



Evaluation of the Impact of Technology Scaling on Delay Testing for Low-Cost AVS

Mahroo Zandrahimi¹ · Philippe Debaud² · Armand Castillejo² · Zaid Al-Ars¹

Received: 3 October 2018 / Accepted: 21 April 2019 / Published online: 20 May 2019
© The Author(s) 2019

Abstract

With the continued down-scaling of IC technology and increase in manufacturing process variations, it is becoming ever more difficult to accurately estimate circuit performance of manufactured devices. This poses significant challenges on the effective application of adaptive voltage scaling (AVS) which is widely used as the most important power optimization method in modern devices. Process variations specifically limit the capabilities of Process Monitoring Boxes (PMBs), which represent the current industrial state-of-the-art AVS approach. To overcome this limitation, in this paper we propose an alternative solution using delay testing, which is able to eliminate the need for PMBs, while improving the accuracy of voltage estimation. The paper shows, using simulation of ISCAS'99 benchmarks with 28nm FD-SOI library, that using delay test patterns result in an error of 5.33% for transition fault testing (TF), error of 3.96% for small delay defect testing (SDD), and an error as low as 1.85% using path delay testing (PDLY). In addition, the paper also shows the impact of technology scaling on the accuracy of delay testing for performance estimation during production. The results show that the 65nm technology node exhibits the same trends identified for the 28nm technology node, namely that PDLY is the most accurate, while, TF is the least accurate performance estimator.

Keywords Adaptive voltage scaling · Performance monitor boxes · Delay testing · Process variations · Power optimization

1 Introduction

Power is one of the primary design constraints and performance limiters in the semiconductor industry. Reducing power consumption can extend battery life-time of portable

systems, decrease cooling costs, as well as increase system reliability [19]. Various low power approaches have been implemented in the IC manufacturing industry, among which adaptive voltage scaling (AVS) has proven to be a highly effective method of achieving low power consumption while meeting the performance requirements. Moreover, with the on going scaling of CMOS technologies, variations in process, supply voltage, and temperature (PVT) have become a serious concern in integrated circuit design. Due to die to die process variations, each chip has its own characteristics which lead to different speed and power consumption. The basic idea of AVS is to adapt the supply voltage of each manufactured chip to the optimal value based on the operation conditions of the system so that in addition to saving power; variations are compensated as well, while maintaining the desired performance.

A standard industrial approach for AVS is the use of on-chip PMBs to be able to estimate circuit performance during production. AVS approaches embed several PMBs in the chip architecture so that based on the frequency responses of these monitors during production, the chip performance is estimated and the optimal voltage is adapted exclusively to each operating point of each manufactured chip. PMBs

Responsible Editor: A. D. Singh

✉ Mahroo Zandrahimi
m.zandrahimi@tudelft.nl

Philippe Debaud
philippe.debaud@st.com

Armand Castillejo
armand.castillejo@st.com

Zaid Al-Ars
z.al-ars@tudelft.nl

¹ Delft University of Technology, Mekelweg 4, 2628CD, Delft, Netherlands

² STMicroelectronics, 12 Rue Jules Horowitz, 38019 Grenoble, France

range from simple inverter based ring oscillators to more complex critical path replicas designed based on the most used cells extracted from the potential critical paths of the design [3–5, 7, 9, 12]. The frequency of PMBs is dependent on various silicon parameters such as NMOS and PMOS speeds, capacitances, leakage, etc.

To be able to estimate the circuit performance based on PMB responses during production, the correlation between frequency of PMBs and circuit frequency should be measured during characterization, an earlier stage of manufacturing. Once PMB responses are correlated to application performance, they are ready to be used for AVS during production. Figure 1 shows the way PMBs can be used for the application of AVS power optimization. The goal is to have the appropriate voltage supply point optimized for each silicon die individually. During production and based on the frequency responses from PMBs, the chip performance will be estimated to enable AVS. This can be used to serve various purposes. First, AVS is used to adapt the voltage in order to compensate for PVT variations. AVS is also used to enhance yield; operating voltage of fast chips is reduced to compensate for extra leakage power, while operating voltage of slow chips is increased to reach the performance target. In addition, AVS can be used to improve power efficiency per die by reducing the voltage supply to the optimum voltage at the transistor level [19].

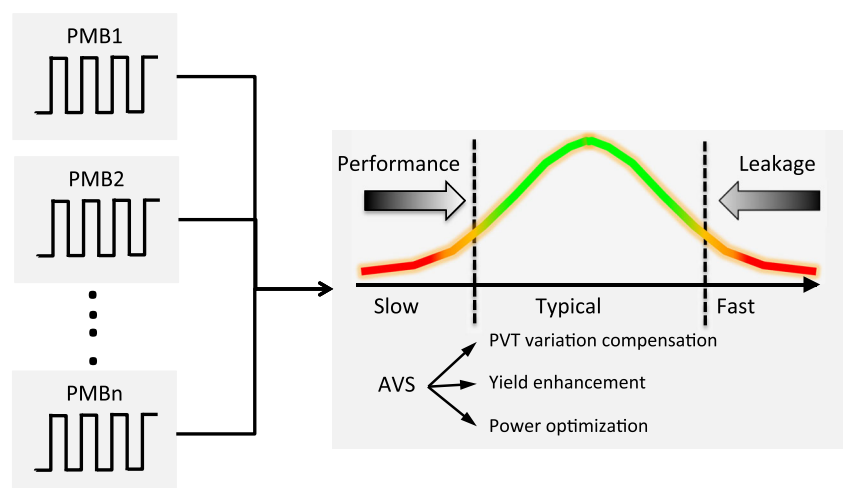
However, trying to predict performance of the many millions of paths in a given design based on information from a single unique path could be difficult and in many cases inaccurate. This results in high costs, extra margins, and consequently yield loss and performance limitations. This approach might work for very robust technologies and when only very few parameters influence performance, such as voltage, process corner, and temperature. However, in deep sub-micron technologies, as intra-die variation and

interconnect capacitances are becoming predominant, it is more complex to estimate the performance of the whole design based on few PMBs. Hence, to improve the accuracy, we should use an alternative approach that increases the number of paths we take into account for performance estimation. Moreover, the more the characterization effort can be reduced, the more cost effective the AVS approach will be.

Previous work in this context, such as [15] and [6], propose techniques for generating optimal set of delay test patterns during the characterization process. These techniques guarantee to invoke the worst-case delays of the circuit. These tests are applied on a small set of chips selected from a batch of first silicon. The reason is to expose systematic timing errors that are likely to affect a large fraction of manufactured chips. Hence, these timing errors may be addressed via redesign before the design moves into high-volume manufacturing. However, they do not propose test generation for the purpose of application to AVS during manufacturing on every chip. Work published in [2] and [11] proposes using a predictive subset testing method which reduces the number of paths that need to be tested. This method is able to find correlations that exist between performance of different paths in the circuit. This way it is possible to predict the performance of untested paths within the desired quality level, thus, improve test complexity and cost. However, due to the increasing effect of intra die process variations in smaller technologies, the correlations between different paths change throughout a single chip rendering this technique ineffective in current manufacturing technologies.

Authors of [13] propose an efficient technique for post manufacturing test set generation by determining only 10% representative paths and estimating the delays of other paths by statistical delay prediction. This technique achieves 94% reduction in frequency stepping iterations during delay

Fig. 1 Implementation of AVS power optimization using PMBs



testing with a slight yield loss. However, the authors are only able to define static power specification for all manufactured chips, which is not able to address AVS utilization for each chip. Shim and Hu [16] introduces a built-in delay testing scheme for online AVS during run time, which offers a good solution for mission critical applications. However, this requires significant software modifications, making it very expensive for non critical applications. Zain Ali [18] investigates the importance of delay testing using all voltage/frequency settings of chips equipped with AVS to guarantee fault-free operation. However, their approach does not enable setting optimal voltage and corresponding frequencies to enable AVS.

