Low power IC design characterization techniques under process variations

Zandrahimi, Mahroo

DOI
10.4233/uuid:4f46e987-87a6-4f66-afa7-de9eacb8dc29

Publication date
2018

Document Version
Final published version

Citation (APA)
Zandrahimi, M. (2018). *Low power IC design characterization techniques under process variations*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:4f46e987-87a6-4f66-afa7-de9eacb8dc29

LOW POWER IC DESIGN CHARACTERIZATION TECHNIQUES UNDER PROCESS VARIATIONS

LOW POWER IC DESIGN CHARACTERIZATION TECHNIQUES UNDER PROCESS VARIATIONS

Proefschrift

ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft,

door

Mahroo ZANDRAHIMI

Master of Science in Computer Architecture
geboren te Birmingham, United Kingdom

Dit proefschrift is goedgekeurd door de promotoren:
Dr. Z. Al-Ars
Prof. dr. K.L.M. Bertels

Samenstelling promotiecommissie:

Rector Magnificus,	voorzitter
Dr. Z. Al-Ars,	Technische Universiteit Delft, promotor
Prof. dr. K.L.M. Bertels,	Technische Universiteit Delft, promotor
Independent members:	
Prof. dr. L. Koskinen,	University of Turku, Finland
Prof. dr. B. Nauta,	University of Twente, Netherlands
Dr. N.P. van der Meijs,	Technische Universiteit Delft
Prof. dr. L.C.N. de Vreede,	Technische Universiteit Delft
Dr. P. Debaud,	STMicroelectronics, France

Keywords: Adaptive voltage scaling, process variations, performance estimation, process monitoring boxes, delay testing, transition fault testing, path delay testing

This work is carried out under the BENEFIC project (CA505), a project labelled within the framework of CATRENE, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics.

Printed in the Netherlands

Dedicated to my family

ACKNOWLEDGEMENTS

I have encountered many good people over the years, who have all done their best to help me through this process of starting, working with, and finally, finishing this dissertation. Here, I would like to express my deepest gratitude. First and foremost, I would like to thank my supervisor and promotor, Dr. Zaid Al-Ars, who always believed in me, helped me gain my confidence and taught me many research skills and life lessons. I owe special thanks to him, since his careful supervision, helpful suggestions and generous support were really encouraging to me.

I would also like to express my deepest thanks to the head of QCE department, Prof. Koen Bertels, who gave me a sense of security and helped me every step of the way, Dr. Carmen G. Almudever, who treated me with the utmost kindness, Dr. Mottaqiallah Taouil who has always been so helpful and understanding, Ms. Lotte Ophey, my HR advisor, who heard my desperate cry for help at the time I were in need of emotional assistance. I would also like to thank the QCE staff, which include QCE secretaries Lidwina Tromp and Joyce van Velzen, as well as system administrator Erik for his administrative and technical assistance throughout. Finally, I would like to thank my fellow PhD students and friends, such as, Anh, Innocent, Hamid, Jintao, Lingling, Troya, Haji, Moritz, Imran, Shanshan, Nauman, Jian, Nader, and Ernst.

I met plenty of inspiring people throughout my stay in France to whom I will be eternally grateful. To name a few, Dr. Philippe Debaud who made me experience Many happy moments and made my work utterly enjoyable and I hope I will be so lucky to work with him again in the future. The next person I would like to mention is Armand Castillejo, who made my collaboration with STMicroelectronics in France possible and also made my stay there much more pleasant. Moreover, I take this opportunity to thank my colleagues, Matthieu Sautier, Mohamedarif Alarakhia, and Rachid Idrissi, members of DFT group in STMicroelectronics, as well as, Pierre Duclos from Synopsys, who were always helpful during my stay in STMicroelectronics.

I am also very grateful to Shermin and Stephan Danaie for their pure kindness and generously accepting me as a part of their family. Furthermore, I would like to express my appreciation to my friend Gerona who has always been there for me in the most desperate times of my life. Also, special thanks to my friends, Cédric, Clément, Frédéric, Annelise, David, and Audrey for their emotional support during my stay in Grenoble.

I am specially indebted to my life coach, Caroline Dessing, who helped me get back on my feet and start working towards achieving my goals. Next, Ilse Meezen, my yoga teacher, who has always selflessly devoted herself to me and encouraged me to find peace.

I would also like to thank dear Panida who taught me how to be confident and stand up for myself. I am also highly obliged to my friends Alireza and Barbara; Alireza for always truly understanding me and Barbara for being the most selfless person I have ever encountered in my life. Without their assistance, I would have never had the courage to finish this thesis. Furthermore, special thanks to my friends, Parvaneh, Sara, Negin, Mohadeseh M, Mohadeseh S, Razieh, Vida, Sima, Sahar, Samira, Mona, Farideh, Arezu, Shirin and Bahareh, who have always supported me and comforted me in my time of need, even though we were far away from each other.

Last but certainly not least, my family who have always been the light of my life even when everything seemed dark. They never doubted me for a second and have been truly supportive during all stages of my life.

Mahroo Zandrahimi
Delft, July 2018

SUMMARY

To overcome the increasing sensitivity to variability in nanoscale integrated circuits, operation parameters (e.g., supply voltage) are adapted in a customized way exclusively to each chip. AVS is a standard industrial technique which has been adopted widely to compensate for process, voltage, and temperature variations as well as power optimization of integrated circuits. For cost and complexity reasons, AVS techniques are usually implemented by means of on-chip performance monitors (so-called PMBs) allowing fast performance evaluation during production or run time. Such on-chip monitoring approaches estimate operation parameters either based on responses from performance monitors with no interaction with the circuit or by monitoring the actual critical paths of the circuit.

In this thesis, we focus on AVS techniques, which estimate operation parameters using responses from on-chip performance monitors with no interaction with the circuit during production. We discuss the challenges that these monitoring methodologies face with decreasing node sizes, in terms of accuracy and effectiveness. We show that the accuracy of these approaches is design dependent, and requires up to 15% added design margin. In addition, we show using silicon measurements of a nanometric FD-SOI device that the required design margin is above 10% of the clock cycle, which leads to significant waste of power.

In this thesis, we introduce the new method of using delay test patterns including TF, SDD, and PDLY test patterns for application of AVS during IC production. The proposed method is able to eliminate the need for PMBs, while improving the accuracy of performance estimation. The basic requirement of using delay-based AVS is that there should be a reasonable correlation between the frequency the chip can attain while passing all delay test patterns and the actual frequency of the chip. Based on simulation results of ISCAS'99 benchmarks with a 28 nm FD-SOI library, using delay test patterns result in an error of 5.33% for TF testing, an error of 3.96% for SDD testing, and an error as low as 1.85% using PDLY testing. Accordingly, PDLY patterns have the capacity to achieve the lowest error in performance estimation, followed by SDD patterns and finally TF patterns. We performed the same analysis using a 65 nm technology node, which showed the same results.

We also did two different silicon measurements on a 28 nm FD-SOI CPU to investigate the effectiveness of the TF-based approach. The results of the first case study on real silicon comparing the performance estimation using functional test patterns and the TF-based approach show a very close correlation between the two, which proves the effectiveness of the TF approach. The second case study compares the accuracy of voltage estimation using PMBs and the TF-based approach. The results show that the PMB approach can only account for 85% of the uncertainty in voltage measurements, which results in considerable power waste. In comparison, the TF-based approach can account for 99% of that uncertainty, thereby providing the ability to reducing that wasted power.

SAMENVATTING

Om de toenemende gevoeligheid voor variatie in geïntegreerde schakelingen te voorkomen, worden bedrijfsparameters (bijv. voedingsspanning) op een specifieke manier exclusief voor elke chip aangepast. AVS is een standaard techniek die vaker wordt toegepast om proces-, spanning- en temperatuurvariaties te compenseren. Vanwege kosten- en complexiteitsredenen worden AVS-technieken meestal geïmplementeerd door middel van on-chip prestatiemonitoren (PMB's genaamd) die snelle evaluatie van de prestaties mogelijk maken tijdens productie of gebruik. Dergelijke on-chip monitoring technieken berekenen de bedrijfsparameters op basis van responsen van de prestatiemonitoren of door het monitoren van de kritieke paden van het circuit. In dit proefschrift concentreren we ons op AVS-technieken, die de bedrijfsparameters berekenen met behulp van responsen van on-chip prestatiemonitoren zonder interactie met het circuit tijdens productie. We bespreken de uitdagingen die deze monitoringmethodologieën met zich meebrengen, in termen van nauwkeurigheid en effectiviteit. We laten zien dat de nauwkeurigheid van deze technieken afhankelijk is van het circuit en vereist een toegevoegde ontwerpmarge van ten minste 15%. Bovendien laten we met behulp van siliciummetingen van een nanometrisch FD-SOI chip zien dat de vereiste ontwerpmarge hoger is dan 10% van de klokperiode, wat leidt tot een aanzienlijke verspilling van energie. In dit proefschrift introduceren we de nieuwe methode voor het gebruik van delay-testpatronen inclusief TF-, SDD- en PDLY-testpatronen voor het uitvoeren van AVS tijdens IC-productie. De voorgestelde methode kan de behoefte aan PMB's elimineren, terwijl de nauwkeurigheid van de prestatiemeting wordt verbeterd. De basisvereiste voor het gebruik van op delay-testpatronen voor AVS is dat er een redelijke correlatie moet zijn tussen de frequentie die de chip kan bereiken tijdens testen en de werkelijke frequentie van de chip. Simulatie resultaten van ISCAS99 testcircuits met een 28 nm FD-SOI-bibliotheek laten zien dat het gebruik van delay-testpatronen maar kleine meetfouten veroorzaken, namelijk, 5,33% voor TF-testen, 3,96% voor SDD-testen en 1,85% voor PDLY-testen. We hebben dezelfde analyse uitgevoerd met behulp van een 65 nm technologie, dat dezelfde resultaten liet zien, wat aangeeft dat deze testgebaseerde benadering kan worden gebruikt voor verschillende technologieën. We hebben ook twee verschillende experimenten op silicium uitgevoerd op een 28 nm FD-SOI CPU om de effectiviteit van de op TF gebaseerde aanpak te onderzoeken. De resultaten van het eerste experiment, waarbij de prestatieberekening van functionele testpatronen vergeleken wordt met de TF gebaseerde aanpak, laten een zeer nauwe correlatie zien, wat de effectiviteit van de TF-aanpak aantoont. Het tweede experiment vergelijkt de nauwkeurigheid van spanningsberekening van PMB's met de op TF gebaseerde aanpak. De resultaten tonen aan dat de PMB-aanpak slechts 85% van de onzekerheid in spanningsmetingen kan identificeren, wat resulteert in aanzienlijk energieverspilling. Ter vergelijking: de op TF gebaseerde aanpak kan 99% van die onzekerheid identificeren, waardoor veel minder energieverspilling wordt veroorzaakt.

CONTENTS

1

INTRODUCTION

Power has been one of the primary design constraints and performance limiters in the semiconductor industry such that reducing power consumption can extend battery lifetime of portable systems, decrease cooling costs, as well as increase system reliability.

The continuous progress in microprocessors performance has been propelled mostly by technology scaling, which results in exponential growth both in transistor density and performance. However, as technology scaling enters the nanometer regime, CMOS devices are facing many problems such as increased leakage currents, large parameter variations, as well as low reliability and yield [1]. The inability to continue to lower the supply voltage halted the ability to increase the clock speed without increasing power dissipation. Therefore, in order to avoid encountering a stall in the future growth of computing performance, high performance microprocessors had to enter the multicore era [2]. However, the growth in the number of cores causes super-linear growth in non-core related area and power; accordingly, the power dissipation problem did not disappear with the shift towards the new multicore era [3, 4]. Therefore, in addition to a focus on multicore design and parallel processing, we need research and development focussed on much more power-efficient computing systems at various levels of abstraction.

In this chapter, Section 1.1 discusses the background and related work. This is followed by Section 1.2 which introduces the limitations of the state of the art industrial AVS methods, which is the reason of investigating new methods for AVS. Next, we define our contributions in Section 1.3. Finally, we describe the thesis organization in Section 1.4.

1.1. BACKGROUND AND RELATED WORK

1.1.1. LOW POWER TECHNIQUES

Figure 1.1 displays a system model that will be considered in this thesis. The model consists of a number of tiles (either a processor or memory), each of which contains a local power management (LPM) unit for local power optimizations. The model also contains a global power management (GPM) unit, which aims to reduce power considering all tiles and interactions among them. The figure also shows the interconnect, which is used for

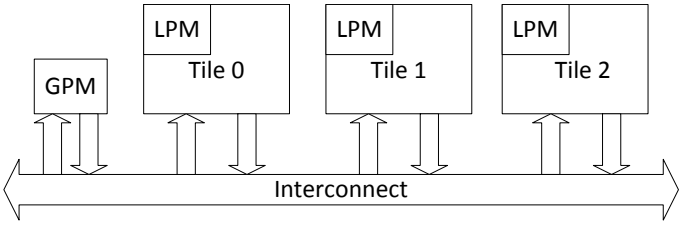

Figure 1.1: System model block diagram to classify power reduction techniques

the interaction among tiles and GPM. Notably, techniques used for LPM are applicable to both single and multicore systems. Based on Figure 1.1, power reduction techniques can be applied to either the tiles or the interconnects, whether inside or outside the cores.

A high-level taxonomy of the power reduction techniques for both single and multicore systems is illustrated in Figure 1.2. Many techniques have been proposed to achieve power reduction at different levels of abstraction, some of which require modification of the process technology, achieving power reduction during fabrication/design stage. Others are run-time techniques that require architectural support, and in some cases, technology support as well. Based on Figure 1.2, there are different techniques which aim to reduce power either during fabrication/design or runtime in the tiles. Power consumption of single and multicore systems can also be reduced in the interconnects or through adaptive voltage scaling techniques in the local and global power management units to dynamically manage power during run-time [5, 6].

More detailed survey on low power techniques for single and multicore systems is given in Chapter 2.

With the ongoing scaling of CMOS technologies, variations in process, supply voltage, and temperature (PVT) have become serious concern in integrated circuit design. Therefore, an individual safety margin for each variation source is added on the top of the supply voltage needed for the nominal case as depicted in Figure 1.3. However, this classical worst-case analysis is quite pessimistic and leads to wasting both system power and/or performance. To overcome this problem, various adaptive design strategies have been proposed. The basic idea is to adapt the supply voltage to the optimal value, based on the current operation conditions of the system so that power is saved; variations are compensated, while maintaining the desired performance.

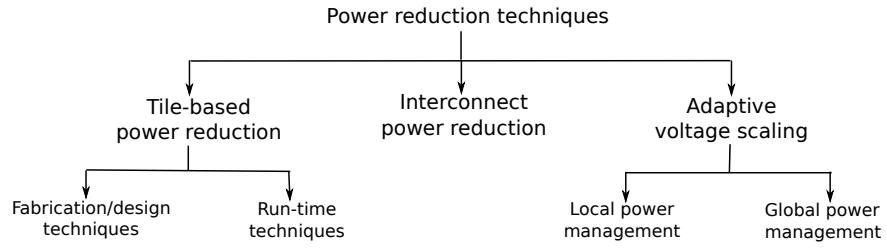

Figure 1.2: Taxonomy of various methods for total power reduction

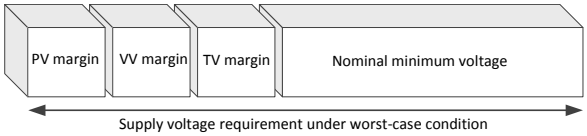

Figure 1.3: Schematic of the worst-case guard-banding approach (PV, VV, and TV stand for process, voltage, and temperature variations, respectively)

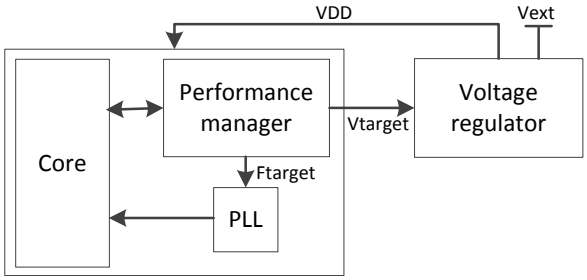

Figure 1.4: Architecture of an AVS system

Adaptive voltage scaling (AVS) systems are very efficient in saving power since the supply voltage has a profound impact on the operating frequency and power consumption of an integrated circuit. Typically, logic delay increases as V_{DD} reduces and power consumption increases super linearly with V_{DD}. Whenever maximum performance is not required, supply voltage can be scaled so that power can be saved while the system can still meet the timing constraints. Figure 1.4 shows the overall architecture of an AVS system [7]. The performance manager predicts performance requirements. Once performance requirement is determined, the performance manager sets the voltage and frequency to values that are just enough to accomplish the performance target of the system. The target frequency is sent to the phase-locked loop (PLL) to accomplish frequency scaling. Based on the target voltage, the voltage regulator is programmed to scale the supply voltage up/down until target voltage is achieved.

Thus, accurate circuit performance estimation is required to set the optimal voltage for the circuit so that the required performance is guaranteed. AVS techniques use on-chip performance monitors to estimate the actual performance of the circuit. Such on-chip performance monitors either have no interaction with the circuit or monitor the actual critical paths of the circuit. Based on this feature, we propose a taxonomy of process monitoring methodologies illustrated in Figure 1.5. According to this figure, AVS is done either using indirect measurement approaches or direct measurement approaches. Indirect measurement approaches estimate actual frequency of the circuit through correlating frequency responses of performance monitors to the circuit frequency, whereas, direct measurement approaches set the circuit operating parameters by monitoring the actual critical paths of the circuit. These two process monitoring methodologies (direct and indirect) will be discussed and illustrated in more details in the next section.

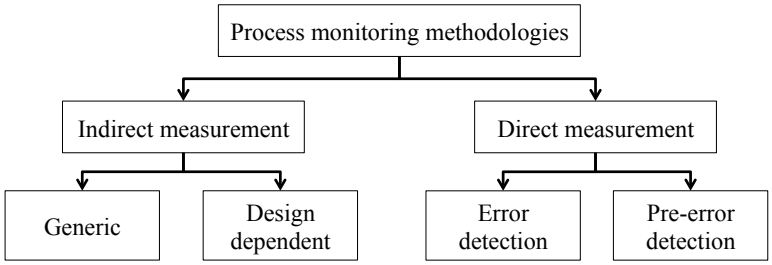

Figure 1.5: Classification of process monitoring methodologies

1.1.2. PROCESS MONITORING METHODOLOGIES

INDIRECT MEASUREMENT APPROACHES

These approaches embed one or various performance monitors in the chip structure. Due to within-die variations, it is more efficient to place various performance monitors close or inside the block which is being monitored so that all types of process variations are captured and taken into account for voltage adaptation. The number of performance monitors depends on the size of the chip. There is no interaction between performance monitors and the circuit.

To be able to estimate the circuit frequency based on performance monitor responses during production, the correlation between performance monitors and circuit frequency should be measured during characterization, which is an earlier stage of manufacturing [8]. This procedure is done for the amount of test chips representative of the process window to find the correlation between performance monitors and circuit frequencies. Once the performance monitors are tuned to the design during characterization, they are ready to be used for voltage estimation for each chip during production. Figure 1.6 shows an example of a chip with multiple voltage islands, among which performance monitors are distributed. During production, based on the frequency responses from these monitors, the circuit frequency is estimated so that operating parameters can be adapted to each voltage domain of the chip.

Various performance monitoring structures have been proposed from simple generic ring oscillators to more complicated design dependent critical path replicas. The technique presented in [9] implements replica-paths, representing the critical paths of the circuit. Alternatively, the critical path replica can be replaced by fan-out of 4 (FO4) ring oscillator [10] or a delay line [11]. They claim that with varying operating conditions, the timing of monitors will change similarly to the actual critical path. Moreover, the method presented in [12] synthesizes a single representative critical path (RCP) for post-silicon delay prediction. They claim that the RCP is designed such that it is highly correlated to all critical paths for some expected process variations.

However, as technology scaling enters the nanometer regime, specially from 45 nm onwards, finding one unique critical path has become impossible. Depending on process and operational conditions (the process corner, voltage and temperature variations, and also workload) many different timing paths might become critical. Therefore for real circuits, the concept of finding only one critical path and creating a critical path replica

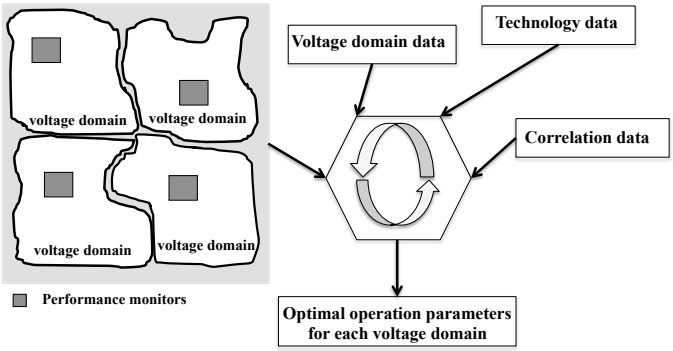

Figure 1.6: Operating parameter estimation using indirect measurement approaches

as a performance monitor is too simplistic. As a result, regardless of using generic ring oscillators or design dependent replica paths, the characterization phase should be done to find the correlation between monitoring responses and the actual performance of the circuit.

DIRECT MEASUREMENT APPROACHES

Direct measurement approaches estimate operation parameters by monitoring actual critical paths of the circuit. These approaches add one in-situ delay monitor per critical path. In-situ delay monitors are special latches or flip-flops, included at the end of critical paths to report the timing behavior of the circuit [13]. Circuit delay characterization using in-situ delay monitors can be done in two different ways. The first is by observing the regular operation of a circuit and to detect timing errors in the circuit itself during operation. With this error information, the critical operation parameters, which are needed for correct operation, can be determined. The second possibility is to observe an over-critical system. Here, a test module which is always slower than the most critical part of the chip is observed, and as soon as the test module fails, the system predicts a delayed data transition called a pre-error [14].

For the in-situ monitors, which are able to detect timing errors, error recovery circuits are needed to repeat single computations after malfunction. In contrast, for in-situ approaches which detect pre-errors, no additional hardware effort and complexity for the recovery circuitry is needed, thus, these approaches are easier to manage. Figure 1.7 shows an in-situ delay flip flop which detects pre-errors. These in-situ flips flops detect pre-errors when the timing slack in critical paths drops below a certain value. The idea is to reduce the operation parameters as long as no pre-error is detected and to raise the operation parameters as soon as the pre-error rate is above a certain value.

With regard to accuracy and tuning effort, direct measurement 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 indirect measurement approaches, since there is no interaction between performance monitors and the circuit, the correlation between performance monitor responses and the actual performance of the circuit is estimated

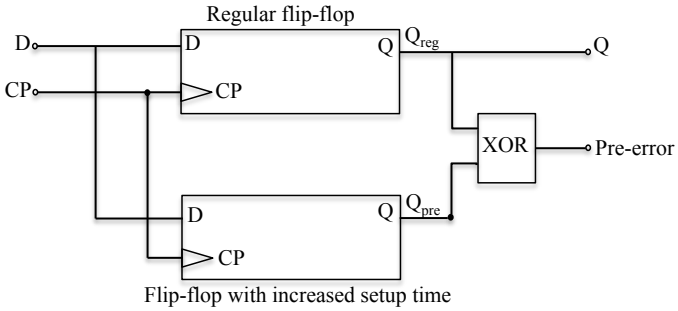

Figure 1.7: Structure of in-situ flip-flops which detect pre-errors

during the characterization phase using an amount of test chips representative of the process window. Since there are discrepancies in the responses of the same performance monitors from different test chips, the estimated correlation between the frequency of performance monitors 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, direct measurement approaches always win.

