select an observation point close to the SUT. For this reason it achieves up to 43.2% less TAT than [9]. For the cases in Table III the proposed architecture requires the following logic on top of the architecture of [9]: [38 95] flip flops for the register files, [0 16] observation cells, 5 counters for the control logic and a 16:1 MUX (worst case). This additional logic leads to 42%-58% more hardware overhead compared to [9] which is less than 0.3% of a design with 157.5K gate equivalents.

VI. CONCLUSIONS

We showed that delay-based testing of power switches must consider the VPDN to deliver 100% fault coverage. To this end, we proposed a new fault coverage recovery DFT architecture that is selected through an algorithm (Section IV) and considers the VPDN to achieve multiple objectives: high fault coverage, low TAT and minimum hardware overhead. The simulations results showed complete fault coverage recovery (Table II), and trade-offs on hardware overhead (Figure 9) and TAT (Figure 10) as well as 43.2% TAT improvement when the proposed DFT is applied on top of a recently proposed DFT (Table III) with minimum hardware overhead of less than 0.3% percent of a design with 157.5K gate equivalents.

ACKNOWLEDGMENTS

This work is supported by EPSRC (UK) under grant no. EP/K000810/1 and by the Department of Electrical Engineering and Electronics, University of Liverpool, UK.

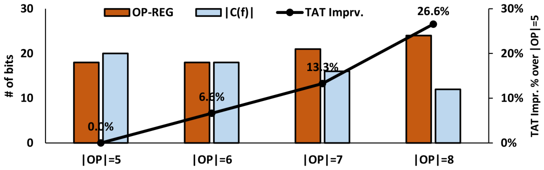

Fig. 10. TAT vs. flip flop requirements by selecting more observation points

TABLE III
PROPOSED METHOD RESULTS WHEN APPLIED ON TOP OF [9]

L	m	Freq. (MHz)		TAT (ns)			FC (%)	
		[9]	[9]+Prop.	[9]	[9]+Prop.	Impr. (%)	[9]	[9]+Prop.
32	64	15	1000	8.60E+03	6.88E+03	20.0	97	100
64	32	28		2.18E+03	1.66E+03	23.7	94	
128	16	59.5		5.54E+02	3.58E+02	35.5	91	
256	8	118		1.44E+02	0.82E+02	43.2	87	

REFERENCES

[1] IWLS'05 circts., online: http://www.iwls.org/iwls2005/benchmarks.html.

[2] M. Abramovici, M. Breuer, and A. Friedman, *Digital Systems Testing and Testable Design*. Piscataway, NJ, USA: IEEE Press, 1998.

[3] K. Chakrabarty, V. Iyengar, and A. Chandra, *Test Resource Partitioning for System-on-a-Chip*. Springer US, 2002.

[4] D. Flynn, R. Aitken, A. Gibbons, and K. Shi, *Low Power Methodology Manual: For System-on-Chip Design*. NY, USA: Springer-Verlag, 2007.

[5] P. Girard, N. Nicolici, and X. Wen, *Power-Aware Testing and Test Strategies for Low Power Devices*.

[6] S. Goel, M. Meijer, and J. de Gyvez, "Testing and diagnosis of power switches in socs," in *Proc. Eur. Test Symp.*, May 2006, pp. 145–150.

[7] H.-H. Huang and C.-H. Cheng, "Using clock-vdd to test and diagnose the power-switch in power-gating circuit," in *Proc. IEEE VLSI Test Symp.*, May 2007, pp. 110–118.

[8] X. Kavousianos and K. Chakrabarty, "Testing for socs with advanced static and dynamic power-management capabilities," in *Proc. ACM/IEEE Des., Autom. & Test in Europe (DATE) Conf.*, March 2013, pp. 737–742.

[9] S. Khursheed, K. Shi, B. Al-Hashimi, P. Wilson, and K. Chakrabarty, "Delay test for diagnosis of power switches," *IEEE Trans. Very Large Scale Integr. Systems*, vol. 22, no. 2, pp. 197–206, Feb 2014.

[10] S. Khursheed, S. Yang, B. Al-Hashimi, X. Huang, and D. Flynn, "Improved dft for testing power switches," in *Proc. IEEE Eur. Test Symp.*, May 2011, pp. 7–12.

[11] C. Long and L. He, "Distributed sleep transistors network for power reduction," in *Proc. Design Automation Conf.*, June 2003, pp. 181–186.

[12] E. J. McCluskey, *Logic design principles - with emphasis on testable semicustom circuits*. Prentice Hall, 1986.

[13] S.-P. Mu, Y.-M. Wang, H.-Y. Yang, M.-T. Chao, S.-H. Chen, C.-M. Tseng, and T.-Y. Tsai, "Testing methods for detecting stuck-open power switches in coarse-grain mtcmos designs," in *Proc. Intern. Conf. on Comp-Aid. Des.*, Nov 2010, pp. 155–161.

[14] K. Roy, S. Mukhopadhyay, and H. Mahmoodi-Meimand, "Leakage current mechanisms and leakage reduction techniques in deep-submicrometer cmos circuits," *Proceedings of the IEEE*, vol. 91, no. 2, pp. 305–327, Feb 2003.

[15] R. Swanson, A. Wong, S. Ethirajan, and A. Majumdar, "Avoiding burnt probe tips: Practical solutions for testing internally regulated power supplies," in *Proc. Eur. Test Symp.*, May 2014, pp. 1–6.

[16] R. Wang, Z. Zhang, X. Kavousianos, Y. Tsiatouhas, and K. Chakrabarty, "Built-in self-test, diagnosis, and repair of multimode power switches," *IEEE Trans. on CAD*, vol. 33, no. 8, pp. 1231–1244, Aug 2014.

[17] C.-J. Wei, H. Chen, and S.-J. Chen, "Design and implementation of block-based partitioning for parallel flip-chip power-grid analysis," *IEEE Trans. on CAD*, vol. 31, no. 3, pp. 370–379, March 2012.

[18] S. Zhong, S. Khursheed, and B. Al-Hashimi, "A fast and accurate process variation-aware modeling technique for resistive bridge defects," *IEEE Trans. on CAD.*, vol. 30, no. 11, pp. 1719–1730, Nov 2011.