In this paper, we introduce a cost effective approach for the estimation of AVS voltages during production using delay test patterns. The contributions of this paper are the following:

- Proposing the new concept of using delay testing for AVS during production.
- A detailed investigation of the delay testing approach including TF, PDLY, and SDD in terms of accuracy and effectiveness using 29 ISCAS'99 benchmarks with 28nm FD-SOI library for 42 different process corners.
- A study on the impact of technology scaling on accuracy and effectiveness of the delay testing approach using 65nm, 40nm, and 28nm FD-SOI libraries.

The rest of this paper is organized as follows. Section 2 explains the implementation of AVS in different levels of the design and manufacturing process. Limitations of PMB-based AVS are introduced in Section 3. Section 4 proposes the new approach of using delay test patterns for AVS. Evaluation of the proposed approach is presented in Section 5 using simulation results on ISCAS'99 benchmarks. Section 6 investigates the impact of technology scaling on accuracy and effectiveness of our proposed method for AVS. Section 7 concludes the paper and proposes potential solutions for future work.

2 Background

AVS can be done either offline during production or online during run-time. Offline AVS approaches estimate optimal voltages for each target frequency during production, while online AVS approaches measure optimal voltages during run-time by monitoring the actual circuit performance.

With regard to accuracy and tuning effort, online AVS approaches are very accurate and no tuning effort is needed, since they monitor the actual critical path of the circuit, and there is no need to add safety margins on top of the measured parameters due to inaccuracies. However, for offline AVS approaches, since there is no interaction

between PMBs and the circuit, the correlation between PMB responses and the actual performance of the circuit is estimated during the characterization phase using the amount of test chips representative of the process window. Since there are discrepancies in the responses of same PMBs from different test chips, the estimated correlation between the frequency of PMBs and the actual performance of the circuit could be very pessimistic, which results in wasting power and performance. Hence in terms of accuracy and tuning effort, online approaches always win [20].

In terms of planning effort and implementation risk, online AVS approaches are considered very risky and intrusive since adding flip-flops at the end of critical paths requires extensive modification in hardware and thus incurs a high cost. Moreover, for some sensitive parts of the design, such as CPU and GPU, which should operate at high frequencies, implementing direct measurement approaches is quite risky since it affects planning, routing, timing convergence, area, and time to market. On the other hand, offline AVS approaches are considered more acceptable in terms of planning and implementation risk, since there is no interaction between PMBs and the circuit, hence, PMBs can even be placed outside the macros being monitored, but not too far due to within die variations. Consequently, offline AVS approaches seem more manageable due to the fact that they can even be considered as an incremental solution for existing devices and the amount of hardware modification imposed to the design is very low. Consequently, according to the application, one can decide which technique more suits a design. For example, for medical applications accuracy and power efficiency are far more important than the amount of hardware modification and planing effort, while, for nomadic applications, such as mobile phones, tablets, and gaming consoles, cost and the amount of hardware modification are considered the most significant.

In this work our focus is on AVS implementation on devices used for nomadic applications. Thus, Our focus is on offline AVS approaches. Offline AVS techniques which are currently being used for nomadic applications in industry use PMBs to estimate performance of each manufactured chip during production to find the optimal voltage for each frequency target accordingly. It is worth mentioning that the use of PMBs is due to the fact that AVS for each chip during production should be done as fast as possible, thus, running functional tests on CPU to measure optimal voltages for each operating point is not feasible. In this section, we explain the implementation of offline AVS in the different stages of the design and manufacturing process. Figure 2 presents the stages along with a discussion.

- **Design:** The process starts with the design stage, where the circuit structure and functionality is described based

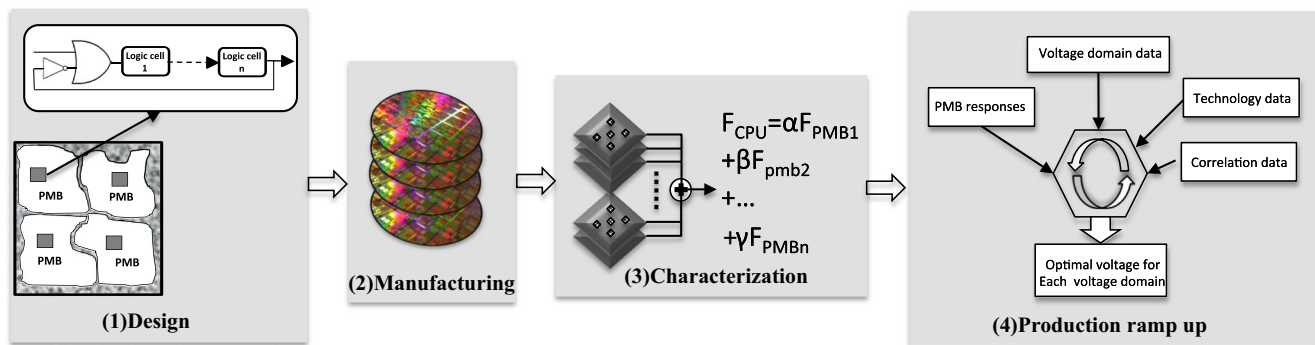


Fig. 2 AVS implementation in different levels of the design and manufacturing process

on a given set of specifications. When the design is completed, various PMBs are embedded in the chip structure. Ring oscillators are the most widely used type of PMBs present today in many products, the frequency of which is dependent on various silicon parameters such as NMOS and PMOS speeds, capacitances, leakage, etc. These ring-oscillator based PMBs are constructed using standard logic components and placed in various locations on the chip to capture all kind of variations (see Fig. 2(1)). Due to intra-die variations, it is more efficient to place various PMBs close or inside the block which is being monitored so that all types of process variations are captured and taken into account for performance estimation. The number of used PMBs depends on the size of the chip. There is no interaction between the PMBs and the circuit.

- **Manufacturing:** When the design stage is completed, the manufacturing stage starts where a representative number of chip samples will be manufactured. The number of chip samples should be representative of the process window to make sure that all kind of process variations are taken into account for the correlation process.
- **Characterization:** To be able to use PMBs for AVS during production, the correlation between PMBs frequency and the actual application behavior is measured during characterization stage. The chip samples are used to find this correlation. The following steps are done for each operating point of each chip sample. 1. The optimal voltage is measured using functional test patterns. 2. The chip is set to the optimal voltage and the frequency of each PMB is captured. 3. The correlation between PMB frequencies and the actual frequency of the chip is calculated. Therefore, based on the data from all chip samples, we find correlation between PMB frequencies and the actual frequency of CPU for the design taking into account all process corners of the technology (see Fig. 2(3)).

- **Production ramp up:** Once PMBs are tuned to the design during the characterization stage, they are ready to be used for voltage estimation during the production ramp up stage. During production and based on the frequency responses from PMBs, the circuit frequency is estimated so that optimal voltage can be predicted exclusively for each operating point of each manufactured chip. Then, margins for voltage and temperature variations as well as aging are added on top of the optimal voltage to make sure that the chip functions properly in different environmental conditions. Finally, optimal voltages for each operating point are either fused in fuse boxes of the chip or stored in a non volatile memory of the chip and are ready to be used for AVS during run-time.

3 Motivation

Although PMB-based AVS is very fast during production, as technology scaling enters the nanometer regime, this technique is showing limitations regarding time to market, cost, and effectiveness in power saving. These limitations are discussed below:

- **Long characterization:** The correlation process (i.e., finding the correlation between PMB responses and the actual frequency of the circuit) should be done for an amount of test chips representative of the process window to make sure (for all manufactured chips) voltage estimation based on PMB responses is correlated with application behavior. This correlation process has a negative impact in terms of design effort and time to market, which makes these approaches very expensive. Our delay test based approach, while does not eliminate the need for characterization, it does reduce the time needed to perform it.
- **Incomplete functional patterns:** finding a complete set of functional patterns that reflects the real system

performance could be very tricky specially for complex systems. Also, we note that identifying the most critical part of the application is not possible in most cases. Although our delay test based approach also does not provide complete coverage, any set of delay test patterns (even very small sets) have an advantage as compared to PMBs. The reason is that PMBs only consider one or few paths, while delay testing considers undoubtedly more paths for voltage estimation.