On the other hand, in terms of planning effort and implementation risk, direct measurement 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. Therefore, indirect measurement approaches are considered more acceptable in terms of planning and implementation risk, since there is no interaction between performance monitors and the circuit. Hence, performance monitors can even be placed outside the macros being monitored, but not too far due to within die variations. Consequently, indirect measurement 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 on the design is very low. As a result, according to the application, one can decide which technique more suits a specific design. For medical applications for example, 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 thesis our focus is on AVS implementation on devices used for nomadic applications. Thus, the performance monitors (which we call Process Monitor Boxes (PMBs) from now on) we consider in this thesis use indirect measurement approaches for performance estimation. PMBs are ring oscillators designed based on the most used cells extracted from the potential critical paths of the design, reported by static timing analysis. So, based on the design, some standard logic cells are put in an oscillator to form performance monitors, which will be distributed among the chip to capture all kinds of variations. During characterization, PMBs are tuned to the design so that during pro-

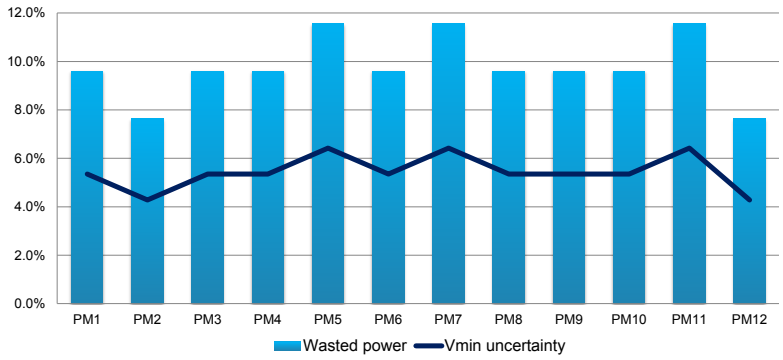

Figure 1.8: Inaccuracy in the optimal operating voltages estimated using different PMBs

duction, according to the frequency responses of PMBs, the voltages are adapted to each chip.

A more detailed survey and discussion of AVS approaches is given in Chapter 2 and in Chapter 3.

1.2. 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 time—**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 rather expensive.

- **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.

- **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.

- **Limited effectiveness—**Since there are discrepancies in the responses of the 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. To validate our claim of low accuracy of PMB approaches, we have done silicon measurement on 625 devices manufactured using nanometric FD-SOI technology. 12 performance monitors (PM) are embedded in each device. We measured the amount of Vmin discrepancy for all 12 monitors, the result of which is presented in Figure 1.8. This figure also presents the wasted power as a results of inaccuracy in Vmin estimation using performance monitors. Results show that optimal 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.

Therefore, we can conclude that 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 older well-understood technologies that have become robust with time 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.

1.3. OUR CONTRIBUTION

We can sum up our contributions in this thesis as follows.

1. We propose a comprehensive taxonomy of power reduction techniques for both tiles and the interconnect as well as run-time techniques for adaptive voltage scaling. We discuss several techniques from each class in the taxonomy along with examples as well as reported power reduction values.

2. An overview of various on-chip performance monitors for online and offline AVS including a discussion of the pros and cons of each approach.

3. We Investigate the limitations of critical path replica performance monitors in terms of accuracy and effectiveness for ISCAS'99 benchmarks using the Nangate 45 nm open cell library with 4 different process corners.

4. A detailed investigation of PMB approaches in terms of accuracy and effectiveness using 29 ISCAS'99 bench- marks with an industrial grade 28 nm FD-SOI library for 42 different process corners with different characteristics in terms of process and environmental variations as well as aging.

5. Proposing the new concept of using delay testing including transition fault testing (TF), single delay defect testing (SDD), and path delay testing (PDLY) for performance estimation during production as an alternative for PMBs.

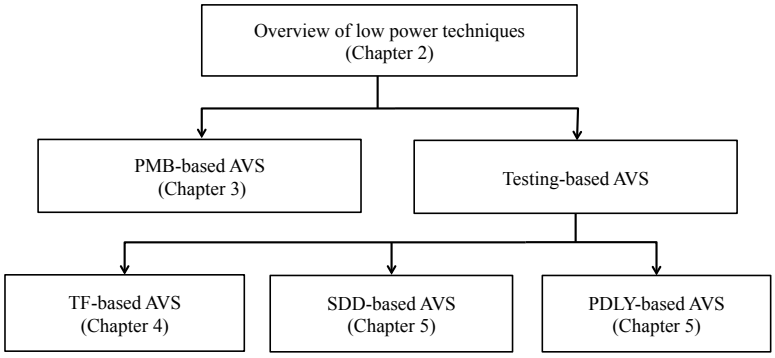

Figure 1.9: Overview of the different thesis topics

6. A detailed investigation on the delay testing approach including TF, PDLY, and SDD in terms of accuracy and effectiveness using 29 ISCAS'99 benchmarks with 28 nm FD-SOI library for 42 different process corners with different characteristics in terms of process and environmental variations as well as aging.

7. A study on the impact of technology scaling on accuracy and effectiveness of the delay testing approach using 65 nm and 28 nm FD-SOI libraries.

1.4. THESIS ORGANIZATION

The various chapters of the thesis and their relationships is presented in Figure 1.9. The thesis is organized as follows.

In **Chapter 2**, we give a survey of low power techniques for single and multicore systems.

In **Chapter 3**, we discuss the state of the art for AVS techniques that are currently being used in industry. We also introduce their limitations in terms of efficiency and cost.

In **Chapter 4**, we introduce our new proposal for AVS using Transition Fault test patterns (TF).

In **Chapter 5**, we discuss our new AVS technique using Single Delay Defect (SDD) and Path Delay (PDLY) testing.

In **Chapter 6**, we investigate on the impact of technology scaling on the effectiveness of AVS techniques using delay testing.

In **Chapter 7**, we summarize the finding s of the thesis and present the conclusions.

REFERENCES

[1] Y.B. Kim, *Challenges for Nanoscale MOSFETs and Emerging Nanoelectronics*, Trans. on Electrical and Electronic Materials, vol. 11, pp. 93-105, 2010.

[2] S.H. Fuller and L.I. Millett, *The Future of Computing Performance: Game Over or Next Level?*, The National Academy of Sciences, 2011.

[3] L. Spracklen and S.G. Abraham, *Chip Multithreading: Opportunities and Challenges*, in HPCA, pp. 248-252, 2005.

[4] H. Esmaeilzadeh, et al., *Dark Silicon and the End of Multicore Scaling*, in ISCA, vol. 46, pp. 5–26, 2011.

[5] Z. Al-Ars, *DRAM Fault Analysis and Test Generation*, Delft University of Technology, Delft, Netherlands, June, 2005.

[6] Z. Al-Ars, S. Hamdioui, G. Gaydadjiev, S. Vassiliadis, *Test Set Development for Cache Memory in Modern Microprocessors*, Trans. on VLSI, vol. 16, no. 6, pp. 725-732.

[7] M. Elgebaly and M. Sachdev, *Variation-Aware Adaptive Voltage Scaling System*, in TVLSI, vol. 15, no. 5, pp. 560-571, 2007.

[8] T. Chan and A.B. Kahng, *Tunable Sensors for Process-Aware Voltage Scaling*, in ICCAD, pp. 7-14, 2012.

[9] A. Drake, et al., *A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor*, in ISSCC, pp. 398-399, 2007.

[10] TD. Burd, et al., *A dynamic voltage scaled microprocessor system*, in ISSCC, pp. 294-295, 2000.

[11] J. Kim and M.A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IJSSC, vol. 37, no. 5, pp. 639-647, 2002.

[12] Q. Liu and S.S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.

[13] M. Wirnshofer, et al., *A Variation-Aware Adaptive Voltage Scaling Technique based on In-Situ Delay Monitoring*, in DDECS, pp. 261-266, 2011.

[14] M. Eireiner, et al., *In-Situ Delay Characterization and Local Supply Voltage Adjustment for Compensation of Local Parametric Variations*, in IJSSC, vol. 42, no. 7, pp. 1583-1592, 2007.

2

LOW POWER TECHNIQUES FOR SINGLE AND MULTICORE SYSTEMS

SUMMARY
This chapter surveys state of the art low-power techniques for both single and multicore systems. Based on our proposed power management model for multicore systems, we present a classification of total power reduction techniques including both leakage and active power. According to this classification, three main classes are discussed: power optimization techniques within the cores, techniques for the interconnect and techniques applicable for the whole multicore system. This chapter describes several techniques from these classes along with a comparison. For the whole multicore system, we focus on adaptive voltage scaling and propose a comprehensive taxonomy of adaptive voltage scaling techniques, while considering process variations.

This chapter is based on the following paper.

Zandrahimi, M.; Al-Ars, Z., *A Survey on Power Low-Power for Single and Multicore Systems*, International Conference on Context-Aware Systems and Applications (ICCASA), 15-16 October 2014, Dubai, United Arab Emirates.

A Survey on Low-Power Techniques for Single and Multicore Systems

Mahroo Zandrahimi
Delft University of Technology
Delft, the Netherlands
m.zandrahimi@tudelft.nl

Zaid Al-Ars
Delft University of Technology
Delft, the Netherlands
z.al-ars@tudelft.nl

ABSTRACT

This paper surveys state of the art low-power techniques for both single and multicore systems. Based on our proposed power management model for multicore systems, we present a classification of total power reduction techniques including both leakage and active power. According to this classification, three main classes are discussed: power optimization techniques within the cores, techniques for the interconnect and techniques applicable for the whole multicore system. This paper describes several techniques from these classes along with a comparison. For the whole multicore system, we focus on adaptive voltage scaling and propose a comprehensive taxonomy of adaptive voltage scaling techniques, while considering process variations.

1. INTRODUCTION

Power has been one of the primary design constraints and performance limiters in the semiconductor industry such that reducing power consumption can extend battery lifetime of portable systems, decrease cooling costs, as well as increase system reliability.

The continuous progress in microprocessors has been maintained mostly by technology scaling, which results in exponential growth both in device density and performance. However, as the technology scaling enters nanometer regime, CMOS devices are facing many problems such as increased leakage currents, large parameter variations, low reliability and yield [1]. The inability to continue to lower the supply voltage halted the ability to increase the clock speed without increasing power dissipation. Therefore, in order to avoid encountering a stall in the future growth of computing performance, high performance microprocessors had to enter the multicore era [2]. However, the growth in the number of cores causes super-linear growth in non-core area and power; accordingly, the power dissipation problem did not disappear in the new multicore regime [3, 4]. Therefore, in addition to a focus on multicore design and parallel processing, we need research and development on much more power-efficient computing systems at various levels of abstraction.

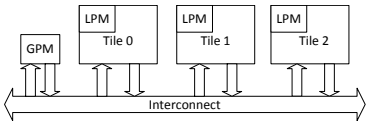

Figure 1: System model block diagram

There are various power reduction techniques published in the literature. This paper provides a survey of these techniques. Fig. 1 displays a system model that will be considered in this survey. The model consists of a number of tiles (either a processor or memory), each of which contains a local power management (LPM) unit for local power optimizations. The model also contains a global power management (GPM) unit, which aims to reduce power considering all tiles and interactions among them. The figure also shows the interconnect, which is used for the interaction among tiles and GPM. Notably, techniques used for LPM are applicable to both single and multicore systems. Based on Fig. 1, power reduction techniques can be applied to either the tiles or the interconnects, whether inside or outside the cores.

A high-level taxonomy of the power reduction techniques for both single and multicore systems is illustrated in Fig. 2. Many techniques have been proposed to achieve power reduction at different levels of abstraction, some of which require modification of the process technology, achieving power reduction during fabrication/design stage. Others are run-time techniques that require architectural support, and in some cases, technology support as well. Based on Fig. 2, there are different techniques which aim to reduce power either during fabrication/design or runtime in the tiles. Power consumption of single and multicore systems can also be reduced in the interconnects or through adaptive voltage scaling techniques in the local and global power management units to dynamically manage power during run-time. The contributions of this survey are as follows:

• We propose a comprehensive taxonomy of power reduction techniques for both tiles and the interconnect as well as run-time techniques for adaptive voltage scaling.

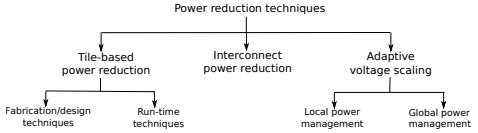

Figure 2: Taxonomy of total power reduction

• We discuss several techniques from each class in the taxonomy along with examples as well as reported power reduction values.

• We address various design and manufacturing issues, which degrade the effectiveness of power reduction techniques such as process and environmental variations and describe several low-power techniques considering these effects.

The rest of this paper is organized as follows. Section 2 presents low-power techniques that are applied either during fabrication/design or run-time stage to the tiles. Section 3 discusses interconnect low-power techniques that are applied dynamically during run-time. Section 4 specifically focuses on adaptive voltage scaling, which is widely used for run-time power optimization under process variations. Finally Section 5 concludes the paper.

2. TILE-BASED POWER REDUCTION

In this section we discuss the fabrication/design as well as run-time techniques for power reduction in the tiles for both single and multicore systems from architecture level to circuit level.

Power consumption of the tiles of single and multicore systems can be diminished at different levels of abstraction from system to layout, among which we will investigate various techniques at architecture, gate, and circuit levels in details. Fig. 3 illustrates a taxonomy of techniques for power reduction in the tiles from architecture to circuit level.

Based on Fig. 3, the tile power at architecture level can be cut back through low power control logic designs, low power memory hierarchies, and low power processor architectures. To explain low power control logic designs, assume the control logic of a processor as a finite state machine (FSM), which activates the appropriate circuitry for each state. Accordingly, optimizations in FSMs can be done for power reduction. Encoding FSM states to minimize the switching activity, or decomposing the FSM into sub-FSMs and activating only the circuitry needed for the currently executing sub-FSM are some examples of FSM optimizations throughout the processor [6]. A summary of attainable power reduction from this and other techniques is given in Table 1. Applying both of these techniques at the same time reduces power from 30-90%, while increasing area from 20-120%.

Another architecture level solution could be designing low power memories and memory hierarchies. Power dissipation in memories can be diminished in two ways, either by reducing the power dissipated in a memory access, or by reducing the number of memory accesses [5]. Moreover, splitting memory into smaller sub-systems is an effective way to reduce power consumed in a memory access. This can be done by partitioning memory into smaller, independently accessible components in different granularities so that only the needed circuitry is activated in each memory access [7]. A combination of subbanking, multiple line buffers and bit-line segmentation can reduce the on-chip cache power dissipation by as much as 75% in a technology-independent manner without compromising the processor cycle time. Augmenting the memory hierarchy with specialized cache structures is another popular method to save power by reducing memory hierarchy accesses. A simple example is a trace cache, which stores traces of instructions in their executed order rather than their compiled order. Hence, if an instruction sequence is already in the trace cache, it does not need to be fetched from the instruction cache and can be decoded directly from the trace cache [8]. However, conventional trace caches (CTC) may increase power in the fetch unit because of the simultaneous access to both the trace cache and the instruction cache. Dynamic direction prediction-based trace cache (DPTC), which avoids simultaneous accesses to the trace cache and the instruction cache achieve 38.5% power reduction over CTC, while only trading a 1.8% performance overhead compared to CTC [8].

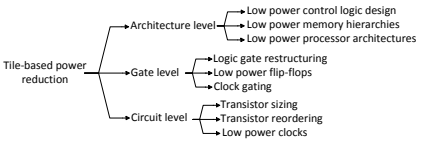

Figure 3: Taxonomy of tile-based power reduction

Another method to save power at architecture level is through adaptive processor architectures, which aim to save power by activating minimum hardware resources needed for the code that is executing. Adaptive caches and adaptive instruction queues are two examples. In an adaptive cache, storage elements (lines, blocks, or sets) can be selectively activated based on the workload. One example of such a cache is the drowsy cache whose lines can be placed in a drowsy mode where they dissipate minimal power, but retain data during drowsy mode and can be activated instantly [9]. In adaptive instruction queues, only the partitions that contain the currently executing instructions are activated at any time. For example, the heuristic proposed in [10], periodically measures the IPC (instructions per cycle) over fixed length intervals. If the IPC of the current interval is smaller than the previous interval, the size of the instruction queue is increased to enhance the throughput. The drowsy cache technique reduces power up to 53% with a performance overhead of 4.06-12.46%. Also, the adaptive instruction queue method achieves up to a 70% power reduction, while the complexity of the additional circuitry needed to achieve this result is almost negligible.

At gate level, logic gate restructuring is one simple method for power reduction. The idea is that since there are many ways to build a circuit out of logic gates, thus, how to arrange the gates and their input signals is important to power consumption [5]. Another possible solution is using low power flip-flops. Power consumption in flip-flops consists of the power dissipated in the clock signal, internal switching, and output transitions. Most of these low power designs for flip-flops reduce the switching activity or the power dissipated by the clock signal. Another method, which is very effective for power reduction at gate level is clock gating. Since clock is always active, and makes two transitions per cycle, it consumes about 40% of total processor power, so clock gating which inhibits clock to unused blocks is useful for power reduction.

Transistor sizing reduces the width of transistors based on the fact that reducing the width of transistors causes an increase in transistor delay, which leads to dynamic power reduction. Thus, the transistors that lie away from the critical paths of a circuit are usually the best candidates for this technique. Algorithms for applying this technique usually associate with each transistor a tolerable delay, which varies depending on how close the transistor is to the critical path. These algorithms then try to scale each transistor to be as

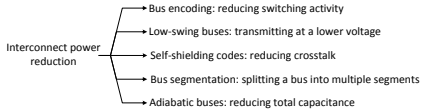

Figure 4: Taxonomy of Interconnect power reduction

small as possible without violating its tolerable delay [11]. Up to 15.3% power reduction can be achieved when 20% of the transistors are resized.

At circuit level, transistor reordering rearranges transistors to minimize their switching activity as their arrangement in a circuit affects power consumption [13, 14]. Another method is using low power clocks such as half-frequency and half-swing clocks, which reduce frequency and voltage respectively. Traditionally, hardware events such as register file writes occur on a rising clock edge. Half-frequency clocks synchronize events using both edges, and they tick at half the speed of regular clocks, thus cutting clock switching power in half. Reduced-swing clocks also often use a lower voltage signal, and hence reduce power quadratically [12]. As can be seen in Table 1, with transistor reordering, power can be reduced by up to 18% with minimum area and no performance overhead. The half-swing clocking scheme cuts power back by up to 67% in the whole chip and 75% in the clocking circuitry with minimal speed degradation.

3. INTERCONNECT POWER REDUCTION

Interconnects dissipate power due to switching of interconnect capacitances. Since efforts to improve chip performance lead to smaller chips with more transistors and more densely packed wires carrying larger currents [15], there arise additional sources of power consumption such as crosstalk. Therefore, power dissipating in interconnects has become one of the important contributors to total chip power consumption. Several methods have been proposed to cut back power consumption in interconnects, each of which tries to reduce power by focusing on a different aspect of power dissipation in the interconnect as depicted in Fig. 4.

A popular way to diminish interconnect power consumption is to reduce switching activity using intelligent bus encoding systems such as bus-inversion, which ensures that at most half of the bus wires switch during a bus transaction [16]. However, because of the cost of the logic required to invert the bus lines, this technique is mainly used in external buses rather than the internal chip interconnect. For every data transmission, the number of wires that switch depends on the current and previous values transmitted. If the Hamming distance between these values is more than half the number of wires, then most of the wires on the bus will switch current. To prevent this, bus-inversion transmits the inverse of the intended value and asserts a control signal alerting recipients of the inversion. For example, if the current binary value to transmit is 110 and the previous was 000, the bus instead transmits 001, the inverse of 110. This technique decreases the I/O peak power dissipation by 50% and the I/O average power dissipation by up to 25%.

Low swing buses transmit the same information but at a lower voltage [17]. Traditionally, logic one is represented by $+5$ volts and logic zero is represented by -5 volts. However, in a low-swing system, logic one and zero are encoded using lower voltages, such as $+300$mV and -300mV. The input signal is split into two signals of opposite polarity bounded by a smaller voltage range. The receiver sees the difference between the two transmitted signals as the actual signal and amplifies it back to normal voltage. This system has several advantages in addition to reduced power consumption. It is immune to crosstalk and electromagnetic radiation effects. Since the two transmitted signals are close together, any spurious activity will affect both equally without affecting the difference between them. However, the costs of increased hardware at the encoder and decoder should be considered. These buses decrease power from 62-78% with approximately 45% performance overhead.

As mentioned above, another source of power consumption in interconnects is crosstalk, which is false activity caused by activity in neighboring wires. One way of reducing crosstalk is to insert a shield wire between adjacent bus wires [18]. Since the shield remains deasserted, no adjacent wires switch in opposite directions, however, this solution doubles the number of wires. Another alternative is using coding systems which are resistant to crosstalk such as self-shielding codes [19, 20]. Just like traditional bus encoding system, a value is encoded and then transmitted. However, this system avoids opposing transitions on adjacent bus wires.

Bus segmentation is another effective technique for interconnect power reduction. In a traditional shared bus architecture, the entire bus is charged and discharged upon every access. Segmentation splits a bus into multiple segments connected by links that regulate the traffic between adjacent segments. Links connecting paths essential to a communication are activated independently, allowing most of the bus to remain powered down. Ideally, devices communicating frequently should be in the same or nearby segments to avoid powering many links. There are different algorithms for partitioning a bus into segments to benefit from this property as much as possible [21]. This technique achieves 24.6-37.21% power reduction with 6% area overhead.

Another solution to reduce power in interconnects is to reduce total capacitance, which is the principal behind adiabatic circuits [22]. These circuits reuse existing electrical charge to avoid creating new charge. In a traditional bus, when a wire becomes deasserted, its previous charge is wasted. A charge-recovery bus recycles the charge for wires about to be asserted. The saved power depends on transition patterns. No energy is saved when all lines rise. The most energy is saved when an equal number of lines rise and fall simultaneously. The biggest drawback of adiabatic circuits is the delay for transferring shared charge. This technique can achieve 28% power reduction.

4. ADAPTIVE VOLTAGE SCALING

With the on going scaling of CMOS technologies, variations in process, supply voltage, and temperature (PVT) have become serious concern in integrated circuit design. Depending on their spatial correlation, process variations can be divided into three groups. Die-to-die (D2D) variations have a correlation distance larger than the die size, i.e., all transistors on a chip are affected the same way. Within-die (WID) variations have a correlation distance smaller than the chip size. Random variations are not correlated at all; every transistor is affected individually. Environmental variations such as power supply noise and crosstalk have also gained significance with increasing current densities and reduced geometric dimensions [32].