- **Not a solution for general logic:** the fact that functional patterns are used for the correlation process makes PMB approaches not suitable for general logic, since even though using functional patterns for programmable parts of the design such as CPU and GPU is possible, the rest of the design such as interconnects are difficult to be characterized using this approach [1].
- **Not effective enough:** since there are discrepancies in the responses of same PMBs from different test chips, the estimated correlation between the frequency of PMBs and the actual performance of the circuit could be very pessimistic, which results in wasting power and performance. In [21], a silicon measurement on 625 devices manufactured using 28nm FD-SOI technology had been done. 12 PMBs are embedded in each device. Results show that optimum voltage estimation based on PMBs lead to nearly 10% of wasted power on average and 7.6% in the best case, when a single PMB is used for performance estimation.

4 Application of Delay Testing for AVS

4.1 Types of Delay Testing

In this paper, we propose an innovative new approach for AVS using delay testing during production. Since delay testing is closely related to the actual functionality of the circuit being tested, and since it covers many path-segments of the circuit design, it can be a much better performance representative than a PMB. Such a test-based approach has a number of unique advantages as compared to PMB-based approaches.

1. First, this approach can be performed *at a lower cost* than PMB approaches, since delay tests are routinely performed during production to test for chip functionality.
2. In addition, since delay testing is performed to explicitly test for actual chip performance, the expensive phase of correlating PMB responses to chip performance is not needed anymore, which reduces the length

of the characterization stage (see Fig. 2(3)), and subsequently dramatically reduces cost and time to market.

3. Moreover, as functional patterns are not used anymore, the delay testing approach could be a solution for general logic, and not only for CPU and GPU components.
4. And last but not least, this approach makes using PMBs redundant, which saves silicon area as well as PMB design time.

TF test patterns target all gates and indirectly cover all path-segments. Hence, it covers all different kinds of gates and interconnect structures. Since several faults can be tested in parallel, we can achieve a high coverage with few patterns [22]. However, automatic test pattern generation (ATPG) algorithms are based on heuristics like SCOAP [8], which tend to minimize computational effort. Thus, when several solutions are available for path sensitization, ATPG will use the easiest, which means that the algorithm tends to target the shorter paths rather than the optimal critical paths of the design [10]. On the other hand, we can alternatively use SDD testing, which sensitizes paths with smallest slacks, as well as PDLY testing, which sensitizes a number of selected most critical paths. Among the three delay testing methods, PDLY has the highest delay test accuracy since it sensitizes functional, long paths, which is an advantage over TF and SDD testing. However, in PDLY testing the objective is to obtain a transition along those critical paths which are on average longer and more complex than the paths targeted in TFs, thus reducing parallel testing capability and thereby reduces the overall coverage achieved.

In this paper, we propose using three different types of delay testing to identify optimal AVS voltages: transition fault testing, small delay defects and path delay testing [17]. As shown in Fig. 3, these three types of testing represent a tradeoff between test accuracy and test coverage, with TF having the highest coverage and lowest accuracy for a given test cost, and PDLY having the lowest coverage and highest accuracy. Despite the fact that these delay testing methods have their limitations as technology scales down, they can

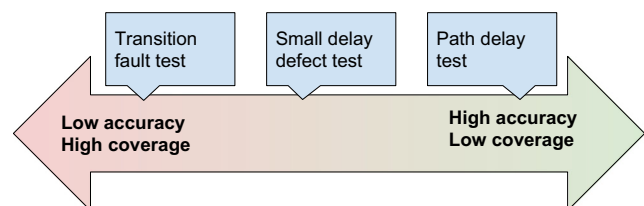


Fig. 3 Tradeoff in accuracy and coverage between different types of delay testing types

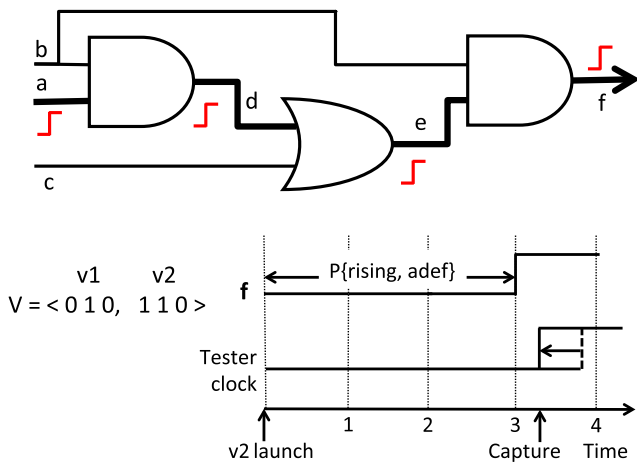


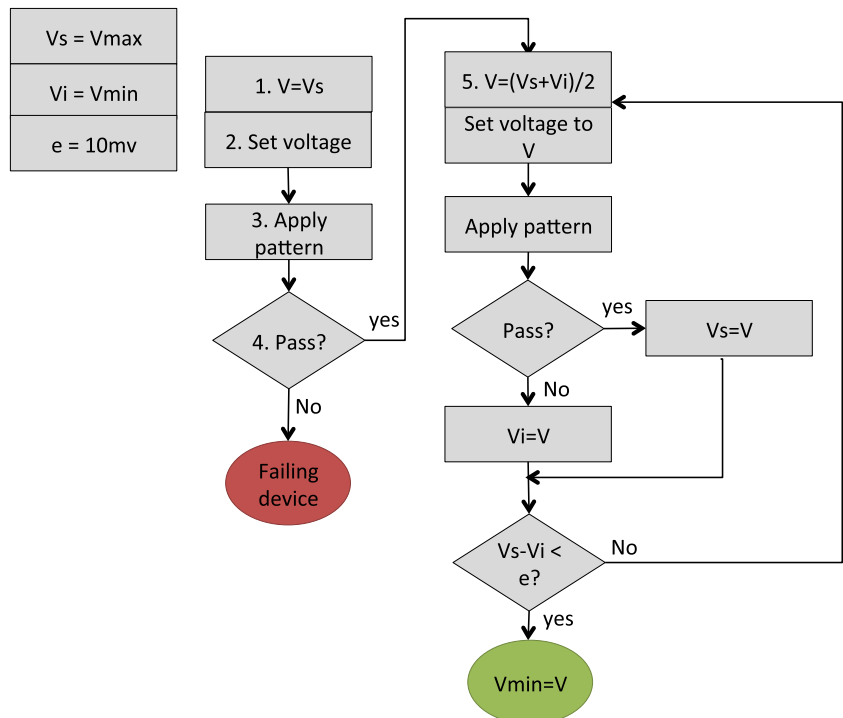
Fig. 4 An example of performance prediction using path delay testing

be used as better representatives than PMBs for on-chip performance prediction.

4.2 Performance Prediction Using Delay Testing

In order to show the basic idea of how circuit performance can be predicted using delay testing, we show a simple example for performance prediction using path delay testing. Figure 4 shows how performance of a circuit is predicted using path delay test patterns. Assume that the path $P\{\text{rising, adef}\}$ in this figure (the highlighted path) is one of the critical paths of the circuit reported by STA.

Fig. 5 Proposed flow to identify AVS voltages using delay testing



The path delay test pattern needed to propagate the rising transition from input a to output f is the vector pair $V = \langle 010, 110 \rangle$. The values for off-input signals (b and c) are 11 and 00. First vector $v1 = 010$ is applied and given some time for signal values to settle. Vector $v2 = 110$ launches the test, and after a delay time dictated by the critical path the output f will exhibit a rising edge. The timing diagram in the figure shows that the critical path delay is 3 time units, corresponding to a delay unit for each gate along the critical path. It is possible to use this information to identify the maximum frequency of the circuit by using a tester clock to capture the correct value of $f = 1$. Any tester clock period larger than 3 time units will be able to capture the correct value of f. By gradually decreasing the tester clock period, we can have an accurate estimation of the delay of the critical path which can be used to calculate the frequency. The accuracy of performance prediction can be increased by taking more critical paths and corresponding path delay test patterns into account. Therefore, depending on the time invested in testing, the accuracy of performance prediction using delay test patterns can be improved.

4.3 AVS Identification Method

Figure 5 proposes a flow to identify AVS voltages using delay test patterns that could be used during production. The proposed flow performs a binary search to identify the minimum voltage (V_{min}), at which the chip can pass all delay test patterns. The following steps are performed for each operation point of the chip:

1. Apply chip setup at nominal values and initialize variables. V_{min} and V_{max} are defined based on the user specifications. Chips which operate at voltages lower than V_{min} are considered too leaky, and will be discarded since they do not meet power specifications. Chips which can only operate at voltages higher than V_{max} are considered too slow, and will be discarded since they do not meet performance specifications.
2. Set supply voltage to V_{max} and wait for stabilization. According to the performance specifications, V_{max} is the maximum voltage at which a chip must be able to operate properly.
3. Apply at speed test using all the delay test patterns in the pattern set, generated using automatic test pattern generation.
4. If the chip fails the test, discard it: any chip which is not able to operate at this voltage will be discarded since it is considered too slow.
5. Otherwise, compute new values and do a binary search to find V_{min} . This voltage is considered the optimal voltage at which the chip can pass all delay test patterns in the specified pattern set.

Conversion from V_{min} to F_{max} might be required depending on either performance estimation is done for yield enhancement or power optimization. “ e ” is an arbitrary value to be set by the users to define the resolution they want.

The basic requirement of using delay testing for AVS is that there should be a reasonable correlation between delay testing frequency the chip can attain while passing all delay test patterns and the actual frequency of the chip. In this case, delay test frequency could be a representative of actual chip performance. Previous research indicated that such a correlation does exist for specific designs [14]. It is important to note that since performance estimation during production should be done as fast as possible, running functional patterns on CPU is therefore most of the time not feasible. We do emphasize, however, that this is only true during production testing. Functional tests are important to validate design behavior in earlier stages of manufacturing.

In order to investigate if such correlation exists for a wider set of designs, we have performed detailed simulations on ISCAS’99 benchmarks, which contain 29 designs with different characteristics.

5 Evaluation Results

5.1 Simulation Setup

This subsection explains the flow we used to explore if delay test frequency correlates with the actual frequency

of the circuits. We use 28nm FD-SOI (<http://www.st.com/content/st.com/en/about/innovation---technology/FD-SOI.html>) libraries to compare the delay fault maximum frequency versus the critical paths of ISCAS’99 benchmarks (<http://www.cad.polito.it/downloads/tools/itc99.html>) using SYNOPSIS tools (<http://www.synopsys.com/tools/pages/default.aspx>). ISCAS’99 contains 29 designs from small circuits like b02 with 22 cells to more complicated designs like b19 with almost 75K cells. The detailed information on ISCAS benchmarks is presented in Table 1 synthesized using 28nm FD-SOI library at SS corner, 0.9V voltage, and 40 °C temperature. 42 different corners of 28nm FD-SOI library have been used with different characteristics in terms of voltage, body biasing, temperature, transistor speed and aging parameters. We used Design Compiler in topographical mode for physical synthesis, Primitime for static timing analysis (STA), Tetramax for automatic test pattern generation (ATPG), and VCS for back annotated simulation. Since functional patterns are not available for ISCAS’99 benchmarks, we use STA instead as a reference for comparison versus delay test frequencies. This choice can be justified by noting that any set of functional patterns cannot be complete, since it is very tricky to select an application which reflects the real system performance specially for complex systems. Here, we note that identifying the

Table 1 Physical data of ISCAS’99 benchmarks synthesized using 28nm FDSOI library at SS corner

Benchmark	Frequency	Total area (um2)	# combin. cells	# sequential cells	# ports
b01	5Ghz	35,90	35	5	9
b02	5Ghz	24,04	22	4	7
b03	2.5Ghz	149,16	66	30	12
b04	5Ghz	891,18	532	109	23
b05	5Ghz	738,53	647	53	42
b06	3.33Ghz	41,45	29	9	12
b07	1.66Ghz	274,39	258	51	13
b08	5Ghz	293,00	195	41	18
b09	5Ghz	179,08	89	28	7
b10	2.5Ghz	114,57	98	20	21
b11	2Ghz	327,71	388	31	17
b12	3.33Ghz	1016,95	785	121	15
b13	3.33Ghz	269,17	208	53	24
b14	909Mhz	3410,12	3897	461	90
b14_1	909Mhz	3025,73	3268	461	90
b15	5Ghz	6459,67	6859	484	110
b15_1	5Ghz	6569,13	6845	484	110
b17	1.5Ghz	13051,00	14750	1520	472
b17_1	1.5Ghz	13066,12	15011	1520	472
b18	909Mhz	33719,30	39363	3964	1188
b18_1	909Mhz	33241,66	38482	3964	1188
b19	909Mhz	66037,68	75934	7929	2456
b19_1	909Mhz	65535,79	74538	7929	2456
b20	909Mhz	7141,85	8446	922	239
b20_1	909Mhz	6458,59	7343	922	239
b21	909Mhz	7197,45	8545	922	239
b21_1	909Mhz	6258,94	7494	922	239
b22	909Mhz	10626,28	12975	1383	329
b22_1	909Mhz	9651,76	11308	1383	329

most critical part of the application is not possible in most cases. We also note that although gate-level simulations provide pessimistic STA delay estimations due to the low level of details for resistance and capacitance values, this pessimistic estimation is also true for the delay test patterns we simulated in our experiments, since all simulations were performed at the gate level.

Figure 6 shows the simulation flow containing 4 steps as follows:

- **Synthesis:** physical synthesis on 29 ISCAS'99 circuits using 28 nm FDSOI physical library to extract the netlists, and other reports required as an input for STA, ATPG and back annotated simulation. (29 netlists and other reports)
- **STA:** timing analysis using 42 corners of 28nm FDSOI library to extract the critical timing of benchmarks in each corner. (42 corners*29 netlists= 1218 critical timing reports)
- **ATPG:** TF, SDD and PDLY test pattern generation to extract test patterns and test benches for each benchmark. We generated 4 TF pattern sets consisting of 50, 100, 200, and 500 patterns, 3 PDLY fault pattern sets consisting of 100, 1000, and 10000 patterns, and 2 SDD pattern sets consisting of 50 and 500 patterns (targeting only register to register paths) for each benchmark. Figure 7 shows some detailed information regarding the number of test patterns that ATPG could generate for each pattern set for each benchmark. For instance, for small benchmarks such as b01 with only 30 cells, increasing pattern count does not have any effect on coverage since the total number of TF patterns is less than 50.
- **Simulation:** applying delay test patterns on back annotated simulation of each benchmark, and searching for maximum frequency at which each device passes the test. Frequency search is done using binary search and STA results as a starting point since the maximum frequency cannot exceed critical timing.

Finally, we compared STA results versus delay fault frequencies of 29 ISCAS'99 circuits in 42 corners.