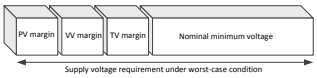

Figure 5: Schematic of the worst-case guard-banding approach (PV, VV, and TV stand for process, voltage, and temperature variations, respectively)

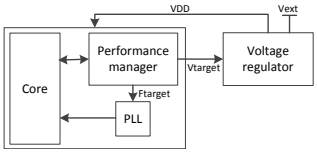

Figure 6: Architecture of an AVS system

Therefore, an individual safety margin for each variation source is added on the top of supply voltage needed for the nominal case as depicted in Fig. 5. However, this classical worst-case analysis is quite pessimistic and leads to power and performance be wasted. To overcome this problem, various adaptive design strategies have been proposed. The basic idea is to adapt the supply voltage to the optimal value, based on the current operation conditions of the system so that power is saved; variations are compensated, while maintaining the desired performance.

In this section, LPM techniques which are used in both single and multicore systems are explored. Specifically we focus on adaptive voltage scaling, which is widely used for run-time power optimization under process variations. In addition, we discuss GPM techniques which are specialized for multicore systems.

4.1 Local power management unit

Adaptive voltage scaling (AVS) systems are very efficient in saving power since the supply voltage has a profound impact on the operating frequency and power consumption of an integrated circuit. Typically, logic delay increases as V_{DD} reduces and power consumption increases super linearly with V_{DD}. Whenever maximum performance is not required, supply voltage can be scaled so that power can be saved while the system can still meet the timing constraints. Fig. 6 shows the overall architecture of an AVS system [28]. The performance manager predicts performance requirements. Once performance requirement is determined, the performance manager sets the voltage and frequency just enough to accomplish the performance target of the system. The target frequency is sent to the phase-locked loop (PLL) to accomplish frequency scaling. Based on the target voltage, the voltage regulator is programmed to scale the supply voltage up/down until target voltage is achieved.

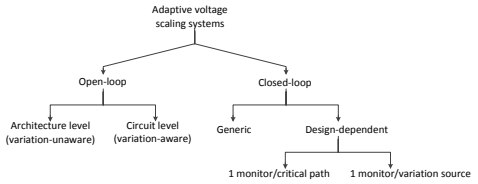

Figure 7: Taxonomy of adaptive voltage scaling systems

Thus, accurate circuit performance estimation is required so that the actual performance of the core running under the scaled voltage is monitored to guarantee a fail-safe operation, while maintaining the required performance [28]. A taxonomy of AVS systems is illustrated in Fig. 7. Based on whether the performance estimation is done early in manufacturing or during run-time, these techniques can be classified as either open or closed-loop [25]. The following subsections explore the commonly used AVS techniques.

4.1.1 Open-loop adaptive voltage scaling

A typical open-loop adaptive voltage scaling system creates a pre-characterized LUT to find the corresponding minimum voltage for a given frequency target. Conventionally, the voltage levels for each domain, as well as the mapping between frequencies and voltages are determined at architecture level without considering variations. One example is the three domain dynamic voltage frequency scaling (DVFS) power management scheme proposed in [26]. In this architecture level technique, the voltage and frequency of each power domain are dynamically scaled according to the performance requirement of each domain. They assumed that each domain has a specific requirement of voltage and frequency due to different workloads that they execute. Using three power domains diminishes power by up to 65% compared to a single domain, while imposes 2.6% area and 9.5% power overhead on the system.

However, with the increasing effect of process variations as a result of technology scaling, the research has become more focused towards the variation-aware adaptive voltage scaling techniques at circuit level. A technique proposed in [27], utilizes a user and process driven dynamic voltage and frequency scaling scheme to adapt voltage to the frequency of a microprocessor in real-time according to processor needs. User-driven frequency scaling (UDFS) uses direct user feedback to determine the processor frequency for individual users. Process-driven voltage scaling (PDVS) creates an LUT which maps frequency and temperature to the operating minimum voltage considering process variations. Using both of these techniques at the same time reduces power by up to 50% for single task and 70% for multi-task workloads compared to Windows XP DVFS. However, since these techniques do not have a feedback mechanism, the LUT is heavily guard-banded to ensure reliable system operation which results in performance and energy wastes. At the same time, characterizing the LUT is a time consuming and expensive procedure. Thus, closed-loop schemes which take advantage of feedback mechanisms during run-time are more efficient in saving power.

4.1.2 Closed-loop adaptive voltage scaling

A closed-loop adaptive voltage scaling system adjusts supply voltage by probing actual chip performance using on-chip monitors, thus, margin required by open-loop systems can be recovered. To track timing performance of a chip, many approaches have been proposed. Based on Fig. 7, in terms of design point of view, performance monitors are classified into design dependent and generic[24].

Generic performance monitors

Generic performance monitors range from simple inverter-based ring oscillators [29] to more complex process-specific ring oscillators (RO) [30] and also alternative monitoring

Table 1: Reported power reduction values

Reference	Section	Technique	Power reduction	Comments
[6]	II.A	Encoding FSM & decomposition to sub-FSMs	30% to 90%	20% to 120% area overhead
[7]	II.A	Splitting memory into smaller sub-systems	75%	sub-banking, bit-line segmentation, multiple-line buffers - no performance overhead
[8]	II.A	DPTC	38.5%	1.8% performance overhead over CTC
[9]	II.A	Drowsy cache	53%	4.06% to 12.46% performance overhead
[10]	II.A	Adaptive instruction queue	70%	Complexity of additional circuitry is negligible
[5]	II.A	Clock gating	up to 40%	small area overhead
[11]	II.A	Transistor sizing	up to 15.3%	20% of transistors are resized
[14]	II.A	Transistor reordering	18%	minimum area overhead, no performance overhead
[12]	II.A	Half-swing clock	67%-75%	small speed degradation
[16]	II.B	Bus inversion	50%-25%	peak and average power reduction of I/O
[17]	II.B	Low swing bus	62% to 78%	45% performance overhead
[21]	II.B	Bus segmentation	24.6% to 37.21%	6% area overhead
[22]	II.B	Adiabatic bus	28%	-
[26]	III.A	Three domain DVFS	65%	power overhead: 9.5%, area overhead: 2.6% compared to single domain
[27]	III.A	UDFS-PDVS	50% to 75%	compared to Windows XP DVFS
[33]	III.A	Universal delay line	13% to 27%	area overhead: 0.01%, power overhead is negligible
[23]	III.A	In-situ delay monitoring (over-critical)	13.5%	compared to the worst-case design, prediction error rate: 1.10^{-15}
[32]	III.A	In-situ delay monitoring (regular)	14%	power overhead: 0.5%, area overhead: 10%
[34]	III.A	Critical path replica	11% to 78%	highly dependent on the benchmark
[36]	III.A	RCP	31%	smaller guard-band than critical path replica, prediction error rate: 2.8%

structure such as PLLs [31]. Although generic monitors are very simple to design and can be used in any product without customizations, they are inadequate to capture design characteristics, and there will be a large error in the measurements due to the difference in gate structure between the actual critical path and the delay monitor. So, delay estimation using generic monitors is less accurate and sometimes incurs larger margins. However, the generic performance monitor proposed in [33] tries to minimize the errors due to gate structure difference by utilizing certain chain of delay gates, as well as the errors due to the within die variations by distributing monitors among the chip. Each performance monitor, which is called a universal delay line, contains a ring oscillator and a counter. The ring oscillator is designed with double stacked NMOSs and PMOSs since this gate structure is the most dominant component in the critical path delay, which minimizes the error due to the gate structure difference. This technique decreases power by up to 27% with a negligible area overhead.

Design-dependent performance monitors

According to Fig. 7, some of the design-dependent techniques implement one monitor per variation source, while the others implement one monitor per critical path. One group of methods that utilize one monitor per critical path are based on in-situ delay monitors, which are special latches or flip-flops, included at the end of critical paths to report the timing behavior of the circuit in order to form a closed loop configuration for voltage adaptation [32]. Circuit delay characterization using in-situ delay monitors can be done in two different ways. The first is by observing the regular operation of a circuit and to detect timing errors in the circuit itself during operation. With the error information, the critical supply voltage, that is the minimum supply voltage which is needed for correct operation, can be determined. The second possibility is to observe an over-critical system. Here, a test module which is always slower than the most critical part of the chip is observed, and as soon as the test module fails, the system detects a late data transition called a pre-error[23]. The regular in-situ method achieves 14% power reduction using two power switches, while imposing 10% area overhead and 0.5% power overhead to the system. The over-critical method compared to the worst-case design reduces power to 13.5% with a negligible error rate of 10^{-15}.

Another approach implements replica-paths, representing the critical paths of the circuit, thus, with varying operating conditions, the timing of the replica-path will change similarly to the actual critical path. So, the timing information of the replica-path can be used to control the supply voltage adaptively. Alternatively, the critical path replica can be replaced by fan-out of 4 (FO4) ring oscillator [34] or a delay line [35]. A safety margin is added to account for any mismatch between the ring oscillator (or the delay line) and the actual critical path. The FO4 technique achieves 11% power reduction for compute-intensive code; the power decreased by up to 78 % for non-speed-critical applications compared to operating at a fixed supply voltage.

Several methods have been proposed which implement one monitor per variation source. For instance, the method presented in [36] synthesizes a single representative critical path (RCP) for post-silicon delay prediction. The RCP is designed such that it is highly correlated to all critical paths for some expected process variations. For both this and the critical path replica method, it is essential to guard-band the prediction. However, the RCP approach with 2.8% predication error rate requires a guard-band 31% smaller than the critical path replica method.

Although design-dependent monitors show good estimation accuracies, most of them rely on monitoring and characterization of one unique critical path, however, due to the increasing effect of process variations, finding one unique critical path is a hard task to do. Depending on the operating point, process corner, and workload many different timing paths might become critical, therefore, for real circuits the concept of finding one critical path and create a critical path replica as a performance monitor is too simplistic. Moreover, techniques that have one monitor per critical path incur high area overhead as well as long design turnaround time to the system[24].

4.2 Global power management unit

We discussed various types of performance monitors used for AVS to locally manage power within each core. All the mentioned types of performance monitors are applicable for both single as well as multicore processors. As we discussed earlier in this section, process variations are static during operation and manifest themselves as D2D, WID variations, while temperature and voltage variations are dynamic. These affect both single as well as multicore processors. However, as the individual core size becomes smaller, there arises another source of process variations that specifically affects multicore systems, called core-to-core variations (C2C). C2C variations occur due to spatially correlated WID

variations, for example due to non-uniformity in the lithographic exposure field [38]. Thus, multicore processors are still threatened by increasing power consumption due to PVT variations since they require large design margins in the supply voltage resulting in large power consumption.

Dynamic power management of multicore processors is extremely important because it allows power savings when not all cores are used. AVS is one of the techniques that is widely used for power reduction in multicore processors. The per-core performance data collected by performance monitors is sent to the global power management unit to decide the supply voltage. AVS for multicore processors can be performed at various levels of granularity: 1) Per-chip, the supply voltage is set globally for the whole chip, 2) Per-core, the supply voltage is set for each core, which means that only cores that require higher frequency are set to the higher supply voltage, while other cores operate at lower supply voltage or are completely shut down, 3) cluster-level, the voltage is set for each cluster which one or more cores are associated with.

5. CONCLUSION

This paper presented a classification of power reduction techniques in single and multicore systems. Three main classes have been discussed: the techniques which aim to reduce power either during fabrication/design or runtime in the tiles, run-time power reduction techniques for interconnects, and adaptive voltage scaling techniques to dynamically manage power during run-time. In addition, a number of design and manufacturing issues (such as process and temperature variations) have been taken into consideration. The paper also discussed a number of examples for each of the classes and presented the published power reduction numbers reported by their respective papers. A summary of these numbers has been listed along with the trade-offs in performance and/or area overhead incurred as a result.

Acknowledgments

This work is carried out under the BENEFIC project (CA505), a project labelled within the framework of CATRENE, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics.

6. REFERENCES

[1] Y. B. Kim, *Challenges for Nanoscale MOSFETs and Emerging Nanoelectronics*, Trans. On Electrical and Electronic Materials, vol. 11, pp. 93-105,2010.

[2] S. H. Fuller and L. I. Millett, *The Future of Computing Performance: Game Over or Next Level?*, The National Academy of Sciences, 2011.

[3] L. Spracklen and S. G. Abraham, *Chip Multithreading: Opportunities and Challenges*, in HPCA, pp. 248-252, 2005.

[4] H. Esmaeilzadeh, et. al, *Dark Silicon and the End of Multicore Scaling*, in ISCA, vol. 46, pp. 5âĂŞ26, 2011.

[5] V. Venkatachalam and M. Franz, *Power Reduction Techniques for Microprocessor Systems*, ACM Computing Surveys, vol. 37, no. 3, pp. 195-237, 2005.

[6] F. Gao and J. P. Hayes, *ILP-based optimization of sequential circuits for low power*, in ISLPED, pp. 140-145, 2003.

[7] K. Ghose, B. Kamble, *Reducing power in superscalar processor caches using subbanking, multiple line buffers and bit-line segmentation*, in ISLPED, Pages 70-75, 1999.

[8] J. Hu, et. al, *Using Dynamic Branch Behavior for Power-efficient Instruction Fetch*, in ISVLSI, pp. 127-132, 2003.

[9] S. N. Kim, et. al, *Drowsy Instruction Caches: Leakage Power Reduction Using Dynamic Voltage Scaling and Cache Sub-bank Prediction*, in MICRO, pp. 219-230, 2002.

[10] A. Buyuktosunoglu, et. al, *An Adaptive Issue Queue for Reduced Power at High Performance*, Proc. of the Inter. Workshop on Power-aware Computer Systems, pp. 25-39, 2001.

[11] J. Ebergen, J. Gainsley, and P. Cunningham, *Transistor sizing: How to control the speed and energy consumption of a circuit*, Proc. of the Inter. Symp. on Asynchronous Circuits and Systems, pp. 51-61, 2004.

[12] H. Kojima, S. Tanaka, and K. Sasaki, *Half-Swing Clocking Scheme for 75% Power Saving in Clocking Circuitry*, in JSSC, vol. 30, no. 4, pp. 432-435, 1995.

[13] E. Kursun, S. Ghiasi, and M. Sarrafzadeh, *Transistor Level Budgeting for Power Optimization*, in ISQED, pp. 116-121, 2004.

[14] A. Sultania, D. Sylvester, and S. Sapatnekar, *Transistor and Pin Reordering for Gate Oxide Leakage Reduction in Dual Tox circuits*, in ICCD, pp. 228-233, 2004.

[15] K. Banerjee and A. Mehrotra, *Global interconnect warming*, IEEE Circuits and Devices Magazine, pp. 16-32, 2001.

[16] M. Stan and W. Burleson, *Bus-invert coding for low-power i/o*, in TVLSI, pp. 49-58, 1995.

[17] H. Zhang, J. Rabaey, *Low-swing interconnect interface circuits*, in ISLPED, pp. 161-166, 1998.

[18] C. N. Taylor, S. Dey, and Y. Zhao, *Modeling and Minimization of Interconnect Energy Dissipation in Nanometer Technologies*, in DAC, pp. 754-757, 2001.

[19] B. Victor and K. Keutzer, *Bus encoding to prevent crosstalk delay*, in ICCAD, pp. 57-63, 2001.

[20] K. N. Patel and I. L. Markov, *Error correction and crosstalk avoidance in dsm busses*, Proc. of ACM Inter. Workshop on System-Level Interconnect Prediction, pp. 9-14, 2003.

[21] W. B. Jone, et. al, *Design theory and implementation for low-power segmented bus systems*, in TODAES, vol. 8, issue 1, pp. 38-54, 2003.

[22] B. Biship, M. .J .Irwin, *Databus Charge Recovery: Practical Considerations*, In ISLPED, pp. 85-87, 1999.

[23] M. Wirnshofer, et. al, *A Variation-Aware Adaptive Voltage Scaling Technique based on In-Situ Delay Monitoring*, in DDECS, pp. 261-266, 2011.

[24] T. Chan, et. al, *DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators*, in ISQED, pp. 633-640, 2012.

[25] T. Chan and A. B. Kahng, *Tunable Sensors for Process-Aware Voltage Scaling*, in ICCAD, pp. 7-14, 2012.

[26] J. Lee, B. G. Nam, and H. J. Yoo, *Dynamic Voltage and Frequency Scaling (DVFS) Scheme for Multi-Domains Power Management*, in ASSCC, pp. 360-363, 2007.

[27] B. Lin, et. al, *User and Process-Driven Dynamic Voltage and Frequency Scaling*, in ISPASS, pp. 11-22, 2009.

[28] M. Elgebaly and M. Sachdev, *Variation-Aware Adaptive Voltage Scaling System*, in TVLSI, vol. 15, no. 5, pp. 560-571, 2007.

[29] T. Yamagishi, et. al, *An Area-Efficient, Standard-Cell Based On-Chip NMOS and PMOS Performance Monitor for Process Variability Compensation*, Proc. of IEEE Inter. Conf. on Cool Chips, pp. 1-3, 2012.

[30] M. Bhushan, et. al, *Ring Oscillators for CMOS Process Tuning and Variability Control*, in TSM, Vol. 19, No. 1, pp. 10-18, 2006.

[31] K. Kang, et. al, *On-Chip Variability Sensor Using Phase-Locked Loop for Detecting and Correcting Parametric Timing Failures*, in TVLSI, vol. 18, no. 2, pp. 270-280, 2010.

[32] M. Eireiner, et. al, *In-Situ Delay Characterization and Local Supply Voltage Adjustment for Compensation of Local Parametric Variations*, in JSSC, vol. 42, no. 7, pp. 1583-1592, 2007.

[33] Y. Ikenaga, et. al, *A 27% Active-Power-Reduced 40-nm CMOS Multimedia SoC with Adaptive Voltage Scaling Using Distributed Universal Delay Lines*, in JSSC, vol. 47, no. 4, pp. 832-840, 2012.

[34] TD. Burd, T. Pering, A. Stratakos, and R. Brodersen, *A dynamic voltage scaled microprocessor system*, in ISSCC, pp. 294 âĂŞ 295, 2000.

[35] J. Kim and M. A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IJSSC, vol. 37, no. 5, pp. 639-647, 2002.

[36] Q. Liu and S. S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.

[37] L. Xie and A. Davoodi, *Representative Path Selection for Post-Silicon Timing Prediction under Variability*, in DAC, pp. 386-391, 2010.

[38] E. Humenay, D. Tarjan, and K. Skadron, *Impact of Process Variations on Multicore Performance Symmetry*, in DATE, pp. 1-6, 2007.

3

AVS TECHNIQUES USING ON-CHIP PERFORMANCE MONITORS

SUMMARY

To overcome the increasing sensitivity to variability in nanoscale integrated circuits, operation parameters (e.g., supply voltage) are adapted in a customized way exclusively to each chip. AVS is a standard industrial technique which has been adopted widely to compensate for process, voltage, and temperature variations as well as power optimization of integrated circuits. For cost and complexity reasons, these techniques are usually implemented by means of performance monitors allowing fast performance evaluation during production or run time. Such on-chip monitoring approaches estimate operation parameters either based on responses from performance monitors with no interaction with the circuit or by monitoring the actual critical paths of the circuit.

In this chapter, we first discuss a number of well-known performance monitoring methodologies and compare them with each other in terms of accuracy, tuning effort, impact on design planning, and implementation risk. This enables evaluating the suitability of various performance monitoring methodologies for specific applications based on their respective requirements in terms of accuracy, power efficiency and cost.

Next, we focus on AVS techniques, which estimate operation parameters using responses from on-chip performance monitors with no interaction with the circuit during production. We discuss the challenges that these monitoring methodologies face with decreasing node sizes, in terms of accuracy and effectiveness. By simulating ISCAS'99 benchmarks using the Nangate 45 nm open cell library, we show that the accuracy of these approaches is design dependent, and requires up to 15% added design margin. In addition, silicon measurements of a nanometric FD-SOI device show that the required design margin is above 10% of the clock cycle, which leads to unacceptable waste of power.

This chapter is based on the following papers.

1. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Industrial Approaches for Performance Evaluation Using On-Chip Monitors*, 11th IEEE International Design Test Symposium (IDT 2016), 18-20 December 2016, Hammamet, Tunisia.

2. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation*, Design, Automation and Test in Europe (DATE 2016), 14-18 March 2016, Dresden, Germany.

Industrial Approaches for Performance Evaluation Using On-Chip Monitors

Mahroo Zandrahimi*, Philippe Debaud†, Armand Castillejo†, Zaid Al-Ars*
*Delft University of Technology, The Netherlands
{m.zandrahimi, z.al-ars}@tudelft.nl
†STMicroelectronics, Grenoble, France
{philippe.debaud, armand.castillejo}@st.com

Abstract—To overcome the increasing sensitivity to variability in nanoscale integrated circuits, operation parameters (e.g., supply voltage) are adapted in a customized way exclusively to each chip. A standard industrial approach to achieve customized circuit adaptations is the use of on-chip monitors that allow fast performance evaluation during production or lifetime. Such on-chip monitoring approaches estimate operation parameters either based on responses from performance monitors with no interaction with the circuit or by monitoring the actual critical paths of the circuit. In this paper, we discuss a number of well-known performance monitoring methodologies and compare them with each other in term of their advantages and disadvantages. This enables evaluating the suitability of various performance monitoring methodologies for specific applications based on their respective requirements in terms of accuracy, power efficiency and cost. In addition, we discuss the challenges that these monitoring methodologies face with decreasing node sizes, in terms of accuracy and effectiveness. By simulating ISCAS'99 benchmarks using the Nangate 45 nm open cell library, we show that the accuracy of these approaches is design dependent, and requires up to 15% added design margin.

I. Introduction

Measurement of operation parameters of integrated circuits can be done either during run-time using online parameter estimation approaches or during production using offline circuit monitoring. Online estimation approaches set the operation parameters for the chip based on the feedbacks they receive from on chip performance monitors. Thus, whenever a change in environmental variations occurs, the system updates the parameter estimation so that all parts of the chip are able to function properly at the target frequency. These approaches are very accurate in estimation and also very efficient in saving power since margins to compensate for environmental variations are measured online. On the other hand, this is rather difficult to implement since the software needs to be manipulated in order to perform online estimation based on the feedbacks received from performance monitors. Furthermore, these techniques are risky for final application since there is a possibility of failure if some parameters are not managed properly.

Offline estimation approaches create a pre-characterized look-up table that links operation parameters to each target frequency. Since parameter estimation for each chip during production should be done as fast as possible, running functional tests on CPU to measure operation parameters for each operating point is not feasible. Moreover, 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 and USB cannot be characterized using this approach. Hence, performance monitors should be embedded in the chip structure. Based on the frequency responses from performance monitors during production, the operation parameters are estimated exclusively for each operating point of each chip. Then, the margins for temperature and voltage variations as well as aging are added on top of the measured parameters to make sure that the chip works even in the worst-case condition. Although these approaches seem very pessimistic and thus not as power efficient as online approaches, they are very much cost effective and easier to implement since no changes in software is needed. Moreover, offline approaches can be seen as an incremental solution for existing devices, which mitigates the risk of the design.