Fig. 6 Simulation flow for comparing delay testing frequency vs. STA for the 29 ISCAS'99 circuits

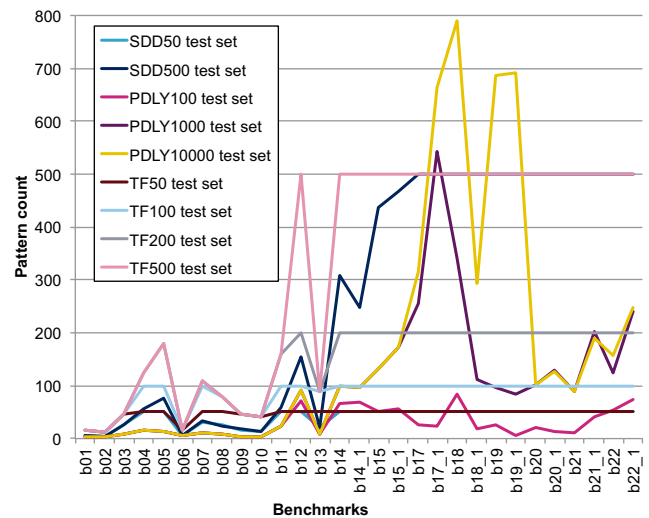
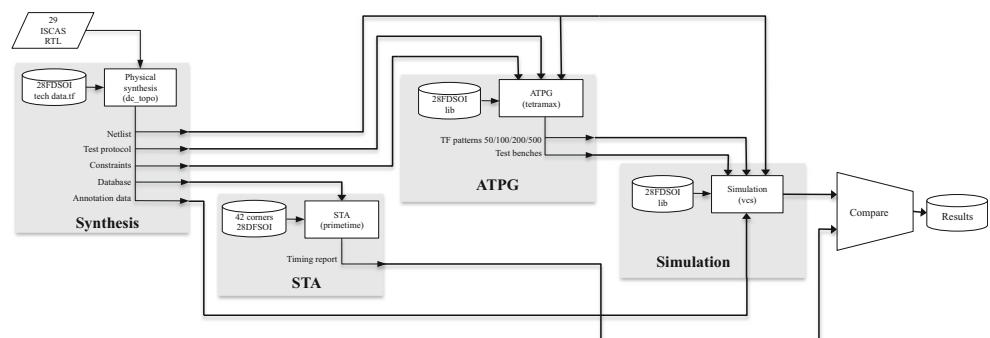


Fig. 7 Number of test patterns generated for each ISCAS'99 design targeting TFs, SDDs and PDLYs

Furthermore, to understand how untestable paths are influencing the results, we have done the following post processing analysis for each circuit: We first extracted the 10K most critical paths and generated a pattern covering that path with the highest effort level. Considering all untestable paths as false paths, we removed all those paths from STA, and updated the comparison of delay fault frequencies versus STA accordingly. The results are presented in the next subsection.

5.2 Simulation Results

To understand if delay testing is a reasonable performance indicator that can be used for AVS during production, we compared the maximum frequency at which each delay pattern set can be performed for each benchmark versus STA results. We estimated the performance of each benchmark in each of 42 corners both using STA and each delay pattern set. In order to present the results, we define a parameter named *error* which is measured for each

Table 2 Error and standard deviation (SD) of error for TF versus STA (darker shaded entries indicate higher values)

Benchmark	TF50		TF100		TF200		TF500	
	error	SD	error	SD	error	SD	error	SD
b01	1.00%	2.54%	1.00%	2.54%	1.00%	2.54%	1.00%	2.54%
b02	3.81%	3.15%	3.81%	3.15%	3.81%	3.15%	3.81%	3.15%
b03	4.04%	1.96%	4.04%	1.96%	4.04%	1.96%	4.04%	1.96%
b04	2.97%	2.74%	2.57%	2.90%	1.70%	3.21%	1.70%	3.21%
b05	1.28%	2.42%	1.28%	2.42%	1.21%	2.38%	1.21%	2.38%
b06	3.64%	1.84%	3.64%	1.84%	3.64%	1.84%	3.64%	1.84%
b07	5.83%	2.25%	2.20%	1.09%	2.20%	1.09%	2.20%	1.09%
b08	2.84%	3.21%	2.00%	3.45%	2.00%	3.45%	2.00%	3.45%
b09	7.50%	1.88%	7.50%	1.88%	7.50%	1.88%	7.50%	1.88%
b10	0.05%	0.93%	0.05%	0.93%	0.05%	0.93%	0.05%	0.93%
b11	2.19%	1.84%	0.46%	1.35%	0.20%	1.07%	0.20%	1.07%
b12	1.82%	3.27%	1.82%	3.27%	1.82%	3.27%	1.67%	3.28%
b13	2.35%	1.87%	2.35%	1.87%	2.35%	1.87%	2.35%	1.87%
b14	18.55%	1.44%	18.52%	1.43%	18.52%	1.43%	11.29%	1.53%
b14_1	19.23%	4.06%	14.01%	0.97%	14.01%	0.97%	13.66%	0.88%
b15	2.80%	2.05%	2.75%	2.01%	2.46%	1.70%	2.06%	1.34%
b15_1	7.44%	2.28%	7.38%	3.71%	3.21%	1.50%	2.57%	1.32%
b17	4.24%	3.14%	4.21%	3.08%	3.71%	2.52%	3.68%	2.48%
b17_1	8.29%	3.09%	5.26%	1.04%	4.91%	1.01%	4.91%	1.01%
b18	15.64%	0.90%	12.25%	2.02%	10.54%	1.13%	6.47%	1.40%
b18_1	14.53%	3.67%	7.89%	1.90%	7.57%	2.56%	7.47%	2.59%
b19	17.80%	1.13%	15.90%	2.07%	15.98%	1.89%	12.42%	2.18%
b19_1	8.83%	1.10%	8.82%	1.10%	8.82%	1.10%	8.82%	1.10%
b20	13.23%	0.44%	12.53%	0.47%	12.29%	0.73%	10.00%	1.33%
b20_1	15.99%	0.77%	15.70%	0.68%	11.48%	0.42%	9.94%	1.44%
b21	12.82%	0.43%	12.82%	0.43%	7.62%	0.42%	7.62%	0.42%
b21_1	4.96%	1.02%	4.47%	0.95%	4.45%	0.90%	3.42%	0.69%
b22	11.22%	1.83%	10.38%	1.40%	10.27%	1.22%	10.27%	1.22%
b22_1	12.05%	3.79%	12.01%	3.81%	11.94%	3.78%	8.54%	2.65%
Average	7.83%	2.11%	6.81%	1.92%	6.18%	1.79%	5.33%	1.80%

benchmark. The concept relates to how much margin should be taken into account due to inaccuracies as a result of performance estimation using delay testing. In addition to this parameter, we also introduce a parameter as SD_{error} for each benchmark which is used to measure the confidence in the estimated $error$. To be able to measure $error$ for each benchmark, first we measured performance error for each corner by:

$$error_{corner} = (P_{STA} - P_{DT}) / P_{STA} \tag{1}$$

where P_{STA} is the performance estimation using STA, and P_{DT} is the performance estimation using delay testing for the corresponding corner. Once $error_{corner}$ is calculated for all process corners, $error$ can be obtained for each benchmark by:

$$error = \max_{all\ corners} [error_{corner}] \tag{2}$$

Then, SD_{error} is calculated for each benchmark using the following equation:

$$SD_{error} = \sqrt{\frac{\sum_{all\ corners} [error_{corner} - \overline{error}]^2}{42}} \tag{3}$$

where $error_{corner}$ is the performance error for each corner, and \overline{error} is the mean of $error_{corner}$ for all 42 different corners.

Tables 2, 3 and 4 present the $error$ and SD_{error} , for all ISCAS'99 benchmarks for TF, SDD and PDLY, respectively. We generated the results for 4 TF pattern sets including 50, 100, 200, and 500 patterns, 2 SDD pattern sets including 50 and 500, and 3 PDLY pattern sets including 100, 1000, and 10000.