Regardless of using online or offline approaches, performance monitors should be embedded in the chip architecture so that based on the frequency responses, the operation parameters could be estimated. Many process monitors have been proposed for both online and offline monitoring from simple ring oscillators to more complicated design dependent critical path replicas and in-situ delay monitors. In this paper we evaluate the accuracy and effectiveness of using performance monitors for operation parameter estimation. The contributions of this paper are the following:

- An overview of various on-chip performance monitors for online and offline circuit adaptation including a discussion about pros and cons of each approach.
- Investigation of the limitations of on chip performance monitors in terms of accuracy and effectiveness for ISCAS'99 benchmarks using Nangate 45 nm open cell library with different process corners.

The rest of this paper is organized as follows. Section II overviews process monitoring methodologies. Section III gives some recommendations of suitable process monitoring techniques based on design specification. Limitations on process monitoring methodologies are presented in Section IV using simulation results on ISCAS99 benchmarks. Section V concludes the paper and proposes potential solutions for future work.

II. Process monitoring methodologies

Fig. 1 illustrates a taxonomy of process monitoring methodologies based on various monitoring architectures. According to this figure, circuit adaptation is done either using indirect measurement approaches or direct measurement approaches. Indirect measurement approaches estimate operating parameters through correlating frequency responses of performance monitors to the circuit frequency, whereas, direct measurement approaches set the circuit operating parameters by monitoring the actual critical paths of the circuit [9].

978-1-5090-4900-4/16/$31.00 ©2016 IEEE

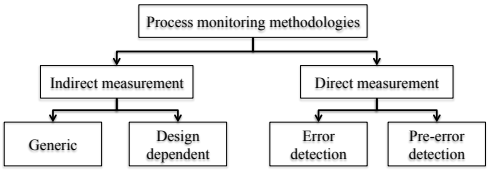

Fig. 1. Classification of process monitoring methodologies

A. Indirect measurement approaches

These approaches embed one or various performance monitors in the chip structure. Due to within-die variations, it is more efficient to place various performance monitors close or inside the block which is being monitored so that all types of process variations are captured and taken into account for parameter adaptation. The number of performance monitors depends on the size of the chip. There is no interaction between performance monitors and the circuit.

To be able to estimate the circuit frequency based on performance monitor responses during production, the correlation between performance monitors and circuit frequency should be measured during characterization, which is an earlier stage of manufacturing [1]. This procedure is done for the amount of test chips representative of the process window to find the correlation between performance monitors and circuit frequencies. Once the performance monitors are tuned to the design during characterization, they are ready to be used for parameter adaptation for each chip during production. Fig. 2 shows an example of a chip with multiple voltage islands, among which performance monitors are distributed. During production, based on the frequency responses from these monitors, the circuit frequency is estimated so that operating parameters can be adapted to each voltage domain of the chip.

Various performance monitoring structures have been proposed from simple generic ring oscillators to more complicated design dependent critical path replicas. The technique presented in [3] implements replica-paths, representing the critical paths of the circuit. Alternatively, the critical path replica can be replaced by fan-out of 4 (FO4) ring oscillator [4] or a delay line [5]. They claim that with varying operating conditions, the timing of monitors will change similarly to the actual critical path. Moreover, the method presented in [6] synthesizes a single representative critical path (RCP) for post-silicon delay prediction. They claim that the RCP is designed such that it is highly correlated to all critical paths for some expected process variations.

However, as the technology scaling enters nanometer regime, specially from 45 nm onwards, finding one unique critical path has become impossible. Depending to the process corner, voltage and temperature variations, and also workload many different timing paths might become critical, therefore, for real circuits the concept of finding one critical path and create a critical path replica as a performance monitor is too simplistic. As a result, regardless of using generic ring oscillators or design dependent replica paths, the characterization phase should be done to find the correlation between monitoring responses and the actual performance of the circuit.

The process monitors, which are widely used today for many products, are ring oscillators designed based on the most used cells extracted from the potential critical paths of the design, reported by static timing analysis. So, based on the design, some standard logic cells are put in an oscillator to form performance monitors, which will be distributed among the chip to capture all kind of variations. During characterization, performance monitors are tuned to the design so that during production, according to the frequency responses of performance monitors, the operation parameters are adapted to each chip.

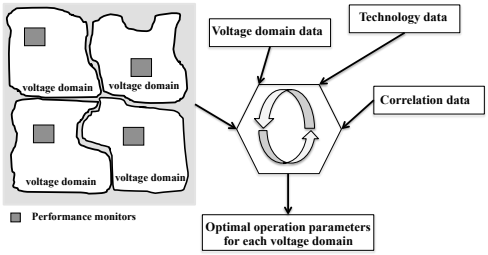

Fig. 2. Operating parameter estimation using indirect measurement approaches

B. Direct measurement approaches

Direct measurement approaches estimate operation parameters by monitoring actual critical paths of the circuit. These approaches add one in-situ delay monitor per critical path. In-situ delay monitors are special latches or flip-flops, included at the end of critical paths to report the timing behavior of the circuit [7]. Circuit delay characterization using in-situ delay monitors can be done in two different ways. The first is by observing the regular operation of a circuit and to detect timing errors in the circuit itself during operation. With the error information, the critical operation parameters, which are needed for correct operation, can be determined. The second possibility is to observe an over-critical system. Here, a test module which is always slower than the most critical part of the chip is observed, and as soon as the test module fails, the system predicts a delayed data transition called a pre-error [8].

For the in-situ monitors, which are able to detect timing errors, error recovery circuits are needed to repeat single computations after malfunction. In contrast, for in-situ approaches which detect pre-errors, no additional hardware effort and complexity for the recovery circuitry is needed, thus, these approaches are easier to manage. Fig. 3 shows an in-situ delay flip flop which detects pre-errors. These in-situ flips flops detect pre-errors when the timing slack in critical paths drops below a certain value. The idea is to reduce the operation parameters as long as no pre-error is detected and to raise the operation parameters as soon as the pre-error rate is above a certain value.

III. WHICH APPROACH SUITS A DESIGN?

In this section we compare indirect measurement versus direct measurement approaches in terms of accuracy, tuning effort, impact on design planning, implementation risk, and area overhead as illustrated in Table I. With regard to accuracy and tuning effort, direct measurement 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 indirect measurement approaches,

TABLE I. COMPARISON OF DIRECT MEASUREMENT VS. INDIRECT MEASUREMENT APPROACHES

Technique	Accuracy	Tuning effort	Impact on design planning	Implementation risk
Direct measurement	high	none	high	medium to high
Indirect measurement	medium	high	low	low

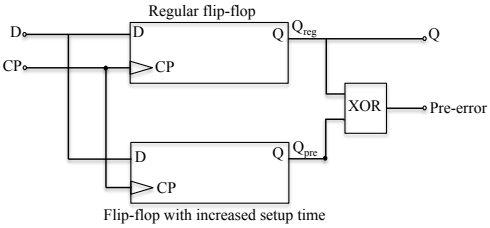

Fig. 3. Structure of in-situ flip-flops which detect pre-errors

since there is no interaction between performance monitors and the circuit, the correlation between performance monitor 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 performance monitors from different test chips, the estimated correlation between the frequency of performance monitors 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, direct measurement approaches always win.

To validate our claim of low accuracy of indirect measurement approaches, we have done silicon measurement on 625 devices manufactured using nanometric FD-SOI technology [10]. 12 performance monitors (PM) are embedded in each device. First, we have measured the real value of optimal voltage (Vmin) for each chip using test patterns. Then, we set an arbitrary voltage for each chip and collected frequency responses from all 12 performance monitors. Finally, we mapped each frequency response of a PM to the Vmin of the chip in which that PM is located. Fig. 4 shows an example of such a plot for one specific PM on all 625 devices measured. To quantify the amount of this discrepancy in this figure, for each value of frequency response, we have looked for the Vmin variation. We take the maximum amount of this variation as the Vmin discrepancy for that PM. We measured the amount of Vmin discrepancy for all 12 monitors, the result of which is presented in Fig. 5. This figure also presents the wasted power as a results of inaccuracy in Vmin estimation using performance monitors. Results show that minimum voltage estimation based on performance monitors lead to nearly 10% of wasted power on average and 7.6% in the best case, when a single PM is used for performance estimation.

In terms of planning effort and implementation risk, direct measurement 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, indirect measurement approaches are considered more acceptable in terms of planning and implementation risk, since there is no interaction between performance monitors and the circuit, hence, performance monitors can even be placed outside

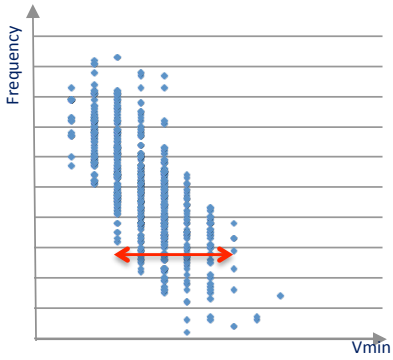

Fig. 4. Example of Vmin discrepancy for one PM on all 625 devices measured

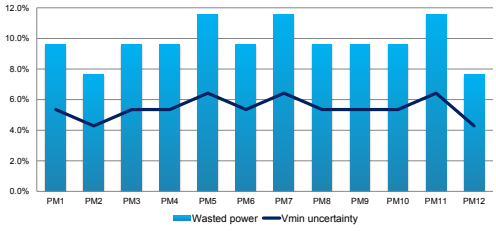

Fig. 5. Inaccuracy in the minimum operating voltages estimated using different performance monitors [10]

the macros being monitored, but not too far due to within die variations. Consequently, indirect measurement 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.

IV. LIMITATIONS OF INDIRECT MEASUREMENT APPROACHES

As we discussed earlier, indirect measurement approaches estimate operation parameters based on responses from performance monitors with no interaction with the circuit. In deep sub-micron technologies, performance monitors are showing limitations to accurately estimate the silicon performance. Within die variations and the amount of parameters that should be taken into account tend to prevent accurate computation of needed optimum operation parameters for a given target frequency. To investigate the variability of critical paths of a design in different corners, first we present an industrial case study regarding critical path variability of a nanometric FD-SOI device through static timing analysis. Next, in order

TABLE II. PERCENTAGE OF CLOCK PERIOD SPENT ON 5000 MOST CRITICAL PATH IN 16 CORNERS

Corner	% of clock period	Corner	% of clock period
1	13.63	9	13.42
2	13.95	10	6.34
3	4.86	11	9.13
4	11.60	12	12.41
5	9.55	13	15.59
6	9.08	14	9.89
7	12.47	15	17.02
8	4.75	16	8.46

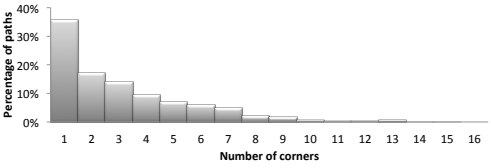

Fig. 6. Percentage of unique paths out of the 5000*16 critical paths present in 1 to 16 corners [10]

to generalize the idea of critical path variability as a result of process and environmental variations, we back up the industrial case study through simulation results on ISCAS'99 benchmarks using Nangate 45 nm open cell libraries.

A. Case study

We have done timing analysis on a nanometric FD-SOI device in sixteen corners with different process and environmental conditions [10]. For each of the sixteen functional corners, we have extracted the 5000 most critical paths of the device. The path lists are sorted from the most critical path to less critical.

In order to understand if five thousand paths are enough for our study, we have computed the distribution of these paths compared to the clock cycle. The objective is to check whether the spread of 5000 paths represents very small part of the clock cycle, which requires to increase the number of paths or is considered enough. For each corner, we have computed paths spread as follows:

$$Spread = (slack_{5000} - slack_1)/(T_{clock}) \quad (1)$$

where $slack_{5000}$ is the slack of the 5000th critical path, $slack_1$ is the slack of the most critical path, and T_{clock} is the clock period. Table II presents the percentage of clock cycle spent on the 5000 most critical path in 16 corners. As it can be seen in this table, depending on the corner, the spread of 5000 paths spans the range from 4.75% to 17% of the clock period, which is considered as enough for our study.

From the sixteen lists of 5000 critical paths, we have extracted the total number of unique paths. We have found 25936 unique paths out of 5000*16. Fig. 6 shows the percentage of the 25936 paths present in 1 or more corners. In this case, only 35.8% of paths are present in 1 corner, and only 53% are present in one or two corners. Two third of the paths are present in maximum 3 corners. None of the paths are present in the list of critical paths of all 16 corners, which means it does not matter which critical path we choose, it does not stay critical even within 5000 most critical paths of all corners.

These results show that identifying a critical path that covers all the corners is not possible. Therefore, when a path is the most critical in a corner, it is important to know how this path is changing across various process, voltage and temperature conditions. Suppose that P_x is the critical path of corner X, P_y is the critical path of corner Y. First, we have computed the distance of the P_x from P_y for all 16 corners against each other in terms of delay. Then, we measured the maximum as well as the average error for each corner if we assume that the critical paths of other corners are the most critical in that corner. Fig. 7 presents average and maximum error measured when the critical path of corner X is used to evaluate performance in corner Y. Results are presented in % of clock period and have been clamped to the value of the 5000th path of the corner Y list. Based on these results, whatever the critical path and the corner we take, maximum error is above 10% of the clock cycle.

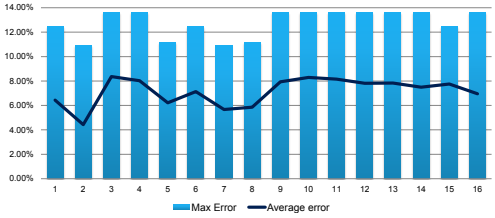

Fig. 7. Performance estimation error using the critical path of another corner [10]

B. Simulation set up

This subsection explains the definition of parameters in order to characterize the simulation results. We use Nangate 45 nm open cell library [11] to investigate critical path variability on ISCAS'99 benchmarks [12] using Cadence RTL Compiler. ISCAS'99 contains 29 designs from small circuits with 21 cells to more complicated designs with almost 44 K cells. Nangate 45 nm library contains 5 different process corners with different characteristics in terms of process and environmental variations. These corners are typical, fast, slow, low temperature (low), and worst low (worst).

In order to characterize the results, we defined a parameter named $error_{max}$ which is measured for each design. If we assume the critical path of each design is the critical path of the typical corner, $error_{max}$ is the maximum percentage of critical path delay change when measured in the other corners. The concept relates to how much margin should be taken into account due to inaccuracies as a result of critical path variability in different corners, if we assume that for each design the critical path remains critical in all process corners. To be able to measure $error_{max}$ for each design, first we check if the critical path in each corner is different from the critical path of the typical corner. In the case of critical path difference, we measure $error_{corner}$ for the process corner by:

$$error_{corner} = (P_{corner} - P_{typ})/P_{corner} \quad (2)$$

where P_{corner} is the delay of the critical path in that corner, and P_{typ} is the delay of the critical path of the typical corner in that corner. Once $error_{corner}$ is measured for all process corners, $error_{max}$ can be obtained for the design by:

$$error_{max} = \max_{all\ corners}[error_{corner}] \quad (3)$$

TABLE III. PERCENTAGE OF $error_{max}$ FOR ISCAS'99 BENCHMARKS USING NANGATE 45 NM LIBRARY

Benchmark	# Cells	$error_{max}$	Benchmark	# Cells	$error_{max}$
$b01$	30	6.93	$b15$	3142	0
$b02$	21	0.10	$b15\_1$	3141	0
$b03$	76	11.65	$b17$	9559	0
$b04$	196	6.29	$b17\_1$	9584	0
$b05$	390	2.85	$b18$	22175	15.03
$b06$	29	1.35	$b18\_1$	22093	0
$b07$	179	0.84	$b19$	43916	9.24
$b08$	71	0	$b19\_1$	43822	0.23
$b09$	94	0.52	$b20$	3970	0.69
$b10$	110	0	$b20\_1$	4025	0.71
$b11$	326	0	$b21$	4022	0.72
$b12$	547	4.19	$b21\_1$	4082	0.71
$b13$	154	0	$b22$	6102	1.12
$b14$	1967	0.69	$b22\_1$	6164	0.74
$b14\_1$	2043	0.67	-	-	-

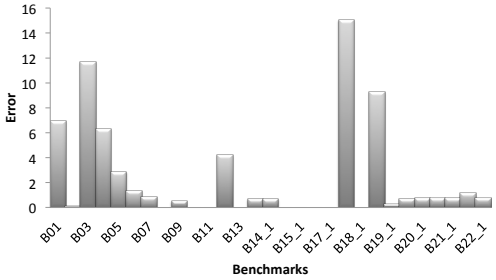

Fig. 8. Percentage of $error_{max}$ for ISCAS'99 benchmarks using Nangate 45 nm library

To further elaborate on how error is measured for each design, here we calculate error for one of the benchmarks ($b03$) with 76 cells. The delay of the critical path of the design in typical corner is 678ps. We name this path as P_{typ}. In the fast corner, P_{typ} is not critical anymore. It drops to the 55th path with the delay of 424ps, while the delay of the critical path of the fast corner (P_{fast}) is 453ps. So, $error_{fast}$ can be measured by:

$$error_{fast} = (453 - 424)/453 = 6.40\% \quad (4)$$

In the slow corner, P_{typ} stays critical, thus $error_{slow}$ equals to zero. For the low temperature corner, P_{typ} drops to the 247th path, and for the worst low corner, P_{typ} drops to the 12th path, hence the errors can be measured in the same way. $error_{low}$ equals to 11.65%, $error_{worst}$ equals to 2.12%. Consequently, $error_{max}$ is obtained by:

$$error_{max} = \max[error_{fast}, error_{slow}, error_{low}, error_{worst}] \quad (5)$$

$$= \max[6.40\%, 0, 11.65\%, 2.12\%] = 11.65\% \quad (6)$$

The $error_{max}$ is measured for all 29 ISCAS'99 benchmarks, the result of which is presented in the next subsection.

C. Simulation results

Fig. 8 illustrates the $error_{max}$ for all 29 ISCAS'99 benchmarks. As shown in this figure, although for some designs the error is zero or negligible, for some other designs the error is rather high and for one case, $b18$, it even reaches 15%. Table III presents the detailed simulation results for all 29 ISCAS'99 benchmarks. According to this table, it is not possible to find a unique critical path for most designs, which stays critical in all 5 corners. Therefore, in order to investigate if the error further can be reduced, we took into account all the paths, which become critical in different corners for performance evaluation.

In order to discover if the error can be reduced for the designs with non-zero $error_{max}$, we estimated delay of each design based on all critical paths in all corners as well as the average critical path delay. To further elaborate, we perform the procedure for benchmark $b01$ as an example. Let $P1$, $P2$, and $P3$ be the paths of $b01$ that become critical in one or more of the 5 process corners. As it can be seen in Table IV, $P1$ is the critical path of the typical and slow corners; $P2$ is the critical path of the fast and low temperature corners; $P3$ is the critical path of the worst low corner. $P1P2P3$ is the average delay of these three critical paths. We let the circuit delay in each corner be the maximum delay of all critical paths ($delay$). We performed a linear least square regression analysis of the correlation between circuit delay and the delay for each critical path as well as the average critical path delay. The 4 regression functions are defined as:

$$est_{P1} = Func1(P1) \quad (7)$$

$$est_{P2} = Func1(P2) \quad (8)$$

$$est_{P3} = Func1(P3) \quad (9)$$

$$est_{P1P2P3} = Func1(P1P2P3) \quad (10)$$

Based on these 4 functions, we computed the delay of the circuit as est_{P1}, est_{P2}, est_{P3}, and est_{P1P2P3}. The estimated delay of $b01$ is defined as the maximum value of 4 estimations in each process corner (column est_{max} in the table).

For the est_{max} values, we calculated the estimation errors ($error_{est}$) as the difference between est_{max} and $delay$, as shown in Table V. According to the table, although we considered all critical paths of $b01$ to estimate the circuit delay, there is still an estimation error of up to 4.5% in delay estimation for different process corners ($error'_{max}$). We performed the same procedure for all benchmarks with non-zero $error_{max}$, the results of which are presented in Table VI. Based on this table, the error can be reduced up to 98.8%, which is for benchmark $b07$. However, although we estimated the design delay considering all critical paths of all corners, there is still some unacceptable error present for some designs such as $b18$. The error of $b18$ is reduced by 47.11%, remained 7.95% out of 15.03%, but still this error is not negligible.

Furthermore, simulation does not fully reflect the actual variations on manufactured silicon. On a physical circuit, other sources of variation, such as within-die variations and IR-drop could promote paths which are not reported as critical by static timing analysis, but will become critical on real silicon. Table VII illustrates paths which are ranked top 9 in one of the corners and the highest ranking of that same path in all other corners. According to this table, a path ranked 1 in one corner, drops above the rank 5000 in one of the other corners. Therefore, for more accurate delay estimation, more paths should be taken into account. The more paths we can cover, the more accurate the delay estimation will be.

We further investigated on the reason of the variability in $error_{max}$ for different designs. Each gate behaves differently when being exposed to process and environmental variations. Thus, corner changes incur a different error value to each design according to the gate structure of the critical path

TABLE IV. DELAY ESTIMATION IN [PS] OF BENCHMARK $b01$ USING CRITICAL PATHS OF ALL CORNERS AS WELL AS THE AVERAGE CRITICAL PATH DELAY

Corner	$P1$	$P2$	$P3$	$P1P2P3$	$delay$	est_{P1}	est_{P2}	est_{P3}	est_{P1P2P3}	est_{max}
Typical	360	356	354	356,66	360	368,26	367,58	367,73	376,21	376.21
Fast	226	238	235	233	238	235,85	233,96	232,87	245,76	245.77
Low	188	202	199	196,33	202	198,30	193,19	192,08	207,09	207.09
Slow	1158	1052	1049	1086,33	1158	1156,76	1155,73	1155,31	1145,86	1156.76
Worst	316	326	326	322,66	326	324,78	333,61	336	340,35	340.35

TABLE V. ERROR ESTIMATION OF BENCHMARK B01 FOR ALL CORNERS

Corner	$error_{est}$
Typical	4.5
Fast	3.26
Low temp	2.52
Slow	0.11
Worst low	4.40
$error'_{max}$	4.50

TABLE VI. ERROR ESTIMATION OF ISCAS'99 BENCHMARKS WITH NON-ZERO $error_{max}$ USING CRITICAL PATH OF ALL CORNERS AS WELL AS THE AVERAGE CRITICAL PATH

Benchmark	$error'_{max}$	reduction	Benchmark	$error'_{max}$	reduction
$b01$	4.50	35.1	$b18$	7.95	47.11
$b03$	6.40	45.1	$b19$	1.94	79.00
$b04$	3.04	51.7	$b19\_1$	0.08	65.22
$b05$	1.83	35.8	$b20$	0.35	49.28
$b06$	1.19	11.8	$b20\_1$	0.48	32.4
$b07$	0.01	98.8	$b21$	0.46	36.1
$b09$	0.35	32.7	$b21\_1$	0.48	32.4
$b12$	1.30	68.9	$b22$	0.46	58.9
$b14$	0.47	31.9	$b22\_1$	0.48	35.14
$b14\_1$	0.47	29.8	-	-	-

of the design. To prove this point, we designed one of the benchmarks, $b03$, using only NAND logic. As it can be seen in Table III, $b03$ is a small design with 76 cells, but the $error_{max}$ is rather high, 11.65%. By designing using only NAND logic, the error dropped to 0. However, since there is no simulated variation of RC delay in different process corners of Nangate 45 nm library, in actual circuits, a small error might be present in this case as well.

V. CONCLUSIONS AND FUTURE WORK

For some products such as nomadic applications, cost and design customization effort are considered significant. Despite the accuracy and effectiveness of direct measurement performance monitoring approaches, cost versus benefit is not proven since the implementation risk and the impact on design planning is high. Thus, indirect measurement performance monitoring approaches are considered more manageable for many low cost products. However, in deep sub-micron technologies, indirect measurement approaches are showing limitations to accurately estimate silicon performance, which leads to unnecessary power loss. Based on simulation results on ISCAS'99 benchmarks as well as static timing analysis of a nanometric FD-SOI device, we showed that depending on the design, critical path can change dramatically as a result of PVT variations. Thus, the accuracy and effectiveness of indirect measurement approaches is low.

Our future work will concentrate on solutions to avoid these limitations. One possible solution could be using delay test patterns for delay estimation of a design. The main challenge of using test patterns for delay estimation is that there should be a reasonable correlation between delay test patterns and functional test patterns. Test time should also be reasonable compared to the indirect measurement approaches which are very fast during production.

TABLE VII. TOP 9 CRITICAL PATH RANKING OF B14 IN DIFFERENT CORNERS

Least rank	Highest rank	Least rank	Highest rank
1	297	5	869
1	>5000	5	18
1	26	6	862
2	646	7	1165
2	56	8	902
3	496	8	21
3	71	8	34
4	2493	9	423
4	3967	9	4902
4	27	9	47
5	1429	-	-

ACKNOWLEDGEMENTS

This work is carried out under the BENEFIC project (CA505), a project labelled within the framework of CATRENE, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics.

REFERENCES

[1] T. Chan and A.B. Kahng, *Tunable Sensors for Process-Aware Voltage Scaling*, in ICCAD, pp. 7-14, 2012.
[2] T. Chan, et al., *DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators*, in ISQED, pp. 633-640, 2012.
[3] A. Drake, et al., *A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor*, in ISSCC, pp. 398-399, 2007.
[4] TD. Burd, et al., *A dynamic voltage scaled microprocessor system*, in ISSCC, pp. 294-295, 2000.
[5] J. Kim and M.A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IJSSC, vol. 37, no. 5, pp. 639-647, 2002.
[6] Q. Liu and S.S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.
[7] M. Wirnshofer, et al., *A Variation-Aware Adaptive Voltage Scaling Technique based on In-Situ Delay Monitoring*, in DDECS, pp. 261-266, 2011.
[8] M. Eireiner, et al., *In-Situ Delay Characterization and Local Supply Voltage Adjustment for Compensation of Local Parametric Variations*, in IJSSC, vol. 42, no. 7, pp. 1583-1592, 2007.
[9] M. Zandrahimi and Z. Al-Ars, *A Survey on Low-power Techniques for Single and Multicore Systems*, in ICCASA, pp. 69-74, 2014.
[10] M. Zandrahimi, et al., *Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation*, in DATE, 2016.
[11] http://www.nangate.com
[12] http://www.cad.polito.it/downloads/tools/itc99.html

Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation

Mahroo Zandrahimi*, Zaid Al-Ars*, Philippe Debaud†, Armand Castillejo†
*Delft University of Technology, The Netherlands
{m.zandrahimi, z.al-ars}@tudelft.nl
†STMicroelectronics, Grenoble, France
{philippe.debaud, armand.castillejo}@st.com

Abstract— Circuit monitoring techniques have been adopted widely to compensate for process, voltage, and temperature variations as well as power optimization of integrated circuits. For cost and complexity reasons, these techniques are usually implemented by means of performance monitors allowing fast performance evaluation during production. In this paper, we demonstrate the limitations of performance monitoring methodologies in terms of accuracy and effectiveness. Silicon measurements of a nanometric FD-SOI device show that the required design margin is above 10% of the clock cycle, which leads to unacceptable waste of power.

I. Introduction

As technology scales, circuit performance becomes extremely sensitive to process, voltage, and temperature variations (PVT). Furthermore, over time, circuit performance degrades due to different wear out mechanisms, such as NBTI, HCI, etc. One possible solution to make sure that the circuit works properly during its lifetime despite these sources of variation is to adapt operation parameters, e.g., supply voltage, exclusively to each chip. Measurement of operation parameters is done using various circuit monitoring techniques, which embed one or more performance monitors on chip. Thus, operation parameters can be measured through correlating frequency responses of performance monitors to the circuit frequency.

Various performance monitoring structures have been proposed from simple generic ring oscillators to more complicated design dependent critical path replicas. The technique presented in [1] implements replica-paths, representing the critical paths of the circuit. Alternatively, the critical path replica can be replaced by fan-out of 4 (FO4) ring oscillator [2] or a delay line [3]. The method presented in [4] synthesizes a single representative critical path (RCP) for post-silicon delay prediction. The paper suggests that the RCP is designed such that it is highly correlated to all critical paths for some expected process variations. However, as technology scaling enters the nanometer regime, specially from 45 nm onwards, finding one unique critical path has become impossible. Depending to the process corner, voltage and temperature variations, and also workload many different timing paths might become critical, therefore, for real circuits the concept of finding one critical path and create a critical path replica as a performance monitor is too simplistic.

In this paper, we evaluate the accuracy and effectiveness of using performance monitors for operation parameter estimation. The rest of this paper is organized as follows. Section II discusses how circuit monitoring techniques work. Section III shows the limitations of performance monitors, first in terms of accuracy using static timing analysis of a nanometric FD-SOI device, and then in terms of power using silicon measurements of the same FD-SOI device. Section IV concludes the paper.

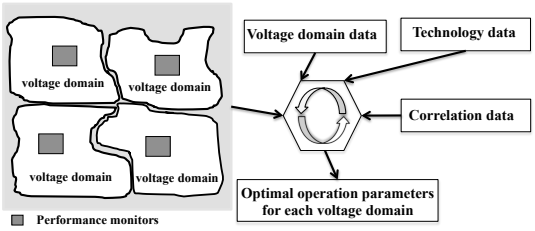

Fig. 1. Operation parameter estimation using monitoring approaches

II. Circuit monitoring techniques

Circuit monitoring techniques are widely used for adapting operation parameters exclusively to each chip for PVT variation compensation as well as power optimization [5]. These techniques embed one or various performance monitors on chip. Using the responses from these performance monitors, operation parameters are measured.

Fig. 1 shows an example of a chip with multiple voltage islands, among which performance monitors are distributed. There is no interaction between performance monitors and the circuit. To be able to estimate the circuit frequency based on performance monitor responses during production, the correlation between performance monitors and circuit frequency should be measured during characterization, which is an earlier stage of manufacturing. This correlation procedure is done for a number of test chips representative of the process window. During production, based on the frequency responses from these monitors, the circuit frequency is estimated so that operation parameters can be adapted to each voltage domain of the chip.

III. Evaluation of performance monitors

A. Accuracy evaluation

To investigate the accuracy of circuit monitoring techniques, we present some industrial experiences regarding critical path variability of a nanometric FD-SOI device through static timing analysis in sixteen corners having different process and environmental conditions. For each of the sixteen functional corners, we have extracted the 5000 most critical paths of the device. The path lists are sorted from the most

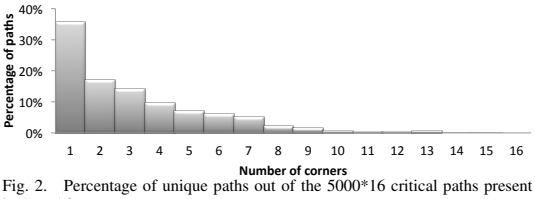

Fig. 2. Percentage of unique paths out of the 5000*16 critical paths present in 1 to 16 corners

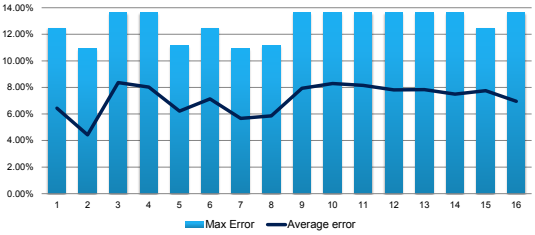

Fig. 3. Performance estimation error using the critical path of another corner

critical path to the least critical.

From the sixteen lists of 5000 critical paths, we have extracted the total number of unique paths. We have found 25936 unique paths out of 5000*16. Fig. 2 shows the percentage of the 25936 paths present in 1 or more corners. In this case, only 35.8% of paths are present in 1 corner, and only 53% are present in one or two corners. Two third of the paths are present in maximum 3 corners. None of the paths are present in the list of critical paths of all 16 corners, which means it does not matter which critical path we choose, it does not stay critical even within 5000 most critical paths of all corners.

These results show that identifying a critical path that covers all the corners is not possible. Therefore, when a path is the most critical in a corner, it is important to know how this path is changing across various process, voltage and temperature conditions. Suppose that P_x is the critical path of corner X, P_y is the critical path of corner Y. First, we have computed the distance of the P_x from P_y for all 16 corners against each other in terms of delay. Then, we measured the maximum as well as the average error for each corner if we assume that the critical paths of other corners are the most critical in that corner. Fig. 3 presents average and maximum error measured when the critical path of corner X is used to evaluate performance in corner Y. Results are presented in % of clock period and have been clamped to the value of the 5000th path of the corner Y list. Based on these results, whatever the critical path and the corner we take, the maximum error is above 10% of the clock cycle. As a result, regardless of using generic ring oscillators or design dependent replica paths, the characterization phase should be done to find the correlation between monitoring responses and the actual performance of the circuit.

B. Power evaluation

The process monitors, which are widely used today for many products, are ring oscillators designed based on the most used cells extracted from the potential critical paths of the design, reported by static timing analysis. So, based on the design, some standard logic cells are put in an oscillator to form performance monitors, which will be distributed among the chip to capture all kind of variations. During characterization, performance monitors are tuned to the design so that during production, according to the frequency responses of performance monitors, the operation parameters are adapted to each chip.

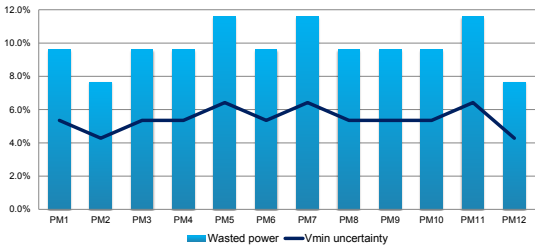

Fig. 4. Voltage discrepancy and power loss using different performance monitors

We have done silicon measurement on 625 devices manufactured using nanometric FD-SOI technology on the same circuit as in SectionIII-A. 12 performance monitors (PMs) are embedded in each device. First, we have measured the real value of optimal voltage (Vmin) for each chip using test patterns. Then, we set an arbitrary voltage for each chip and collected frequency responses from all 12 performance monitors. Finally, we mapped each frequency response of a PM to the Vmin of the chip in which that PM is located. Results show variations of a PM frequency. We take the maximum amount of this variation as the Vmin discrepancy for that PM. We measured the amount of Vmin discrepancy for all 12 monitors, the result of which is presented in Fig. 4. This figure also presents the wasted power as a results of inaccuracy in Vmin estimation using performance monitors. Results show that minimum voltage estimation based on performance monitors lead to nearly 10% of wasted power on average and 7.6% in the best case, when a single PM is used for performance estimation.

IV. Conclusions and Future Work

In deep sub-micron technologies, circuit monitoring approaches are showing limitations to accurately estimate silicon performance, which leads to unnecessary power loss. Based on static timing analysis of a nanometric FD-SOI device, we showed that depending on the design, a critical path can change dramatically as a result of PVT variations. Silicon measurements of the same device show that the required design margin is above 10% of the clock cycle leading to unacceptable waste of power. Thus, new power optimization methods are needed.

Acknowledgements

This work is carried out under the BENEFIC project (CA505), a project labelled within the framework of CATRENE, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics.

References

[1] A. Drake, et al., *A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor*, in ISSCC, pp. 398-399, 2007.

[2] TD. Burd, et al., *A dynamic voltage scaled microprocessor system*, in ISSCC, pp. 294-295, 2000.

[3] J. Kim and M.A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IJSSC, vol. 37, no. 5, pp. 639-647, 2002.

[4] Q. Liu and S.S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.

[5] M. Zandrahimi and Z. Al-Ars, *A Survey on Low-power Techniques for Single and Multicore Systems*, in ICCASA, pp. 69-74, 2014.

4

TF-BASED AVS

SUMMARY
In this chapter we propose an alternative solution for AVS during production using transition fault (TF) test patterns, which is able to eliminate the need for PMBs, while improving the accuracy of performance estimation. The basic requirement of using TF-based AVS is that there should be a reasonable correlation between TF frequency the chip can attain while passing all TF test patterns and the actual frequency of the chip. In this case, TF frequency could be a representative of actual chip performance. We present three different analysis studies to verify that such correlation exists and measure it's characteristics.

1. A case study on real silicon comparing the performance estimation using functional test patterns and the TF-based approach on a 28 nm FD-SOI CPU. The results show a very close correlation between TF test patterns and functional patterns.

2. A case study on real silicon comparing the accuracy of voltage estimation using PMBs and the TF-based approach on a 28 nm FD-SOI device. The results show that the PMB approach can only account for 85% of the uncertainty in voltage measurements, which results in power waste, while the TF-based approach can account for 99% of that uncertainty.

3. Simulation of ISCAS'99 benchmarks using an industrial grade 28 nm FD-SOI library, which includes 42 corners with different characteristics in terms of voltage, body biasing, temperature, transistor speed and aging parameters. The results show that TF-based AVS results in an error as low as 5.33%.

This chapter is based on the following papers.

1. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Using Transition Fault Test Patterns for Cost Effective Offline Performance Estimation*, 12th International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS 2017), 4-6 April 2017, Palma de Mallorca, Spain.

2. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Transition Fault Testing for Offline Adaptive Voltage Scaling*, International Test Conference (ITC 2017), 31 October - 2 November 2017, Fort Worth, USA.

3. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Industrial Evaluation of Transition Fault Testing for Cost Effective Offline Adaptive Voltage Scaling*, Design, Automation and Test in Europe (DATE 2018), 19-23 March 2018, Dresden, Germany.

Using Transition Fault Test Patterns for Cost Effective Offline Performance Estimation

Mahroo Zandrahimi* Philippe Debaud† Armand Castillejo† Zaid Al-Ars*
*Delft University of Technology, The Netherlands
{m.zandrahimi, z.al-ars}@tudelft.nl
†STMicroelectronics, Grenoble, France
{philippe.debaud, armand.castillejo}@st.com

Abstract—Process variation occurring during fabrication of complex VLSI devices induce uncertainties in operation parameters (e.g., supply voltage) to be applied to each device in order for it to fit within the allowed power budget and get the optimum power efficiency. Therefore, an efficient post manufacturing performance estimation mechanism is needed in order to tune operation parameters for each device during production. The current state-of-the-art approach of using Process Monitoring Boxes (PMBs) have shown some limitations in terms of cost and accuracy that limit their benefit. Simulation results on ISCAS'99 benchmarks using 28nm FD-SOI library show that the accuracy of PMB approaches is design dependent, and requires up to 8.20% added design margin. To overcome those limitations, in this paper we propose an alternative solution using transition fault (TF) test patterns, which is able to eliminate the need for PMBs, while improving the accuracy of performance estimation. The paper discusses a case study on real silicon comparing the performance estimation using functional test patterns and the TF based approach on a 28nm FD-SOI CPU. The results show a very close correlation between TF test patterns and functional patterns.

I. Introduction

As technology scales, integrated circuits become more sensitive to process variations. Due to inter die process variations, each chip has its own characteristics which leads to different speed and power consumption. In order to tune each chip during production, a post manufacturing performance estimation mechanism is needed. Since performance estimation during production should be done as fast as possible, running functional patterns on CPU, which reflects the final application is therefore most of the time not feasible. A standard industrial approach for performance estimation is the use of on-chip Performance Monitor Boxes (PMBs), which are very fast during production. They 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 [1]–[6]. 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 performance estimation during production. During production, based on the frequency responses from these monitors, the chip performance will be estimated. According to figure 1, the information could be used to either sort devices based on their speed in order to sell them as a fast or slow device, adapt voltage to enhance yield, or optimize power and battery lifetime, such as voltage scaling and body biasing [7].

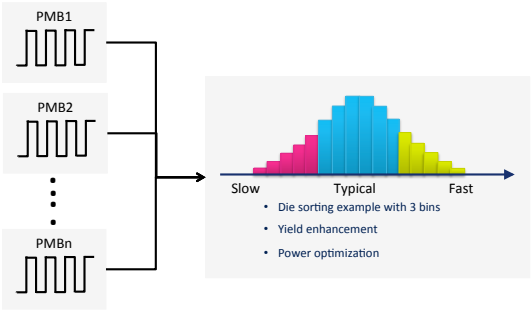

Fig. 1. Performance estimation using PMBs

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 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.

In this paper we introduce a cost effective approach for performance estimation during production using transition fault test patterns, which can be used for general logic as well. The contributions of this paper are the following:

- A detailed investigation of PMB approach in terms of accuracy and effectiveness using 29 ISCAS'99 benchmarks with 28nm FD-SOI library for 42 different process corners.
- Proposing the new concept of using transition fault (TF) testing for performance estimation during production.

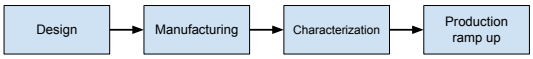

Fig. 2. Stages of the chip design and manufacturing process

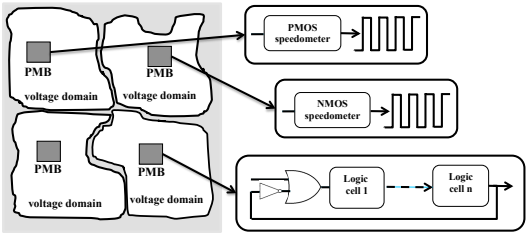

Fig. 3. Performance estimation using PMBs

- A case study on silicon for evaluating the accuracy of performance estimation using TF based approach on a 28nm FD-SOI CPU.

The rest of this paper is organized as follows. Section II introduces the limitations of PMB approaches, which is the reason of investigating new methods for performance estimation during production. Section III proposes the new approach of performance estimation using transition fault test patterns. Evaluation of the proposed approach is presented in Section IV using silicon measurements of a 28nm FD-SOI CPU. Section V concludes the paper and proposes potential solutions for future work.

II. Motivation

Figure 2 shows the various industrial stages of the design and manufacturing process of integrated circuits. The process starts with the design stage, where the circuit structure and functionality is specified based on a given set of specifications. When the design is completed, the manufacturing stage starts where a representative number of chip samples will be manufactured. These chip samples will be used during the characterization stage to find the correlation between PMB responses and the actual performance of the chip. Finally during the production ramp up stage the integrated circuits will be mass produced. In this stage, the PMB correlation measured during the characterization stage will be used to adapt various parameters exclusively to each produced chip.

Figure 3 shows an example of a chip, on which various kinds of PMBs are distributed. The figure shows two PMBs created using PMOS and NMOS speedometers that indicate the speed of PMOS and NMOS transistors. These kind of PMBs are called generic since they can be used for different designs without modifications. The third shown PMB is a critical path replica designed based on the most used logic cells extracted from the potential critical paths of the design, therefore, these kind of PMBs are design dependent. During production based on the frequency responses from these monitors, chip performance is estimated. This information could be used to either sort devices based on their speed (so-called speed binning), adapt voltage to enhance yield, or optimize power and battery lifetime using voltage scaling and body biasing [7].

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. This process is done during the characterization stage. During this stage, functional patterns are executed on each chip, and the frequency of each PMB and the whole chip are measured. These measurements are repeated for a given amount of test chips representative of the process window to make sure that the information from all process corners have been extracted. Based on this information, the correlation between PMBs and the actual frequency of the circuit is determined. Once PMB responses are correlated to application performance, they are ready to be used for performance estimation during production. However, this correlation process has a negative impact in terms of design effort and time to market, since the process should be repeated for a large amount of test chips to make sure that the calculated correlation reflects the actual chip performance for all manufactured chips. The long correlation process makes these approaches very expensive. Moreover, 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 [8].

On the other hand, 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 [9], a silicon measurement on 625 devices manufactured using nanometric FD-SOI technology had been done. 12 PMBs are embedded in each device. Figure 4 shows an example of Vmin discrepancy for one of the 12 PMBs. The Y axis shows the frequency responses of the PMB on all 625 devices, while the X axis shows the optimal voltage of each chip where the corresponding PMB is located. The optimal minimum voltage for each chip is measured using test patterns. To quantify the amount of Vmin discrepancy in this figure, for each value of frequency response, Vmin variation is measured (the red arrow). The maximum amount of this variation is considered as the Vmin discrepancy for that PMB. This inaccuracy in Vmin

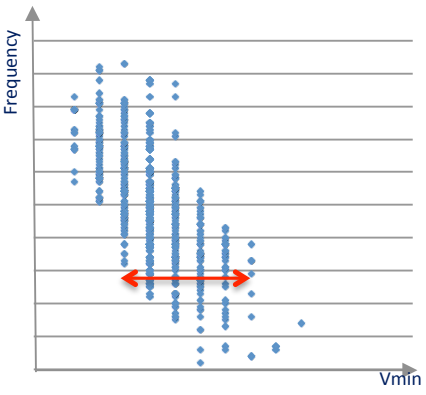

Fig. 4. Example of Vmin discrepancy for one PMB on all 625 devices measured [9]

TABLE I. FEATURES OF DIFFERENT CORNERS OF 28NM FD-SOI LIBRARY USED IN SIMULATIONS

Corner	Voltage [V]	Temperature [°C]	Biasing	Aging	Corner	Voltage [V]	Temperature [°C]	Biasing	Aging
SS	0.7	-40	no	no	SS	0.7	0	no	no
SS	0.7	125	no	no	SS	0.7	-40	no	yes
SS	0.7	0	no	yes	SS	0.7	125	no	yes
SS	0.8	-40	no	no	SS	0.8	0	no	no
SS	0.8	125	no	no	SS	0.8	-40	no	yes
SS	0.8	0	no	yes	SS	0.8	125	no	yes
TT	0.8	25	no	no	TT	0.8	125	no	no
SS	0.85	-40	no	no	SS	0.85	0	no	no
SS	0.85	125	no	no	SS	0.85	-40	no	yes
SS	0.85	0	no	yes	SS	0.85	125	no	yes
TT	**0.85**	**25**	**no**	**no**	TT	0.85	125	no	no
SS	0.9	-40	yes	no	SS	0.9	125	yes	no
SS	0.9	-40	no	no	SS	0.9	0	no	no
SS	0.9	125	no	no	SS	0.9	-40	no	yes
SS	0.9	0	no	yes	SS	0.9	125	no	yes
TT	0.9	125	no	no	TT	0.9	25	no	no
FF	0.9	-40	no	no	FF	0.9	125	no	no
SS	0.95	-40	no	no	SS	0.95	0	no	no
SS	0.95	125	no	no	SS	0.95	-40	no	yes
SS	0.95	0	no	yes	SS	0.95	125	no	yes
TT	0.95	25	no	no	TT	0.95	125	no	no

T and S stand for typical and slow corners, respectively.