As it can be seen in these tables, depending on the size of each benchmark, and with increasing pattern count, the $error$ is reduced. For TF, for example, the reduction in $error$ is higher than 5% for 7 benchmarks (b14, b14.1, b18, b18.1,

Table 3 Error and SD of error for SDD versus STA (darker shaded entries indicate higher values)

Benchmark	SDD50		SDD500		Benchmark	SDD50		SDD500	
	error	SD	error	SD		error	SD	error	SD
b01	0.79%	1.64%	0.79%	1.64%	b15	2.77%	0.93%	1.56%	1.07%
b02	4.33%	2.78%	4.33%	2.78%	b15_1	3.60%	1.60%	1.08%	0.71%
b03	4.12%	1.98%	4.12%	1.98%	b17	3.43%	2.22%	1.32%	1.41%
b04	1.70%	3.21%	1.70%	3.21%	b17_1	4.37%	0.91%	3.18%	0.51%
b05	1.21%	2.38%	1.21%	2.38%	b18	5.00%	0.59%	4.86%	0.66%
b06	4.36%	2.23%	4.36%	2.23%	b18_1	8.13%	1.51%	6.92%	3.85%
b07	5.21%	1.24%	5.21%	1.24%	b19	11.77%	2.59%	11.16%	1.85%
b08	2.84%	3.21%	2.84%	3.21%	b19_1	8.83%	1.10%	8.82%	1.10%
b09	7.42%	1.77%	7.42%	1.77%	b20	8.04%	1.63%	3.33%	0.82%
b10	0.18%	0.87%	0.18%	0.87%	b20_1	9.75%	1.22%	7.24%	1.15%
b11	0.20%	1.07%	0.20%	1.07%	b21	7.03%	0.46%	5.86%	0.53%
b12	1.75%	3.27%	1.67%	3.28%	b21_1	2.47%	1.28%	2.16%	0.87%
b13	2.35%	1.87%	2.35%	1.87%	b22	6.34%	0.33%	5.07%	0.78%
b14	12.16%	1.05%	6.52%	0.85%	b22_1	8.44%	2.35%	5.65%	1.69%
b14_1	10.15%	0.71%	3.77%	0.79%	Average	5.13%	1.66%	3.96%	1.59%

Table 4 Error and SD of error for PDLY versus STA (darker shaded entries indicate higher values)

Benchmark	PDLY100		PDLY1000		PDLY10000	
	error	SD	error	SD	error	SD
b01	0,33%	0,77%	0,33%	0,77%	0,33%	0,77%
b02	0,11%	0,82%	0,11%	0,82%	0,11%	0,82%
b03	4,05%	1,96%	4,05%	1,96%	4,05%	1,96%
b04	1,70%	3,21%	1,70%	3,21%	1,70%	3,21%
b05	1,21%	2,38%	1,21%	2,38%	1,21%	2,38%
b06	3,64%	1,84%	3,64%	1,84%	3,64%	1,84%
b07	2,20%	1,09%	2,20%	1,09%	2,20%	1,09%
b08	1,95%	3,46%	1,95%	3,46%	1,95%	3,46%
b09	7,50%	1,88%	7,50%	1,88%	7,50%	1,88%
b10	0,05%	0,93%	0,05%	0,93%	0,05%	0,93%
b11	0,20%	1,07%	0,20%	1,07%	0,20%	1,07%
b12	1,82%	3,27%	1,82%	3,27%	1,82%	3,27%
b13	2,35%	1,87%	2,35%	1,87%	2,35%	1,87%
b14	16,35%	1,54%	0,23%	0,43%	0,23%	0,43%
b14_1	13,35%	0,73%	0,22%	0,32%	0,22%	0,32%
b15	2,46%	1,70%	0,80%	0,79%	0,80%	0,79%
b15_1	3,25%	1,55%	0,57%	0,76%	0,57%	0,76%
b17	3,69%	2,42%	2,31%	1,43%	1,82%	1,25%
b17_1	4,99%	1,04%	1,89%	0,50%	1,89%	0,50%
b18	10,54%	1,13%	0,14%	0,32%	0,46%	0,63%
b18_1	7,96%	2,81%	4,02%	1,11%	4,03%	1,11%
b19	12,35%	2,07%	5,30%	2,74%	4,58%	3,33%
b19_1	8,82%	1,10%	8,76%	0,99%	8,70%	0,92%
b20	11,69%	0,77%	1,50%	0,51%	1,50%	0,51%
b20_1	12,36%	1,05%	0,43%	0,84%	0,43%	0,84%
b21	8,56%	0,84%	0,50%	0,40%	0,50%	0,40%
b21_1	4,45%	0,90%	0,33%	1,32%	0,33%	1,32%
b22	10,29%	1,24%	0,34%	0,32%	0,17%	0,61%
b22_1	10,90%	3,35%	12,16%	3,37%	0,41%	0,46%
Average	5,83%	1,68%	2,30%	1,40%	1,85%	1,34%

b19, b20_1 and b21), with the largest reduction in error realized for b18 with an error reduction of 9.18% (from 15.64% down to 6.47%). For SDD, the reduction in error is higher than 5% for 2 benchmarks (b14 and b14_1), with the largest reduction in error realized for b14_1 with an error reduction of 6.38% (from 10.15% down to 3.77%). In the same way, for PDLY the reduction in error is higher than 5% for 9 benchmarks (b14, b14_1, b18, b19, b20, b20_1, b21, b22, b22_1), with the largest reduction in error realized for b14 with an error reduction of 16.12% (from 16.35% down to 0.23%). These specific benchmarks particularly benefit from increasing the number of patterns due to the fact that they represent some of the biggest circuits in the ISCAS'99 benchmark. However, it is important to note that b14 and b14_1 are not the biggest circuits among the benchmarks, which means that the design complexity of the circuits plays an important role as well.

Therefore, depending on the time invested in testing during production, the accuracy of performance estimation using delay testing can be improved. As mentioned earlier, for some small benchmarks such as b01 with only 30 cells, the error remains unchanged with increasing number of patterns since there are no more patterns that can be used to increase the coverage.

Considering the average error (listed in the last row of the tables), this figure shows that increasing the pattern count for TF testing from 50 to 500 results in 2.50% error improvement from 7.83% down to 5.33% for ISCAS'99 benchmarks. In the same way, increasing pattern count from

50 to 500 for SDD testing improves the average error by up to 1.17%, from 5.13% down to 3.96%. Increasing PDLY pattern count from 100 to 10000 causes 3.98% improvement (from 5.83% down to 1.85%) for the average error of PDLY testing for performance prediction. According to these results, we can conclude that using TF testing for performance estimation achieves an average inaccuracy as low as 5.33% with a standard deviation of 1.80%, while, using SDD testing results in 3.96% performance estimation error with 1.59% standard deviation. PDLY testing for performance estimation results in the most accurate estimation error of only 1.85% with a standard deviation of 1.34%.

5.3 Discussion and Evaluation

We can use the measured $error$ and SD_{error} to get a good estimation of the amount of performance margin that needs to be added to each benchmark in order to allow for a reliable application of adaptive voltage scaling. This measured $error$ means that in order to make sure the performance estimation using delay testing is accurate enough, a margin should be added on top of the estimated performance, while SD_{error} represents the confidence in the estimated $error$. Therefore, it is desirable to have $error$ and SD_{error} measurements that are as low as possible for each benchmark since such low measurements allow us to have a margin that is as low as possible.

Figure 8 illustrates the average SD_{error} plotted versus the average $error$ measured using each pattern set for all the circuits in the ISCAS'99 benchmark. The size of each plotted measurements circle in the figure reflects the size of the test pattern set. The figure shows that for each type of delay test, the larger the size of the used test pattern set, the more predictable the performance estimation will be. Therefore, depending on the time invested on testing during production, the accuracy of performance estimation using delay testing can be improved. However, also note that for TF testing, moving from 200 to 500 patterns, the average standard deviation remains unchanged, which means that increasing pattern count up to a limit reduces uncertainty, after which the uncertainty remains unchanged even though the error is improved.

The figure also shows that PDLY patterns have the capacity to achieve the lowest error with the lowest uncertainty, followed by SDD patterns and finally TF patterns. At the same time, the figure shows that if a lower number of patterns is used than actually required by the circuit complexity, the accuracy of the estimation can degrade significantly. This can be seen, for example, for the test set PDLY100, which has an accuracy significantly lower than other PDLY test sets with higher number of patterns.