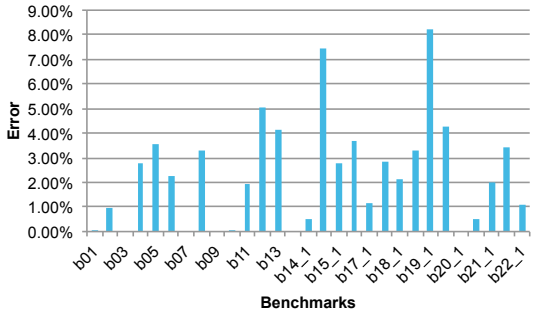

Fig. 5. Percentage of $error$ for ISCAS'99 benchmarks using 28nm FD-SOI library

TABLE II. ERROR IN PERFORMANCE ESTIMATION USING ONE PMB FOR ISCAS'99 BENCHMARKS WITH 28NM FD-SOI LIBRARY

Benchmark	# Cells	$error$	Benchmark	# Cells	$error$
$b01$	30	0.02%	$b15$	3142	7.45%
$b02$	21	0.96%	$b15\_1$	3141	2.77%
$b03$	76	0.00%	$b17$	9559	3.67%
$b04$	196	2.80%	$b17\_1$	9584	1.14%
$b05$	390	3.53%	$b18$	22175	2.86%
$b06$	29	2.27%	$b18\_1$	22093	2.14%
$b07$	179	0.00%	$b19$	43916	3.31%
$b08$	71	3.31%	$b19\_1$	43822	8.20%
$b09$	94	0.00%	$b20$	3970	4.25%
$b10$	110	0.07%	$b20\_1$	4025	0.00%
$b11$	326	1.96%	$b21$	4022	0.48%
$b12$	547	5.04%	$b21\_1$	4082	2.02%
$b13$	154	4.12%	$b22$	6102	3.45%
$b14$	1967	0.00%	$b22\_1$	6164	1.08%
$b14\_1$	2043	0.49%	-	-	-

measurement results in wasting power. The same procedure is done for all 12 PMBs, and the results show that minimum 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.

To further investigate the accuracy and effectiveness of PMB approaches, we performed static timing analysis (STA) with Primetime (SYNOPSYS tool for STA [15]) on ISCAS'99 benchmarks [14] using 28nm FD-SOI library. ISCAS'99 contains 29 benchmarks from small circuits with 21 cells to more complicated benchmarks with almost 44K cells. Table I lists the characteristics of the 42 different corners used in the STA simulation for the 28nm FD-SOI library with voltage, body biasing, temperature, transistor speed and aging parameters.

The results of the simulation are expressed in terms of the performance error in the PMB estimation. We assume that the PMB performance estimation for each benchmark is represented by the critical path reported by STA in the typical corner for that benchmark. The characteristics of the typical corner simulation are (TT, 0.85, 25, no, no), as highlighted as the bold row in Table I. Then, we estimate the performance of the design in the 41 other corners using that PMB (represented by the typical corner simulation). In order to quantify the results, we define a parameter named $error$ which is measured for each benchmark. The concept relates to how much margin should be taken into account due to inaccuracies as a result of performance estimation using PMBs. To be able to measure $error$ for each benchmark, first we check if the critical path in each corner is different from the critical path of the typical corner (PMB for each benchmark). In the case of critical path difference, we measure $error_{corner}$ for the process corner by:

$$error_{corner} = (P_{corner} - \text{PMB})/P_{corner} \quad (1)$$

where P_{corner} is the delay of the critical path measured in $corner$, and PMB is the delay of the critical path identified in the typical corner but measured in $corner$. Once $error_{corner}$ is calculated for all process corners, $error$ can be obtained for each benchmark by:

$$error = \max_{all\ corners}[error_{corner}] \quad (2)$$

Figure 5 illustrates the $error$ for all 29 ISCAS'99 benchmarks. As shown in this figure, although for some designs the error is zero or negligible, for some other designs the error is rather high and for one case, $b19\_1$, it even reaches a maximum of 8.20%. Table II presents the detailed simulation results for all 29 ISCAS'99 benchmarks. According to this table, it is not possible to find a unique critical path for most designs, which stays critical in all 42 corners. Hence, we can conclude that 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 grossly inaccurate. 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 submicron technologies, as intra-die variations and interconnect capacitances are becoming predominant, it is more complex to estimate the performance of the whole design based on one or a couple of 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.

III. TF BASED PERFORMANCE ESTIMATION

In this paper, we propose an innovative new approach for performance estimation using delay testing during production. Since delay testing covers many path-segments of the design, it can be a better performance representative than a PMB. Such an approach has a number of unique advantages as compared to PMB-based approaches.

1) First, this approach can be performed *at no extra cost*, 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 Figure 2), 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.

There are three different types of delay test patterns: TF tests, small delay defect tests, and path delay tests [10]. 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. However, ATPG choices are based on heuristics like SCOAP [11], which tend to minimize computational effort. Thus, when several solutions are available for path sensitization, ATPG will use the easiest, which means that the tool tends to target short paths and not critical paths of the design [12]. On the other hand, we can alternatively use small delay defect testing, which sensitizes paths with smallest slacks, as well as path delay testing, which sensitizes a selected path. Among these two delay testing methods, path delay seems more promising since it sensitizes functional, long paths, which is an advantage over TF testing. However, in path delay testing the objective is to obtain a transition along critical paths which are on average longer and more complex than the paths targeted in transition fault, thus reducing parallel testing capability and thereby reduces the overall coverage achieved. Therefore, we target TF test patterns in this paper for performance estimation during production since these give the highest path coverage of the three delay test alternatives.

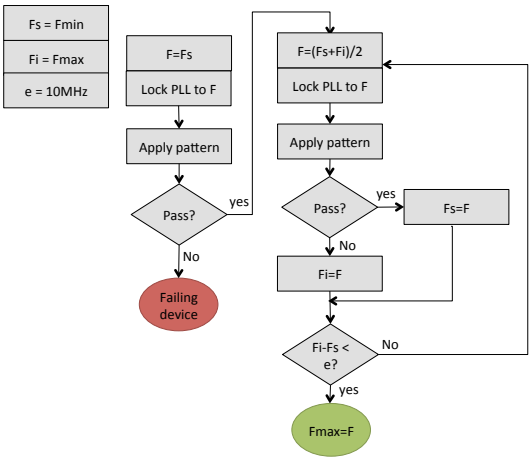

Fig. 6. Proposed flow for performance estimation using TF test patterns

Figure 6 proposes a flow of the TF based approach that could be used during production. The proposed flow performs a binary search to identify the maximum frequency (Fmax) the chip can attain while passing all TF test patterns. The following steps are performed for each operation point of the chip: 1. apply chip setup at nominal values and initialize variables, 2. set PLL to Fmin and wait for stabilization, 3. apply transition fault at speed test, 4. if the chip fails the test, discard it, otherwise, 5. compute new values and do a binary search to find Fmax. Conversion from Fmax to Vmin might be required depending on either performance estimation is done for yield enhancement or power optimization. ”e” is an arbitrary value which is up to the users to define the resolution they want.

IV. EVALUATION RESULTS

The basic requirement of using TF-based AVS is that there should be a reasonable correlation between TF frequency the chip can attain while passing all TF test patterns and the actual frequency of the chip. In this case, TF frequency could be a representative of actual chip performance. In order to investigate if such correlation exits, we performed measurements on-silicon using both TF test patterns and functional patterns. Since running functional patterns on CPU reflects the final application, and thus the actual performance of the chip, we used functional frequency as a reference for comparison versus TF frequency. 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.

The device under test is a high speed 28nm FD-SOI CPU. This device is equipped with an Adaptive Voltage Scaling system (AVS), which means whenever maximum performance is not required, supply voltage can be scaled so that power can be saved while the system can still meet the timing constraints. Therefore, during production, the optimal voltage should be measured for each frequency point of the chip. We have performed the following steps to compare TF frequency

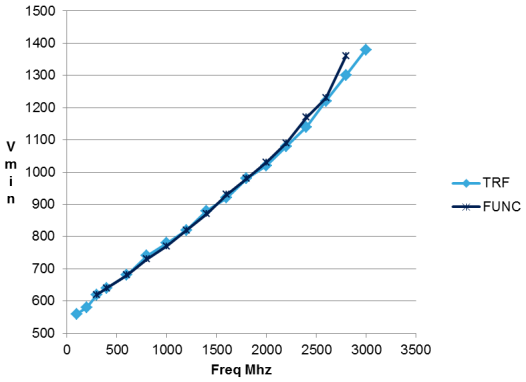

Fig. 7. Correlation of TF and functional patterns on a 28nm FD-SOI CPU

versus functional frequency, which reflects the actual frequency of the chip:

1) We first performed functional test patterns on CPU, and measured the optimal voltage for each frequency point of the chip.
2) Then we have done the same flow discussed in Figure 6, which performs a binary search to identify the minimum voltage (Vmin), at which the chip can pass all TF test patterns for each operating point.

Results are shown in Figure 7. In this figure, the light blue line represents the minimum voltage (y-axis) for each operating point (x-axis) estimated using TF test patterns. The dark blue line represents the minimum voltage (y-axis) measured for each frequency settings (x-axis) of the chip using functional patterns. According to this figure, there is a very close correlation between TF test patterns and functional patterns, which indicates that TF frequency is a very accurate indicator of performance, and therefore can be used for performance estimation during production as an alternative for PMB approach.

V. Conclusions and Future Work

Process variation occurring in deep sub-micron technologies limit PMB effectiveness in silicon performance estimation leading to unnecessary power and yield loss. Simulation results on ISCAS'99 benchmarks using 28nm FD-SOI library show that the accuracy of PMB approaches is design dependent, and requires up to 8.20% added design margin. Thus, we can conclude that 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 efforts have a negative impact in terms of cost and time to market. Finally the fact that functional patterns are used for the correlation process makes PMB approaches not suitable for general logic.

Alternatively, this paper proposes a new approach that uses transition fault testing for performance estimation during production. Since transition fault test patterns target all gates and indirectly cover all path-segments, it can be a better performance representative than a PMB. This approach can be performed at no extra cost, remove the expensive correlation phase and reduces time to market dramatically. Moreover, as functional patterns are not used anymore, testing approach could be a solution for general logic, not only for CPU and GPU. Based on silicon measurements on a high speed 28 nm FD-SOI CPU, there is a very close correlation between TF test patterns and functional patterns proving the relevancy of the TF based approach.

Acknowledgements

This work is carried out under the BENEFIC project (CA505), a project labelled within the framework of CATRENE, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics.

References

[1] T. Chan and A.B. Kahng, *Tunable Sensors for Process-Aware Voltage Scaling*, in ICCAD, pp. 7-14, 2012.

[2] T. Chan, et al., *DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators*, in ISQED, pp. 633-640, 2012.

[3] A. Drake, et al., *A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor*, in ISSCC, pp. 398-399, 2007.

[4] TD. Burd, et al., *A dynamic voltage scaled microprocessor system*, in ISSCC, pp. 294-295, 2000.

[5] J. Kim and M.A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IJSSC, vol. 37, no. 5, pp. 639-647, 2002.

[6] Q. Liu and S.S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.

[7] M. Zandrahimi and Z. Al-Ars, *A Survey on Low-power Techniques for Single and Multicore Systems*, in ICCASA, pp. 69-74, 2014.

[8] M. Zandrahimi, Z. Al-Ars, P. Debaud, and A. Castillejo, *Industrial Approaches for Performance Evaluation Using On-Chip Monitors*, in IDT, 2016.

[9] M. Zandrahimi, Z. Al-Ars, P. Debaud, and A. Castillejo, *Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation*, in DATE, 2016.

[10] M. Tehranipoor et al., Test and Diagnosis for Small-Delay Defects, in Springer Science+Business Media, LLC, 2011.

[11] L.H. Goldstein, E.L. Thigpen, SCOAP: Sandia Controllability/Observability Analysis Program, in DAC, 1980.

[12] B. Kruseman, A. Majhi, and G. Gronthoud, On Performance Testing with Path Delay Patterns, in VTS, 2007.

[13] http://www.st.com/content/st_com/en/about/innovation—technology/FD-SOI.html

[14] http://www.cad.polito.it/downloads/tools/itc99.html

[15] http://www.synopsys.com/tools/pages/default.aspx

Transition Fault Testing for Offline Adaptive Voltage Scaling

Mahroo Zandrahimi*, Philippe Debaud+, Armand Castillejo+ and Zaid Al-Ars*
Delft University of Technology, The Netherlands*, STMicroelectronics, Grenoble, France+

Abstract

In this paper, we propose using transition fault test patterns to perform adaptive voltage scaling (AVS) as a low-cost alternative to process monitoring boxes (PMBs) while improving accuracy of voltage estimation. The paper discusses a case study on real silicon comparing the accuracy of voltage estimation using PMBs and the TF-based approach on a 28nm FD-SOI device. The results show that the PMB approach can only account for 85% of the variability in the measurements, while the TF-based approach can account for 99% of that variability.

1. Introduction

AVS has been used widely to compensate for process, voltage, and temperature variations as well as power optimization of integrated circuits. The current industrial state-of-the-art AVS approaches embeds several PMBs on chip 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 [1,2]. PMBs have shown some limitations in terms of cost and accuracy that limit their benefit [3]. This paper proposes using TF testing during production as an alternative approach that is both cheaper and more accurate. Since transition fault testing covers many path-segments of the design [4], it can be a better performance representative than a PMB.

Here, we propose a flow of the TF-based AVS approach that could be used during production. The proposed flow performs a binary search to identify the minimum voltage (Vmin), at which the chip can pass all TF test patterns. The following steps are performed for each operation point of the chip: 1) Apply chip setup at nominal values and initialize variables; 2) Set supply voltage to Vmax and wait for stabilization; 3) Apply transition fault at speed test; 4) If the chip fails the test, discard it, otherwise; 5) compute new values and do a binary search to find Vmin.

2. Industrial case study

In this section, we compare PMB versus TF for AVS during production using measurements on real silicon. Our case study is a 28nm FD-SOI device on which a number of PMBs are distributed. During the characterization phase of chip production, the correlation between frequency of PMBs and the actual frequency of the device is measured for a number of chips representative of the process window so that during production, and according to the frequency responses of PMBs, optimal voltage estimation is done for each chip. Alternatively, voltage estimation can be done using transition fault testing during production as well at no extra costs. Also, since transition fault testing represents a direct measurement of chip performance, the expensive correlation during the characterization phase is not needed anymore, which reduces time to market dramatically.

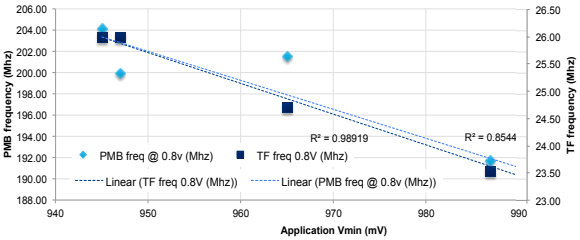

Figure 1. Application Vmin versus TF and PMB results

We have done silicon measurement on 5 chip samples. First, we have measured the real value of optimal voltage (Vmin) for each chip operating at its nominal frequency using functional patterns. To understand whether PMB or TF is more accurate for performance prediction, we have to identify which of them is more correlated with application Vmin. Therefore, we mapped both frequency response of PMB and the TF frequency to the Vmin of the chip in which that PMB is located. Then, we performed a linear least square regression analysis of the correlation between application Vmin and PMB frequency as well as the TF frequency, and measured the coefficient of determination (R^2) for both correlation functions. R^2 is a key output of regression analysis. It is interpreted as the proportion of the variance in the dependent variable (Vmin in this case) that is predictable from the independent variable (PMB frequency and TF frequency). Results are presented in Figure 1. R^2 for the correlation of application Vmin versus PMB is 0.85, while it has a value of 0.99 for the correlation versus TF, which means that Vmin estimation using the PMB approach can only account for 85% of the variability in the measurements, while Vmin estimation using the TF approach can account for 99% of that variability. These results confirm that we can achieve higher accuracy in Vmin estimation using TF.

3. References

[1] M. Zandrahimi and Z. Al-Ars, A Survey on Low-power Techniques for Single and Multicore Systems, in ICCASA, pp. 69-74, 2014.
[2] Q. Liu and S.S. Sapatnekar, Capturing Post-Silicon Variations Using a Representative Critical Path, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.
[3] M. Zandrahimi, Z. Al-Ars, P. Debaud, and A. Castillejo, Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation, in DATE, 2016.
[4] B. Kruseman, A. Majhi, and G. Gronthoud, On Performance Testing with Path Delay Patterns, in VTS, 2007.

Industrial Evaluation of Transition Fault Testing for Cost Effective Offline Adaptive Voltage Scaling

Mahroo Zandrahimi* Philippe Debaud† Armand Castillejo† Zaid Al-Ars*
*Delft University of Technology, The Netherlands
{m.zandrahimi, z.al-ars}@tudelft.nl
†STMicroelectronics, Grenoble, France
{philippe.debaud, armand.castillejo}@st.com

Abstract—Adaptive voltage scaling (AVS) has been used widely to compensate for process, voltage, and temperature variations as well as power optimization of integrated circuits. The current industrial state-of-the-art AVS approaches using Process Monitoring Boxes (PMBs) have shown several limitations such as huge characterization effort, which makes these approaches very expensive, and a low accuracy that results in extra margins, which consequently lead to yield loss and performance limitations. To overcome those limitations, in this paper we propose an alternative solution using transition fault test patterns, 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 AVS using transition fault testing (TF-based AVS) results in an error as low as 5.33%. The paper also shows that the PMB approach can only account for 85% of the uncertainty in voltage measurements, which results in power waste, while the TF-based approach can account for 99% of that uncertainty.

I. 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 [1]. 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 leads 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 [2]–[7].

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 submicron 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 [9] and [10], propose techniques for generating optimal set of delay test patterns during the characterization process, which guarantees 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 to expose systematic timing errors that are likely to affect a large fraction of manufactured chips so it 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. Authors of [8] 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 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. [12] 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. [11] 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 more accurate, cost effective approach for the estimation of AVS voltages during production (post manufacturing) using transition fault test patterns. We focus on test generation for application of AVS during the manufacturing process on every manufactured copy of the chip. Our work optimizes power based on the frequency specification defined at the design stage by setting optimal voltage for each chip to meet performance constraints. The contributions of this paper are the following:

- Proposing the new concept of using transition fault (TF) testing for AVS during production.
- A detailed investigation of the TF-based approach in terms of accuracy and effectiveness using 29 IS-

978-3-9819263-0-9/DATE18/©2018 EDAA

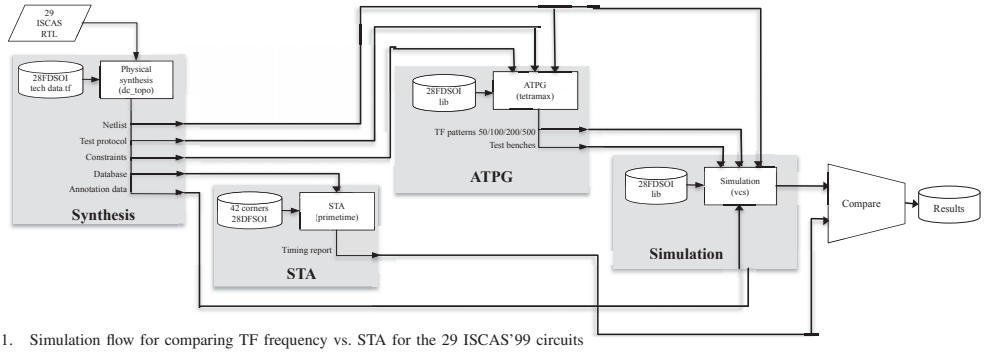

Fig. 1. Simulation flow for comparing TF frequency vs. STA for the 29 ISCAS'99 circuits

CAS'99 benchmarks with 28nm FD-SOI library for 42 different process corners.

This paper is organized as follows. Section II proposes the new approach of using transition fault test patterns for AVS. Evaluation of the proposed approach is presented in Section III using simulation results on ISCAS'99 benchmarks. Section IV presents the results of an industrial study performed on actual 28nm chips to validate the approach. Section V concludes the paper.

II. TF-BASED AVS

In this paper, we propose an innovative new approach for AVS using transition fault testing during production. Since transition fault testing covers many path-segments of the design, it can be a better performance representative than a PMB. Such an approach has a number of unique advantages as compared to PMB-based approaches. First, this approach can be performed *at no extra cost*, since delay tests are routinely performed during production to test for chip functionality. In addition, since delay testing is performed to explicitly test for actual chip performance, the expensive phase of correlating PMS responses to chip performance is not needed anymore, which reduces the length of the characterization stageand subsequently dramatically reduces cost and time to market. 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, and last but not least, this approach makes using PMBs redundant, which saves silicon area as well as PMB design time.

TF testing is one of the three different types of delay test patterns, which includes small delay defect tests and path delay tests, in addition to TF tests [14]. 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. However, automatic test pattern generation (ATPG) algorithms are based on heuristics like SCOAP [15], 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 [16]. On the other hand, we can alternatively use small delay defect testing, which sensitizes paths with smallest slacks, as well as path delay testing, which sensitizes a selected path. Among these two delay testing methods, path delay seems more promising since it sensitizes functional, long paths, which is an advantage over TF testing. However, in path delay 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 transition fault, thus reducing parallel testing capability and thereby reduces the overall coverage achieved. Therefore, we target TF test patterns in this paper for AVS during production since these give the highest path coverage of the three delay test alternatives.

The proposed flow performs a binary search to identify the minimum voltage (Vmin), at which the chip can pass all TF test patterns. The following steps are performed for each operation point of the chip: 1. Apply chip setup at nominal values and initialize variables, 2. Set supply voltage to Vmax and wait for stabilization, 3. Apply transition fault at speed test, 4. If the chip fails the test, discard it, otherwise, 5. compute new values and do a binary search to find Vmin. Conversion from Vmin to Fmax might be required depending on either performance estimation is done for yield enhancement or power optimization. "e" is an arbitrary value which is up to the users to define the resolution they want.

The basic requirement of using TF-based AVS is that there should be a reasonable correlation between TF frequency the chip can attain while passing all TF test patterns and the actual frequency of the chip. In this case, TF frequency could be a representative of actual chip performance. To investigate if this correlation exists, we perform simulations on ISCAS'99 benchmarks which contain 29 designs with different characteristics.

III. EVALUATION RESULTS

A. Simulation setup

This subsection explains the flow we used to explore if TF tests correlate with the actual frequency of the circuits. We use 28nm FD-SOI libraries to compare the transition fault maximum frequency versus the critical paths of ISCAS'99 benchmarks [18] using SYNOPSYS tools. ISCAS'99 contains 29 designs from small circuits with 21 cells to more complicated designs with almost 44K cells. 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, Primetime for static timing analysis (STA), Tetramax for automatic test pattern generation (ATPG), and Vcs for back annotated simulation. Since

TABLE I. ERROR OF TF VERSUS STA

Benchmark	50p	100p	200p	500p	Benchmark	50p	100p	200p	500p
$b01$	1.00%	1.00%	1.00%	1.00%	$b15$	2.80%	2.75%	2.46%	2.06%
$b02$	3.81%	3.81%	3.81%	3.81%	$b15\_1$	7.44%	7.38%	3.21%	2.57%
$b03$	4.04%	4.04%	4.04%	4.04%	$b17$	4.24%	4.21%	3.71%	3.68%
$b04$	2.97%	2.57%	1.70%	1.70%	$b17\_1$	8.29%	5.26%	4.91%	4.91%
$b05$	1.28%	1.28%	1.21%	1.21%	$b18$	15.64%	12.25%	10.54%	6.47%
$b06$	3.64%	3.64%	3.64%	3.64%	$b18\_1$	14.53%	7.89%	7.57%	7.47%
$b07$	5.83%	2.20%	2.20%	2.20%	$b19$	17.80%	15.90%	15.98%	12.42%
$b08$	2.84%	2.00%	2.00%	2.00%	$b19\_1$	8.83%	8.82%	8.82%	8.82%
$b09$	7.50%	7.50%	7.50%	7.50%	$b20$	13.23%	12.53%	12.29%	10.00%
$b10$	0.05%	0.05%	0.05%	0.05%	$b20\_1$	15.99%	15.70%	11.48%	9.94%
$b11$	2.19%	0.46%	0.20%	0.20%	$b21$	12.62%	12.82%	7.62%	7.62%
$b12$	1.82%	1.82%	1.82%	1.67%	$b21\_1$	4.96%	4.47%	4.45%	3.42%
$b13$	2.35%	2.35%	2.35%	2.35%	$b22$	11.22%	10.38%	10.27%	10.27%
$b14$	18.55%	18.52%	18.52%	11.29%	$b22\_1$	12.05%	12.01%	11.94%	8.54%
$b14\_1$	19.23%	14.01%	14.01%	13.66%	-	-	-	-	-

functional pattens are not available for ISCAS'99 benchmarks, we use STA instead as a reference for comparison versus TF 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. Also, we note that identifying the most critical part of the application is not possible in most cases.

Fig. 1 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 FD-SOI library to extract the critical timing of benchmarks in each corner. (42 corners*29 netlists= 1218 critical timing reports)
- **ATPG**: transition fault test pattern generation to extract test patterns and test benches for each benchmark. We generated 4 pattern sets (targeting only register to register paths) for each benchmark including 50, 100, 200, and 500 patterns. (29 netlists * 4 pattern sets = 116 pattern sets and test benches).
- **Simulation**: applying transition fault test patterns on back annotated simulation of each benchmark, and searching for maximum frequency at which each device passes the test. Transition fault frequency search is done using binary search and STA results as a starting point since TF maximum frequency cannot exceed critical timing.

Finally, we compared STA results versus transition fault frequencies of 29 ISCAS'99 circuits in 42 corners. The results are presented in the next subsection.

B. Simulation results

To understand if TF testing is a reasonable performance indicator that can be used for AVS during production, we compared the maximum frequency at which TF 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 TF. In order to present the results, we define a parameter named $error$ which is measured for each benchmark. The concept relates to how much margin should be taken into account due to inaccuracies as a result of performance estimation using TF. To be able to measure $error$ for each benchmark, first we measured performance error for each corner by:

$$error_{corner} = (P_{STA} - P_{TF})/P_{STA} \quad (1)$$

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

$$error = \max_{all\ corners} [error_{corner}] \quad (2)$$

Table I presents the $error$ for all benchmarks. We generated the results for 4 pattern sets including 50, 100, 200, and 500 patterns. As it can be seen in this table, depending on the size of each benchmark, with increasing pattern count, the $error$ is reduced. Therefore, depending on the time invested on testing during production, the accuracy of performance estimation using TF can be improved. As mentioned earlier, for some small benchmarks such as b01 with only 30 cells, the error remains unchanged since there are no more patterns that can be used to increasing the coverage.

As a conclusion, we presented the average error of all ISCAS'99 benchmarks for each pattern set in Table II. Increasing pattern count from 50 to 500 results in 19.76% increase in coverage, and thus 2.47% error improvement in average for ISCAS'99 benchmarks. According to these results, we can conclude that using transition fault testing for performance estimation achieves inaccuracy as low as 5.33%.

This measured $error$ means that in order to make sure the performance estimation using TF is accurate enough, a margin should be added on top of the estimated performance. If the inaccuracy of performance estimation using TF is predictable,

TABLE II. AVERAGE ERROR PERCENTAGE OF TF VERSUS STA FOR 50,100,200, AND 500 PATTERN SETS

Pattern count	Coverage	error
50p	43.21%	7.85%
100p	49.82%	6.81%
200p	55.01%	6.18%
500p	62.97%	5.33%

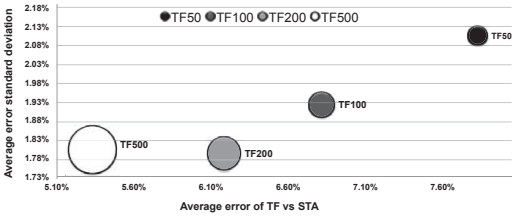

Fig. 2. Average error vs average standard deviation of error for 50, 100, 200, and 500 pattern sets; the size of the bubble represents the size of the pattern set used

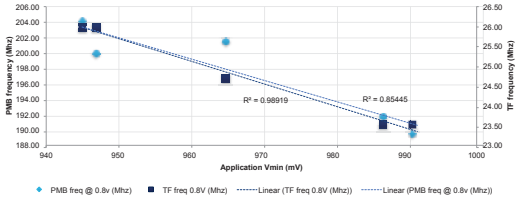

Fig. 3. Application Vmin versus TF and PMB results [19]

it is possible to come up with a safe margin. Figure 2 illustrates the average standard deviation of the estimation error plotted versus the average error measured using TF for all the circuits in the ISCAS'99 benchmark. The plotted measurements are clustered by the size of the test pattern set (reflected by the size of the circle in the plot). The figure shows that 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 TF can be improved. However, moving from 200 to 500 patterns, the average standard deviation remained unchanged, which means that increasing pattern count up to a limit reduces the uncertainty. Further than that, the uncertainty remains unchanged even though the error is improved.

IV. Industrial Case Study

In this section, we compare PMB versus TF for AVS during production using measurements on real silicon. Our case study is a 28nm FD-SOI device on which a number of PMBs are distributed. These PMBs are ring oscillators designed based on the most frequently used cells extracted from the critical paths of various designs. During production, according to the frequency responses of PMBs, optimal voltage estimation is done for each chip. Alternatively, voltage estimation can be done using transition fault testing during production as well at no extra costs. Also, since transition fault testing represents a direct measurement of chip performance, the expensive correlation during the characterization phase is not needed anymore, which reduces time to market dramatically.

We have done silicon measurement on 5 chip samples. First, we have measured the real value of optimal voltage (Vmin) for each chip operating at its nominal frequency using functional patterns. Then, we set an arbitrary voltage for each chip (0.8v) and collected frequency responses from PMBs. We also measured TF maximum frequency for each chip using the same voltage settings (0.8v). To understand whether PMB or TF is more accurate for performance prediction, we have to identify which of them is more correlated with application Vmin. Therefore, we mapped both frequency response of PMB and the TF frequency to the Vmin of the chip in which that PMB is located. Then, we performed a linear least square regression analysis of the correlation between application Vmin and PMB frequency as well as the TF frequency, and measured the coefficient of determination (R^2) for both correlation functions. R^2 is a key output of regression analysis. It is interpreted as the proportion of the variance in the dependent variable (Vmin in this case) that is predictable from the independent variable (PMB frequency and TF frequency). Results are presented in Figure 3. R^2 for the correlation of application Vmin versus PMB is 0.85, while it has a value of 0.99 for the correlation versus TF, which means that Vmin estimation using the PMB approach can only account for 85% of the variability in the measurements, while Vmin estimation using the TF approach can account for 99% of that variability. These results confirm that we can achieve higher accuracy in Vmin estimation using TF [19].

V. Conclusions

This paper proposed a new approach that uses transition fault testing 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 TF tests (already used during production) can replace a long and expensive process of PMB characterization, reducing cost and time to market dramatically. Moreover, since transition fault test patterns target all gates and indirectly cover all path-segments, it is a better performance representative than PMBs. As functional patterns are not used anymore, testing approach could be a solution for general logic, not only for CPU and GPU. According to simulation results of the 29 ISCAS'99 chips on 42 corners of a 28nm FD-SOI library, using transition fault testing for performance estimation ends up with an inaccuracy as low as 5.33% and a standard deviation of 1.8%. The paper also showed that the PMB approach can only account for 85% of the uncertainty in voltage measurements, which results in power waste, while the TF-based approach can account for 99% of that uncertainty.

References

[1] M. Zandrahimi and Z. Al-Ars, *A Survey on Low-power Techniques for Single and Multicore Systems*, in ICCASA, pp. 69-74, 2014.
[2] T. Chan and A.B. Kahng, *Tunable Sensors for Process-Aware Voltage Scaling*, in ICCAD, pp. 7-14, 2012.
[3] T. Chan, et al., *DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators*, in ISQED, pp. 633-640, 2012.
[4] A. Drake, et al., *A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor*, in ISSCC, pp. 398-399, 2007.
[5] TD. Burd, et al., *A dynamic voltage scaled microprocessor system*, in ISSCC, pp. 294-295, 2000.
[6] J. Kim and M.A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IJSSC, vol. 37, no. 5, pp. 639-647, 2002.
[7] Q. Liu and S.S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.
[8] G. Li Zhang, et al., *EffiTest: Efficient Delay Test and Statistical Prediction for Configuring Post-silicon Tunable Buffers*, in DAC 2016.
[9] M. Sauer, et al., *On the Quality of Test Vectors for Post-Silicon Characterization*, in ETS 2012.
[10] P. Das, et al., *On Generating Vectors for Accurate Post-Silicon Delay Characterization*, in ATS 2011.
[11] N. B. Zain Ali, et al., *Dynamic Voltage Scaling Aware Delay Fault Testing*, in ETS 2006.
[12] K. N. Shim and J. Hu, *A low Overhead Built-In Delay Testing with Voltage and Frequency Adaptation for Variation Resilience*, in DFT 2012.
[13] M. Zandrahimi, Z. Al-Ars, P. Debaud, and A. Castillejo, *Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation*, in DATE, 2016.
[14] M. Tehranipoor et al., Test and Diagnosis for Small-Delay Defects, in Springer Science+Business Media, LLC, 2011.
[15] L.H. Goldstein, E.L. Thigpen, SCOAP: Sandia Controllability/Observability Analysis Program, in DAC, 1980.
[16] B. Kruseman, A. Majhi, and G. Gronthoud, On Performance Testing with Path Delay Patterns, in VTS, 2007.
[17] M. Zandrahimi, P. Debaud, A. Castillejo, and Z. Al-Ars, *Using Transition Fault Test Patterns for Cost Effective Offline Performance Estimation*, in DTIS, 2017.
[18] http://www.cad.polito.it/downloads/tools/itc99.html
[19] M. Zandrahimi, P. Debaud, A. Castillejo, and Z. Al-Ars, *Transition Fault Testing for Offline Adaptive Voltage Scaling*, in ITC, 2017.

5

SDD-BASED AND PDLY-BASED AVS

In this chapter we introduce the new method of using SDD as well as PDLY test patterns, not only to validate the functionality of the devices, but also as an alternative solution for performance estimation, that can be used for offline adaptive voltage scaling. Next, we investigate the effectiveness of the proposed methods on ISCAS'99 benchmarks using an industrial grade 28 nm FD-SOI library developed for low power devices. The results show that using SDD testing results in 3.96% performance estimation error with 1.59% error standard deviation. PDLY testing for performance estimation results in a more accurate estimation error of only 1.85% with a standard deviation of 1.34%.

This chapter is based on the following papers.

1. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Cost Effective Adaptive Voltage Scaling Using Path Delay Fault Testing*, East-West Design & Test Symposium (EWDTS 2018), 14-17 September, Kazan, Russia.

2. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *An Industrial Case Study of Low Cost Adaptive Voltage Scaling Using Delay Test Patterns*, Design, Automation and Test in Europe (DATE 2018), 19-23 March 2018, Dresden, Germany.

Cost Effective Adaptive Voltage Scaling Using Path Delay Fault Testing

Mahroo Zandrahimi*, Philippe Debaud†, Armand Castillejo†, Zaid Al-Ars*
*Delft University of Technology, The Netherlands
{m.zandrahimi, z.al-ars}@tudelft.nl
†STMicroelectronics, Grenoble, France
{philippe.debaud, armand.castillejo}@st.com

Abstract–Application of manufacturing testing during the production process of integrated circuits is considered essential to ensure the quality of the devices used in the field. However, it is desirable to use the information gathered during the test process to add value to other aspects of the manufacturing process. This paper proposes a method to use path delay (PDLY) test patterns, not only to validate the functionality of the devices, but also as an alternative solution for performance estimation, that can be used for offline adaptive voltage scaling. This approach has many advantages over the currently used industrial performance estimation methods, so-called performance monitoring boxes (PMBs). Using simulation of ISCAS'99 benchmarks with 28nm FD-SOI libraries, the paper shows that the PDLY based approach reduces the inaccuracy of performance prediction from 2.32% (achieved by the classic PMB approach) to 1.85%, without the need for any on-chip monitors.

I. INTRODUCTION

The proliferation of the use of integrated circuits as ubiquitous building blocks in various application domains, including mission critical applications, imposes stringent quality requirements on devices used in the field. This results in the application of long and expensive test programs that may cost more than 30% of the total price tag of a given device to ensure these high quality requirements [1]. Such elaborate test programs, however, are able to identify not only failure mechanisms, but also provide a detailed characterization of the behavior of tested devices. Viewed in this light, testing can provide insights that can have added value in various stages of the manufacturing process, a concept that is referred to as value-added testing [2], [3].

This paper describes an approach to apply the concept of value-added testing using path delay (PDLY) test patterns as a cost effective method for characterizing the performance of manufactured devices. PDLY patterns are generated to activate and test the top most critical paths in a given device, and therefore provide an accurate representation of the time-dependant behavior of the device under test. This can subsequently be used as a source of information to guide various design related processes. This paper is mainly concerned with using PDLY patterns for the application of adaptive voltage scaling (AVS) during production.

AVS is one of the most important power reduction techniques used in the semiconductor industry today to manage the ever increasing power dissipation of IC devices [4]. The AVS technique scales the supply voltage of the device to manage power consumption, which subsequently also affects the performance of the device. In order to do this effectively, AVS needs accurate information about the performance characteristics of each and every manufactured device. The current state-of-the-art technique today to allow for this accurate performance characterization is carried out using a number of performance monitoring boxes (PMBs) distributed through each device [5]–[9]. The accuracy of these PMBs continues to degrade with the decreasing feature sizes of integrated circuits, thereby reducing their accuracy and effectiveness.

In order to use PMBs for performance estimation during production, the correlation between PMB frequencies and actual frequency of the circuit is measured during the characterization stage, which is an early stage in the manufacturing process. Once the actual performance of the circuit can be estimated through PMB frequency responses using the correlation function measured during the characterization, PMBs are ready to be used for performance prediction during production. Figure 1 shows an example of a chip with different voltage islands on which various kinds of PMBs are placed. To identify the speed of NMOS and PMOS transistors for each chip during production, two PMBs are designed using PMOS and NMOS speedometers, while, the third shown PMB is a critical path replica designed based on the most used logic cells extracted from the potential critical paths of the design reported by static timing analysis. During production and based on the frequency responses from these monitors, chip performance is estimated, and optimal voltages are measured for each manufactured chip. This information could be used to either sort devices based on their speed (so-called speed binning), adapt voltage to enhance yield, or optimize power and battery lifetime using voltage scaling and body biasing [10].

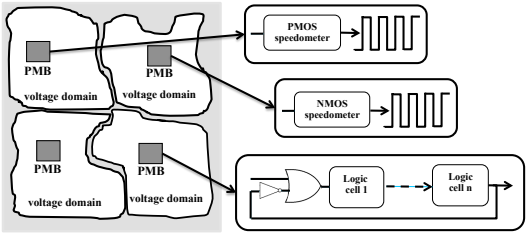

Fig. 1. Performance prediction using PMBs

However, the process of finding the correlation function with which we can measure the actual frequency of the circuit using PMB frequency responses, should be done for an amount of test chips representative of the process window to make sure (for all manufactured chips) performance prediction 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. On the other hand, since functional patterns are used in the process of finding the correlation of PMB responses to the actual frequency of the circuit, PMB approaches are not suitable for general logic. 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.

This paper applies the idea of value-added-testing by using PDLY test patterns as a cost effective approach for AVS during production. This paper has the following contributions:

- Proposing the concept of using PDLY as an alternative to PMBs for performance prediction.
- Introducing a test flow to enable the application of PDLY testing for AVS during production.
- An analysis of the accuracy as well as the effectiveness of PDLY test patterns for AVS using 29 ISCAS'99 benchmarks based on a 28nm FD-SOI industrial library, in various process corners (42 corners in total).
- A comparison of the accuracy of PDLY based approach as compared to PMBs to predict circuit performance, showing comparable prediction capabilities by the PDLY based approach.

This paper is organized as follows. First, we discuss in Section II the challenges of using PMBs in nanometric technologies and the need to propose new more effective performance prediction methods. Section III proposes the concept of using path delay testing for performance prediction. It also introduces a test flow to enable AVS using path delay test patterns during production. Section IV evaluates the proposed approach using simulation models for ISCAS'99 benchmarks. Section V concludes the paper.

II. Motivation

With the continued reduction in feature sizes, process variations play a more significant role in defining the performance characteristics of manufactured devices. This, however, results in significant limitations on the effectiveness of PMBs to accurately estimate circuit performance. As a result, an increasing number of environmental parameters should be taken into account (such as voltage and temperature variations and aging) and tend to prevent accurate performance prediction, which is used to compute optimum operation voltages exclusively for each chip during production. To investigate the accuracy and effectiveness of PMB approaches, we performed static timing analysis (STA) using Primetime [15] (SYNOPSYS tool for STA) on ISCAS'99 benchmarks [16] using 28nm FD-SOI library. ISCAS'99 contains 29 benchmarks from small circuits with 21 cells to more complicated benchmarks with almost 44K cells. Table I lists the characteristics of the 42

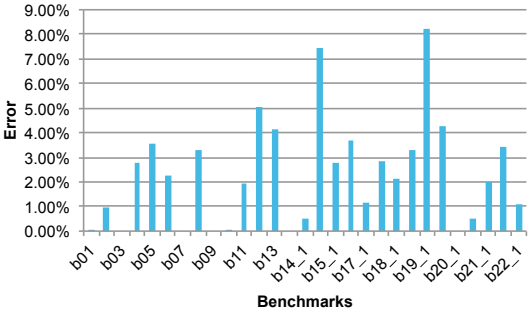

Fig. 2. Percentage of *error* for ISCAS'99 benchmarks using 28nm FD-SOI library [17]

different corners used in the STA simulation for the 28nm FD-SOI library with voltage, body biasing, temperature, transistor speed and aging parameters.

The results of the simulation are expressed in terms of the performance error in the PMB prediction. We assume that the PMB performance prediction for each benchmark is represented by the critical path reported by STA in the typical corner for that benchmark. The characteristics of the typical corner simulation are (TT, 0.85, 25, no, no), as highlighted as the bold row in Table I. Then, we estimate the performance of the design in the 41 other corners using that PMB (represented by the typical corner simulation). In order to quantify the results, we define a parameter named *error* which is measured for each benchmark. The concept relates to how much margin should be taken into account due to inaccuracies as a result of performance prediction using PMBs. To be able to measure *error* for each benchmark, first we check if the critical path in each corner is different from the critical path of the typical corner (PMB for each benchmark). In the case of critical path difference, we measure $error_{corner}$ for the process corner by:

$$error_{corner} = (P_{corner} - \text{PMB})/P_{corner} \quad (1)$$

where P_{corner} is the delay of the critical path measured in $corner$, and PMB is the delay of the critical path identified in the typical corner but measured in $corner$. Once $error_{corner}$ is calculated for all process corners, *error* can be obtained for each benchmark by:

$$error = Average_{all\ corners}[error_{corner}] \quad (2)$$

We performed the analysis to calculate the *error* for the different ISCAS'99 benchmark circuits. The results of the analysis are shown in Figure 2, which indicates that only a small minority of the benchmarks have a negligible error. Most benchmarks actually have a rather high simulated error that can go up to a value as high as 8.20% in the case of $b19\_1$. The detailed list of simulated error percentage for all 29 ISCAS'99 benchmarks is given in Table II. As the results in the table indicate, most of the simulated benchmarks have different critical paths in different corners, which results in an alleviated error measurement. Since any moderately complex

TABLE I. FEATURES OF DIFFERENT CORNERS OF 28NM FD-SOI LIBRARY USED IN SIMULATIONS

Corner	Voltage [V]	Temperature [°C]	Biasing	Aging	Corner	Voltage [V]	Temperature [°C]	Biasing	Aging
SS	0.7	-40	no	no	SS	0.7	0	no	no
SS	0.7	125	no	no	SS	0.7	-40	no	yes
SS	0.7	0	no	yes	SS	0.7	125	no	yes
SS	0.8	-40	no	no	SS	0.8	0	no	no
SS	0.8	125	no	no	SS	0.8	-40	no	yes
SS	0.8	0	no	yes	SS	0.8	125	no	yes
TT	0.8	25	no	no	TT	0.8	125	no	no
SS	0.85	-40	no	no	SS	0.85	0	no	no
SS	0.85	125	no	no	SS	0.85	-40	no	yes
SS	0.85	0	no	yes	SS	0.85	125	no	yes
TT	**0.85**	**25**	**no**	**no**	TT	0.85	125	no	no
SS	0.9	-40	yes	no	SS	0.9	125	yes	no
SS	0.9	-40	no	no	SS	0.9	0	no	no
SS	0.9	125	no	no	SS	0.9	-40	no	yes
SS	0.9	0	no	yes	SS	0.9	125	no	yes
TT	0.9	125	no	no	TT	0.9	25	no	no
FF	0.9	-40	no	no	FF	0.9	125	no	no
SS	0.95	-40	no	no	SS	0.95	0	no	no
SS	0.95	125	no	no	SS	0.95	-40	no	yes
SS	0.95	0	no	yes	SS	0.95	125	no	yes
TT	0.95	25	no	no	TT	0.95	125	no	no

T and S stand for typical and slow corners, respectively.