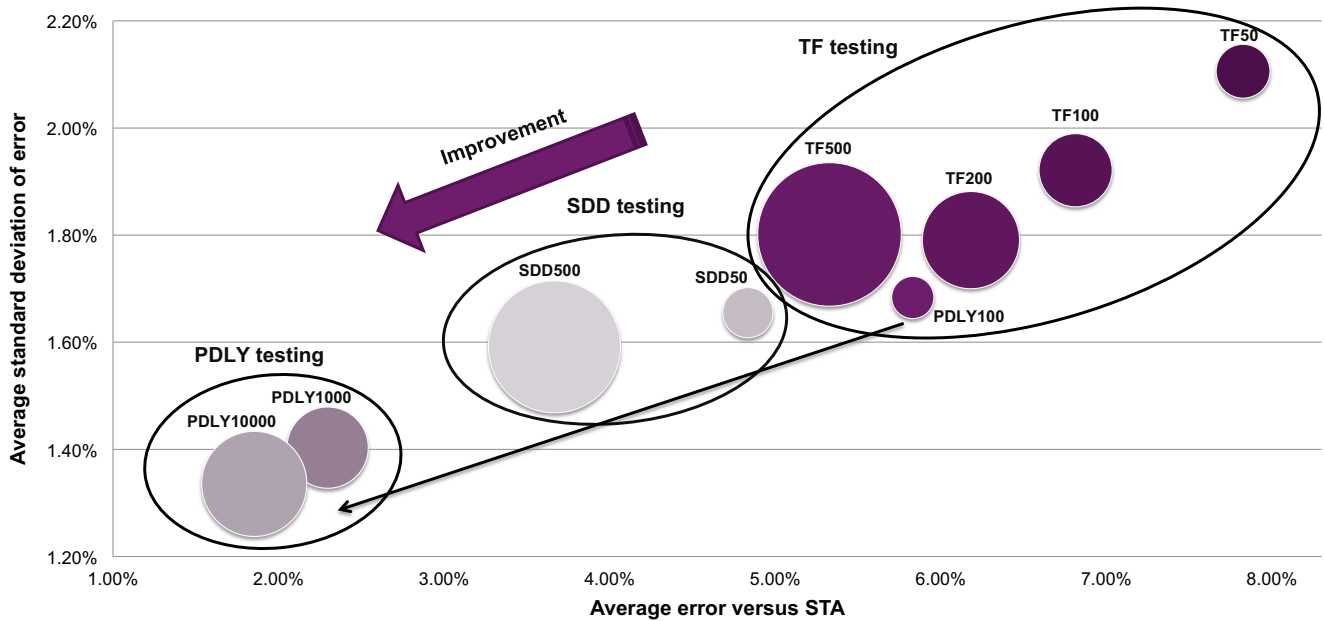


Fig. 8 Average error vs average standard deviation of error for all different test pattern types and test set sizes, in 28nm technology node. TF testing with 50, 100, 200, and 500 pattern sets, SDD testing with

50 and 500 pattern sets, and PDLY testing with 100, 1000, and 10000 pattern sets. The size of the bubble represents the average size of the pattern set used for all benchmarks

6 Impact of Technology Scaling

With the continued reduction in feature sizes and continued scaling of technology nodes, performance estimation becomes increasingly more difficult to achieve using PMBs. In this section, we present an analysis of the impact of technology scaling on the effectiveness of delay testing approaches. For this analysis, we perform elaborate simulations using two technology node libraries: 65nm and 28nm. The simulations are performed for all the circuits in the ISCAS’99 benchmark using all delay test approaches (TF, SDD and PDLY) and with all test set sizes discussed in this paper.

In order to illustrate the impact of technology scaling on the various delay tests in this paper, Fig. 9 plots the average SD_{error} against the average error measured for 65nm and 28nm technology nodes. These measurements are made using each pattern set for all the circuits in the ISCAS’99 benchmarks, and are represented as circles, the size of which reflects the average size of the test pattern set used for all benchmarks. The figure shows that the 65nm technology node exhibits the same trends identified for the 28nm technology node (Fig. 8): for each type of delay test, the larger the size of the used test pattern set, the more predictable the performance estimation will be. Therefore, depending on the time invested in testing during production,

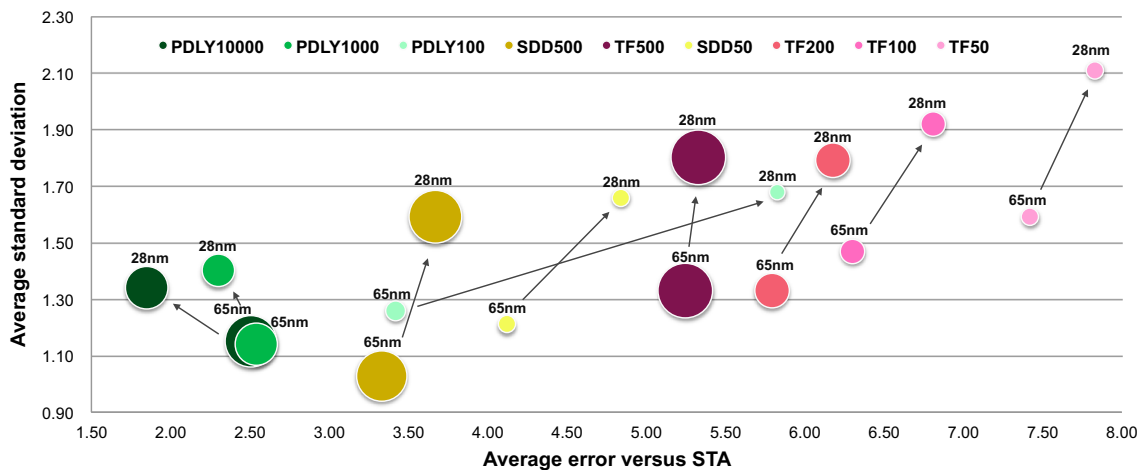


Fig. 9 Impact of technology scaling on average error and standard deviation of different delay test approaches for 65nm and 28nm

the accuracy of performance estimation using delay testing can be improved.

First we consider the impact of migrating to lower technology nodes on the confidence in measured performance. The figure shows that the average standard deviation is always higher for 28nm as compared to 65nm. This means that the smaller the technology node becomes the less confidence there is in the performance measurement made by the test patterns. This is inline with our expectation that more advanced technology nodes add more process variations and increase the uncertainty in measured circuit performance.

In terms of the measured performance error, the results are slightly different. For TF patterns, SDD patterns and very low coverage PDLY100 patterns, the figure shows that for the 28nm node the error is higher than that for 65nm, which is inline with expectation. However, for higher coverage PDLY1000 and PDLY10000, the figure shows that these test patterns are actually able to measure performance with lower error at 28nm as compared to 65nm, which is unique as compared to TF and SDD. This can be attributed to the fact that PDLY measure actual delay of the most critical paths in the circuit, rather than an indicator to this delay. This makes the average performance measurement more accurate and reduces the error. Also note that for the 65nm node, PDLY10000 does not have any accuracy advantage as compared to PDLY1000. This indicates lower variation in the 65nm node that does not require a high number of test patterns to capture.

7 Conclusion

Process variations occurring in deep sub-micron technologies limit PMB effectiveness in silicon performance prediction leading to unnecessary power and yield loss. Estimation of overall application performance from one or few oscillating paths is becoming more and more challenging in nanoscale technologies where parameters such as intra-die variation and interconnect capacitances are becoming predominant. All those effects have a negative impact in terms of cost and time to market. Finally, the fact that functional patterns are needed for the estimation process makes PMB approaches not suitable for general logic.

This paper proposed a new approach that uses three types of delay test patterns (TF, SDD, and PDLY) for AVS characterization during IC production, which serves as an alternative to the industry standard of using PMBs. This approach represents a powerful example of value-added testing, in which delay tests (already used during production) can replace a long and expensive process of PMB characterization, at low extra cost and can reduce time to market dramatically. Moreover, since delay test patterns target all gates and indirectly cover all

path-segments, they are better at representing performance than PMBs. As functional patterns are not used anymore, the testing approach could be a solution for general logic as well, not only for CPU and GPU. According to simulation results of the 29 ISCAS'99 benchmarks on 42 corners of a 28 nm FD-SOI library, using TF testing for performance estimation ends up with an inaccuracy of 5.33% and a standard deviation of 1.80%; using SDD for performance estimation ends up with an inaccuracy of 3.96% and a standard deviation of 1.59%; using PDLY for performance estimation results in an average error as low as 1.85% and standard deviation of only 1.34%, which makes PDLY the most accurate performance estimator for defining AVS voltages during production. Since TF testing does not necessarily target critical paths of the design, which might be a limitation of the model, performance estimation using TF showed less accuracy as compared to SDD and PDLY testing. Since SDD and PDLY test patterns allow us to focus on paths that are more critical, the results are very promising to improve performance estimation accuracy at the cost of extra patterns.

We also presented an analysis of the impact of technology scaling on the effectiveness of delay testing approaches using two technology nodes: 28nm and 65nm. The results show that the 65nm technology node exhibits the same trends identified for the 28nm technology node, namely that PDLY is the most accurate performance estimation method, while TF is the least accurate performance estimator. Based on the results, we also conclude that for each type of delay test, the larger the size of the used test pattern set, the more predictable the performance estimation will be. Therefore, depending on the time invested in testing during production, the accuracy of performance estimation using delay testing can be improved.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Al-Ars Z et al (2008) Test set development for cache memory in modern microprocessors. *IEEE Trans Very Large Scale Integration (TVLSI) Syst* 16(6):725–732
2. Brockman JB, Director SW (1989) Predictive subset testing: Optimizing IC parametric performance testing for quality, cost, and yield, vol 2
3. Burd TD et al (2000) A dynamic voltage scaled microprocessor system. In: *Proceedings of the IEEE international solid-state circuits conference (ISSCC)*, pp 294–295

4. Chan T, Kahng AB (2012) Tunable sensors for process-aware voltage scaling. In: Proceedings of the IEEE/ACM international conference on computer aided design (ICCAD), pp 7–14
5. Chan T et al (2012) DDRO: a novel performance monitoring methodology based on design-dependent ring oscillators. In: Proceedings of the IEEE international symposium on quality electronic design (ISQED), pp 633–640
6. Das P et al (2011) On generating vectors for accurate post-silicon delay characterization. In: Proceedings of the IEEE Asian test symposium (ATS), pp 251–260
7. Drake A et al (2007) A distributed critical-path timing monitor for a 65nm high-performance microprocessor. In: Proceedings of the IEEE international solid-state circuits conference (ISSCC), pp 398–399
8. Goldstein LH, Thigpen EL (1980) SCOAP: Sandia controllability/observability analysis program. In: Proceedings of the IEEE/ACM design automation conference, pp 190–196
9. Kim J, Horowitz MA (2002) An efficient digital sliding controller for adaptive power-supply regulation. *IEEE Journal of Solid-State Circuits (JSSC)* 37(5):639–647
10. Kruseman B, Majhi A, Gronthoud G (2007) On performance testing with path delay patterns. In: Proceedings of the IEEE VLSI test symposium (VTS)
11. Lee J et al (1999) IC performance prediction for test cost reduction. In: Proceedings of the IEEE international symposium on semiconductor manufacturing (ISSM), pp 111–114
12. Liu Q, Sapatnekar SS (2010) Capturing post-silicon variations using a representative critical path, vol 29
13. Li Zhang G et al (2016) EffiTest: Efficient delay test and statistical prediction for configuring post-silicon tunable buffers. In: Proceedings of the IEEE/ACM design automation conference (DAC)
14. Pant P, Skeels E (2011) Hardware hooks for transition scan characterization. In: Proceedings of the IEEE international test conference (ITC), pp 1–8
15. Sauer M et al (2012) On the quality of test vectors for post-silicon characterization. In: Proceedings of the IEEE European test symposium (ETS), pp 1–6
16. Shim KN, Hu J (2012) A low overhead built-in delay testing with voltage and frequency adaptation for variation resilience. In: Proceedings of the IEEE international symposium on defect and fault tolerance in VLSI and nanotechnology systems (DFT), pp 170–177
17. Tehranipoor M et al (2011) Test and diagnosis for small-delay defects. In: Springer science+business media LLC
18. Zain Ali NB et al (2006) Dynamic voltage scaling aware delay fault testing. In: Proceedings of the IEEE European test symposium (ETS)
19. Zandrahimi M, Al-Ars Z (2014) A survey on low-power techniques for single and multicore systems. In: Proceedings of the EAI international conference on context-aware systems and applications (ICCASA), pp 69–74
20. Zandrahimi M et al Industrial approaches for performance evaluation using on-chip monitors. In: The proceedings of the IEEE international design & test symposium (IDT 2016), 18–20 December 2016, Hammamet, Tunisia
21. Zandrahimi M et al (2016) Challenges of using on-chip performance monitors for process and environmental variation compensation. In: Proceedings of the IEEE Design, automation and test in europe conference (DATE), pp 1018–1019
22. Zandrahimi M et al (2017) Using transition fault test patterns for cost effective offline performance estimation. In: Proceedings of the IEEE international conference on design & technology of integrated systems in nanoscale Era (DTIS)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Mahroo Zandrahimi is a PhD student at the Quantum and Computer Engineering Department of the Delft University of Technology. She is performing her research in collaboration with STMicroelectronics in Grenoble, France, where she was based for a period of 2 years. Ms. Zandrahimi completed her BSc in Computer Engineering at Shahid Beheshti University of Iran, and MSc in Computer Architecture at Amirkabir University of Technology, both with honors. She published more than 10 papers in the field of chip performance characterization and AVS design optimization. Her research interests include low-power design, Design for Testability (DFT), fault-tolerance and dependability.

Philippe Debaud holds a PhD in Computer Science and an MSc in Electrical and Electronics Engineering from Paris VI University. Dr. Debaud is currently heading the Design for Test effort of the STMicroelectronics Imaging Division in Grenoble, France, where his work focuses on imager and sensors test optimization. Prior to joining the Imaging Division, Dr. Debaud spent a number of years in ST-Ericsson heading the company Design for Test department as well as several other STMicroelectronics divisions. He was also part of the French Atomic Agency, focusing on test quality and high volume low-power product testing.

Armand Castillejo graduated in Physics, Material Sciences and Microelectronics from Grenoble University. Mr. Castillejo is a Collaborative R&D Program Manager in STMicroelectronics in Grenoble, France, where he leads several RD Programs addressing several aspects of IoT (Internet of Things). His specific domain of interest concerns the development of platforms and solutions optimizing lowpower, cyber security, edge computing and connectivity requirements. Since 2015, the majority of his activities focus on medical projects developing specific Connected Medical Devices. Currently, he is the European Project Leader of the SERENE-IoT project aiming at developing disruptive secured and energy efficient Medical Devices (from specification to clinical evaluation) with major medical OEM manufacturers and key clinical partners.

Zaid Al-Ars is an associate professor at the Quantum and Computer Engineering Department of the Delft University of Technology, where he leads the research activities of the big data architectures specialization. His work focuses on developing computational infrastructures to enable efficient processing of data-centric applications using optimized solution alternatives for system performance, memory, power, etc. Dr. Al-Ars is also cofounder of Bluebee, a big data company specialized in high performance genomics solutions. Prior to joining Delft University of Technology, Dr. Al-Ars spent a number of years in the Product Engineering Group of Infineon Technologies and Siemens Semiconductors in Munich, Germany.