TABLE II. ERROR IN PERFORMANCE PREDICTION USING ONE PMB FOR ISCAS'99 BENCHMARKS WITH 28NM FD-SOI LIBRARY

Benchmark	# Cells	*error*	Benchmark	# Cells	*error*
*b*01	30	0.02%	*b*15	3142	7.45%
*b*02	21	0.96%	*b*15_1	3141	2.77%
*b*03	76	0.00%	*b*17	9559	3.67%
*b*04	196	2.80%	*b*17_1	9584	1.14%
*b*05	390	3.53%	*b*18	22175	2.86%
*b*06	29	2.27%	*b*18_1	22093	2.14%
*b*07	179	0.00%	*b*19	43916	3.31%
*b*08	71	3.31%	*b*19_1	43822	8.20%
*b*09	94	0.00%	*b*20	3970	4.25%
*b*10	110	0.07%	*b*20_1	4025	0.00%
*b*11	326	1.96%	*b*21	4022	0.48%
*b*12	547	5.04%	*b*21_1	4082	2.02%
*b*13	154	4.12%	*b*22	6102	3.45%
*b*14	1967	0.00%	*b*22_1	6164	1.08%
*b*14_1	2043	0.49%	Average	-	2.32%

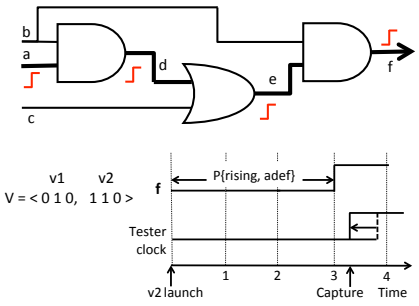

Fig. 3. An example of performance prediction using path delay testing

circuit has a large amount of critical paths to evaluate, it is rather difficult to predict the delay of all these paths in such a circuit based on information from a single unique path. Such an attempting to predict the delay with only one path would result in quite inaccurate predictions. In addition, this prediction using PMBs is becoming increasingly more difficult with the decreasing feature sizes and increasing variations in deep sub-micron technologies. As a result, we need to come up with new approaches to increase the accuracy of performance prediction by taking more critical paths into consideration.

III. PDLY BASED AVS

In this section, we discuss the way path delay tests can be used to measure the performance of a circuit, and then we propose an algorithm to predict the voltage.

A. Basic concept

In order to show the basic idea of how circuit performance can be predicted using path delay testing, we show a simple example, presented in Figure 3, how performance of a circuit is predicted using path delay test patterns. Assume that the path P{rising, adef} in this figure (the highlighted path) is one of the critical paths of the circuit reported by STA. The path delay test pattern needed to propagate the rising transition from input a to output f is the vector pair $V =< 010, 110 >$. 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 that 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.

B. Voltage prediction

In this paper, we propose an innovative test flow for AVS using path delay testing during production. Since delay testing

covers many path-segments of the circuit, it can be a better performance representative than a PMB. Such an approach has a number of unique advantages as compared to PMB-based approaches. First, this approach can be performed *at a less cost*, since delay tests are routinely performed during production to test for chip functionality. In addition, since delay testing is performed to explicitly test for actual chip performance, the expensive effort of correlating PMB responses to chip performance during the characterization stage of manufacturing is not needed anymore, which reduces the cost and time to market dramatically. 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. And last but not least, this approach makes using PMBs redundant, which saves silicon area as well as PMB design time.

There are three different types of delay test patterns we can consider for performance prediction: transition fault (TF) tests, small delay defect tests (SDD), and path delay tests (PDLY) [11]. 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. However, ATPG choices are based on heuristics like SCOAP [12], which tend to minimize computational effort. Thus, when several solutions are available for path sensitization, ATPG will use the easiest, which means that the tool tends to target short paths and not critical paths of the design [13]. On the other hand, we can alternatively use SDD testing, which sensitizes paths with smallest slacks, as well as PDLY testing, which sensitizes a selected path (potential critical paths reported by STA). In PDLY testing, the objective is to obtain a transition along critical paths which are on average longer and more complex than the paths targeted in TFs. At the cost of a higher pattern count, PDLY can be seen as more promising in terms of accuracy in predicting circuit performance since they sensitize the longest paths (i.e., those which are limiting the chip frequency), which is an advantage over TF testing. In this paper, we target PDLY test patterns for performance prediction.

Figure 4 proposes a flow of PDLY based approach that could be used during production. The proposed flow performs a binary search to identify the optimal voltage (Vmin) at which the chip can pass all path 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, 2. set supply voltage to Vmax and wait for stabilization, 3. apply PDLY at speed test, 4. if the chip fails the test, discard it, otherwise, 5. compute new values and do a binary search to find Vmin. Conversion from Vmin to Fmax might be required depending on either performance prediction is done for power optimization or yield enhancement. ”e” is an arbitrary value which is up to the users to define the resolution they want.

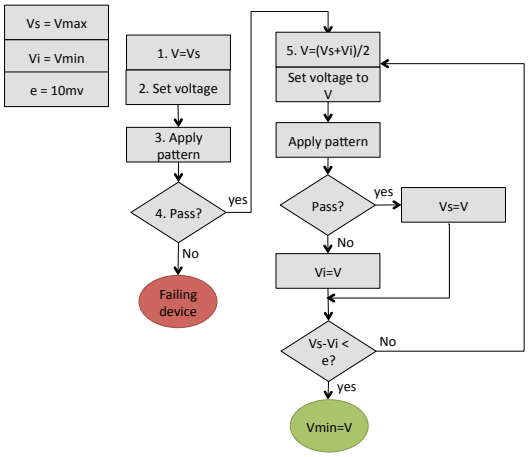

Fig. 4. Proposed flow for using path delay test patterns for offline AVS

IV. Evaluation

A. Simulation set up

This subsection explains the flow we used to explore if PDLY tests correlate with the actual frequency of the circuits. We use 28nm FD-SOI [14] libraries to compare PDLY maximum frequency versus the critical paths of ISCAS’99 benchmarks using SYNOPSYS tools [15]. We used Design Compiler in topographical mode for physical synthesis, Primetime 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 PDLY frequencies. This choice can be justified by noting that STA represents worst case circuit performance, which can be used as a lower bound for maximum circuit frequency. On the other hand, 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. This especially true since identifying the most critical part of the application is not possible in most cases.

Figure 5 shows the simulation flow containing 4 steps as follows:

- Synthesis: physical synthesis on 29 ISCAS’99 circuits using 28nm 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 FD-SOI library to extract the critical timing of benchmarks in each corner. (42 corners*29 netlists= 1218 critical timing reports)
- ATPG: PDLY test pattern generation to extract test patterns and test benches for each benchmark. We generated a PDLY pattern set for each benchmark with 10000 targeted paths. Pattern set targeting 10000 paths lead to an average of 174 patterns.
- Simulation: applying delay test patterns on back annotated simulation of each benchmark, and searching for maximum frequency at which each device passes the test. PDLY frequency search is done using binary search and STA results as a starting point since the maximum frequency cannot exceed critical timing.

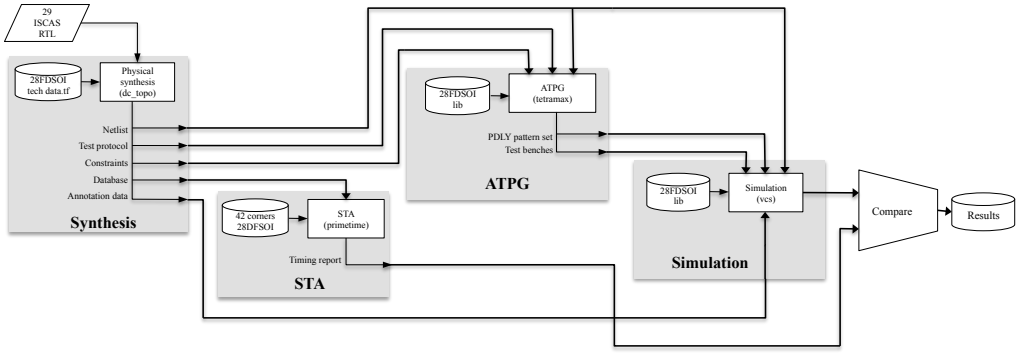

Fig. 5. Simulation flow for comparing PDLY frequency vs. STA for the 29 ISCAS'99 circuits.

Finally, we compared STA results versus PDLY frequency of ISCAS'99 circuits in 42 corners and calculated the error, which is the percentage of difference between STA results and performance prediction using PDLY pattern set targeting 10000 paths. The results are presented in the next subsection.

B. Simulation results

To understand if PDLY based approach is a reasonable performance indicator that can be used for AVS during production, we compared the maximum frequency at which PDLY 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 PDLY. In order to present the results, we define a parameter named $error$ which is measured for each benchmark. The concept relates to how much margin should be taken into account due to inaccuracies as a result of performance prediction using PDLY. To be able to measure $error$ for each benchmark, first we measured performance error for each corner by:

$$error_{corner} = (P_{STA} - P_{PDLY})/P_{STA} \quad (3)$$

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

$$error = Average_{\textit{all corners}}[error_{corner}] \quad (4)$$

To illustrate this process of calculating the error, we show an example of calculating error of performance prediction using PDLY for one of the benchmarks (b17) with 9559 cells. First STA is done, and critical timing of b17 in each of 42 corners is determined, which will be used as a reference for comparison against performance prediction using PDLY (Table III). Then path delay test patterns for b17 is generated using Tetramax. Next step is the maximum frequency search using the binary search, during which the maximum frequency at which b17 passes all test patterns is determined. The binary search is done to find the minimum period between when the test is launched and the correct output is captured for all test patterns. This period represents the performance prediction using PDLY. The error (the difference between performance prediction using PDLY and STA divided by STA) is presented in Table III for all 42 corners. Finally, we assume the error of performance prediction using PDLY for benchmark b17 is the average error of 42 corners, which is 1.82%.

TABLE III. ERROR CALCULATION OF PERFORMANCE PREDICTION USING PDLY FOR B17

Corner	error	Corner	error
1	0.04%	22	1.69%
2	0.02%	23	2.13%
3	2.77%	24	3.76%
4	0.01%	25	1.51%
5	0.03%	26	1.07%
6	2.75%	27	3.62%
7	0.07%	28	0.57%
8	0.45%	29	1.15%
9	2.29%	30	3.67%
10	0.13%	31	3.57%
11	0.36%	32	2.28%
12	2.36%	33	1.12%
13	1.09%	34	3.41%
14	2.26%	35	1.17%
15	0.53%	36	1.73%
16	1.17%	37	4.12%
17	2.87%	38	2.34%
18	0.41%	39	2.81%
19	1.98%	40	3.06%
20	2.81%	41	1.86%
21	1.31%	42	4.12%

Table IV presents the $error$ for all benchmarks. We generated the results for a pattern set including 10000 PDLY patterns. As it can be seen in this table, the calculated error changes depending on the simulated benchmark circuit. The error ranges between values as low as 0.05% up to a maximum of 8.70%. Only 2 benchmark circuits out of 29 have an error that is higher than 5%. Almost half the benchmarks have an error that is lower than 1%.

On average, the PDLY induced error calculated for all simulated benchmarks is equal to 1.85%. In comparison, the

TABLE IV. ERROR OF PDLY VERSUS STA

Benchmark	error	Benchmark	error
b01	0,33%	b15	0,80%
b02	0,11%	b15_1	0,57%
b03	4,05%	b17	1,82%
b04	1,70%	b17_1	1,89%
b05	1,21%	b18	0,46%
b06	3,64%	b18_1	4,03%
b07	2,20%	b19	4,58%
b08	1,95%	b19_1	8,70%
b09	7,50%	b20	1,50%
b10	0,05%	b20_1	0,43%
b11	0,20%	b21	0,50%
b12	1,82%	b21_1	0,33%
b13	2,35%	b22	0,17%
b14	0,23%	b22_1	0,41%
b14_1	0,22%	Average	1,85%

average PMB induced error for all benchmarks is equal to 2.32%, which is higher than the error for PDLY. It is important to note that the simulated PMB error is the ideal best case error that PMB can offer. In practice, this error will be worse due to a number of reasons, such as: the PMB on the chip is located at a distance from the critical paths, PMBs are designed as replicas of critical paths, and most importantly the fact that PMBs are only able to approximate critical path behavior using a correlation function.

V. CONCLUSIONS

Classic PMB approaches for AVS during production show more and more limitations in terms of high cost and increased complexity as technology scaling enters the nanometer regime. These techniques predict circuit performance indirectly based on PMB frequency responses and correlating them with the actual frequency of the circuit. The correlation function is measured during the characterization stage of chip manufacturing using a sample of chips representative of the process window, including fast and slow devices. This makes the approach both expensive to design and lengthy to implement. As an alternative, this paper proposes using information gathered during the application of PDLY test patterns to characterize the performance of manufactured chips, as an example of the concept of value-added-testing. The PDLY based approach can replace PMBs to apply AVS during production. Using simulation of ISCAS99 benchmarks with 28nm FD-SOI libraries, the paper shows that the PDLY based approach reduces the inaccuracy of performance prediction from 2.32% (achieved by the classic PMB approach) to 1.85%, without the need for any on-chip monitors.

ACKNOWLEDGEMENTS

This work is carried out under the BENEFIC project (CA505), a project labelled within the framework of CATRENE, the EUREKA cluster for Application and Technology Research in Europe on NanoElectronics.

REFERENCES

[1] Z. Al-Ars, *DRAM Fault Analysis and Test Generation*, Delft University of Technology, Delft, Netherlands, June, 2005.

[2] J. Jahangiri and D. Abercrombie, *Value-added defect testing techniques*, in IEEE Design & Test of Computers, vol. 22, no. 3, pp. 24-231, 2005.

[3] Z. Al-Ars and A.J. van de Goor, *Impact of memory cell array bridges on the faulty behavior in embedded DRAMs*, in Asian Test Symposium, pp. 282-289, 2000.

[4] TD. Burd, et al., *A dynamic voltage scaled microprocessor system*, in International Solid-State Circuits Conference, pp. 294-295, 2000.

[5] T. Chan and A.B. Kahng, *Tunable Sensors for Process-Aware Voltage Scaling*, in International Conference On Computer Aided Design, pp. 7-14, 2012.

[6] T. Chan, et al., *DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators*, in International Symposium on Quality Electronic Design, pp. 633-640, 2012.

[7] A. Drake, et al., *A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor*, in International Solid-State Circuits Conference, pp. 398-399, 2007.

[8] J. Kim and M.A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IEEE Journal of Solid-State Circuits, vol. 37, no. 5, pp. 639-647, 2002.

[9] Q. Liu and S.S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in IEEE Transactions on Computer-Aided Design, vol. 29, no. 2, pp. 211-222, 2010.

[10] M. Zandrahimi and Z. Al-Ars, *A Survey on Low-power Techniques for Single and Multicore Systems*, in International Conference on Context-Aware Systems and Applications, pp. 69-74, 2014.

[11] M. Tehranipoor et al., Test and Diagnosis for Small-Delay Defects, in Springer Science+Business Media, LLC, 2011.

[12] L.H. Goldstein, E.L. Thigpen, SCOAP: Sandia Controllability/Observability Analysis Program, in Design Automation Conference, 1980.

[13] B. Kruseman, A. Majhi, and G. Gronthoud, On Performance Testing with Path Delay Patterns, in VLSI Test Symposium, 2007.

[14] http://www.st.com/content/st_com/en/about/innovation—technology/FD-SOI.html

[15] http://www.synopsys.com/tools/pages/default.aspx

[16] http://www.cad.polito.it/downloads/tools/itc99.html

[17] M. Zandrahimi, et al., Using Transition Fault Test Patterns for Cost Effective Offline Performance Estimation, in International Conference on Design Technology of Integrated Systems in Nanoscale Era, 2017.

An Industrial Case Study of Low Cost Adaptive Voltage Scaling Using Delay Test Patterns

Mahroo Zandrahimi*, Philippe Debaud†, Armand Castillejo†, Zaid Al-Ars*
*Delft University of Technology, The Netherlands
{m.zandrahimi, z.al-ars}@tudelft.nl
†STMicroelectronics, Grenoble, France
{philippe.debaud, armand.castillejo}@st.com

Abstract–In deep sub-micron technologies, the increasing effect of process and environmental variations has lead chip manufacturers to use adaptive voltage scaling techniques in order to adapt operation parameters exclusively to each chip. The increasing effect of process variation is limiting the effectiveness of current chip monitoring approaches, such as on-chip performance monitor boxes (PMBs), which results in yield loss and high design margins, thus high power consumption. This paper proposes an alternative solution for adaptive voltage scaling using delay test patterns, which is able to eliminate the need for PMBs, and thus the long expensive characterization phase of tuning PMBs to each design, while improving the yield as well as power optimization. Results show, using an industrial grade 28nm FD-SOI library developed for low power devices, that delay testing for performance prediction reduces the inaccuracy down to 1.85%.

I. INTRODUCTION

Adaptive voltage scaling (AVS) has become a standard approach used by chip manufacturers to ensure low power consumption of their devices [2]. The effectiveness of the AVS approaches depends on appropriately predicting the performance of every manufactured device under specific input voltage values. This device-specific prediction is realized by using on chip performance monitoring boxes (PMBs) integrated on each device, that allows fast performance prediction and voltage pairing during production. Fig. 1 shows an example of a chip, on which various kinds of PMBs are distributed. The figure shows two PMBs created using PMOS and NMOS speedometers that indicate the speed of PMOS and NMOS transistors, while the third shown PMB is a critical path replica designed based on the most used logic cells extracted from the potential critical paths of the design. During production and based on the frequency responses from these monitors, chip performance is estimated, and corresponding voltage is fused for that operation point [1].

However, the correlation process during the characterization stage (i.e., finding the correlation between PMB responses and the actual frequency of the circuit) makes these techniques very expensive, since it should be done for an amount of test chips representative of the process window to make sure (for all manufactured chips) performance prediction based on PMB responses is correlated with application behavior.

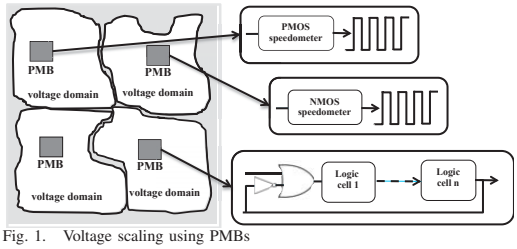

Fig. 1. Voltage scaling using PMBs

In this paper we introduce a cost effective approach for performance prediction during production using small delay defect (SDD) and path delay (PDLY) test patterns, which can be used for general logic as well. We investigate the proposed approach in terms of accuracy and effectiveness using 29 ISCAS'99 benchmarks with an industrial grade 28nm FD-SOI library for 42 different process corners with different characteristics in terms of process and environmental variations as well as aging.

The rest of this paper is organized as follows. Section II proposes the concept of using delay faults for performance prediction. Evaluation of the proposed approach is presented in Section III using simulation results on ISCAS'99 benchmarks. Section IV concludes the paper.

II. AVS USING DELAY TESTING

In this paper, we propose an innovative test flow for voltage estimation using delay testing during production. Since delay testing covers many path-segments of the circuit [3], [4], it can be a better performance representative than PMBs. Such an approach has a number of unique advantages as compared to PMB-based approaches. Since delay testing is performed to explicitly test for actual chip performance, the expensive effort of correlating PMB responses to chip performance during the characterization stage of manufacturing is not needed anymore, which reduces the cost and time to market dramatically. 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. And last but not least, this approach makes using PMBs redundant, which saves silicon area as well as PMB design time.

The proposed flow performs a binary search to identify the minimum voltage (Vmin), 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, 2. Set supply voltage to Vmax and wait for stabilization, 3. Apply transition fault at speed test, 4. If the chip fails the test, discard it, otherwise, 5. compute new values and do a binary search to find Vmin. Conversion from Vmin to Fmax might be required depending on either performance estimation is done for yield enhancement or power optimization.

978-3-9819263-0-9/DATE18/©2018 EDAA

TABLE I. ERROR OF SDD AND PDLY VERSUS STA

Benchmark	SDD50	SDD500	PDLY100	PDLY1000	PDLY10000	Benchmark	SDD50	SDD500	PDLY100	PDLY1000	PDLY10000
b01	0.79%	0.79%	0.33%	0.33%	0.33%	b15	2.77%	1.56%	2.46%	0.80%	0.80%
b02	4.33%	4.33%	0.11%	0.11%	0.11%	b15 1	3.60%	1.08%	3.25%	0.57%	0.57%
b03	4.12%	4.12%	4.05%	4.05%	4.05%	b17	3.43%	1.32%	3.69%	2.31%	1.82%
b04	1.70%	1.70%	1.70%	1.70%	1.70%	b17 1	4.37%	3.18%	4.99%	1.89%	1.89%
b05	1.21%	1.21%	1.21%	1.21%	1.21%	b18	5.00%	4.86%	10.54%	0.14%	0.46%
b06	4.36%	4.36%	3.64%	3.64%	3.64%	b18 1	8.13%	6.92%	7.96%	4.02%	4.03%
b07	5.21%	5.21%	2.20%	2.20%	2.20%	b19	11.77%	11.16%	12.35%	5.30%	4.58%
b08	2.84%	2.84%	1.95%	1.95%	1.95%	b19 1	8.83%	8.82%	8.82%	8.76%	8.70%
b09	7.42%	7.42%	7.50%	7.50%	7.50%	b20	8.04%	3.33%	11.69%	1.50%	1.50%
b10	0.18%	0.18%	0.05%	0.05%	0.05%	b20 1	9.75%	7.24%	12.36%	0.43%	0.43%
b11	0.20%	0.20%	0.20%	0.20%	0.20%	b21	7.03%	5.86%	8.56%	0.50%	0.50%
b12	1.75%	1.67%	1.82%	1.82%	1.82%	b21 1	2.47%	2.16%	4.45%	0.33%	0.33%
b13	2.35%	2.35%	2.35%	2.35%	2.35%	b22	6.34%	5.07%	10.29%	0.34%	0.17%
b14	12.16%	6.52%	16.35%	0.23%	0.23%	b22 1	8.44%	5.65%	12.16%	10.90%	0.41%
b14 1	10.15%	3.77%	13.35%	0.22%	0.22%	Average	5.13%	3.96%	5.87%	2.25%	1.85%

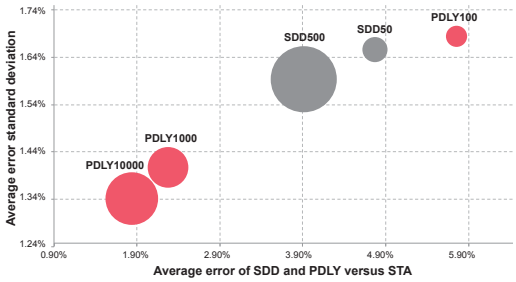

Fig. 2. Average error vs average standard deviation of error for 50 and 500 SDD pattern sets, and 100, 1000 and 10000 PDLY targeted paths; the size of the bubble represents the size of the pattern set used

III. EVALUATION

In this subsection we explore if SDD and PDLY tests correlate with the actual frequency of the circuits. We use low-power 28nm FD-SOI libraries to compare SDD and PDLY maximum frequency versus the critical paths of ISCAS'99 benchmarks using SYNOPSYS tools. We used Design Compiler in topographical mode for physical synthesis, Primetime for static timing analysis (STA), Tetramax for automatic test pattern generation (ATPG), and Vcs for back annotated simulation. Since functional pattens are not available for ISCAS'99 benchmarks, we use STA instead as a reference for comparison versus SDD and PDLY frequencies. This choice can be justified by noting that STA represents worst case circuit performance, which can be used as a lower bound for maximum circuit frequency. On the other hand, 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. This especially true since identifying the most critical part of the application is not possible in most cases.

We compared the maximum frequency at which each test pattern can be performed for each benchmark versus STA results. Table I presents the differences, what we call error, between each delay test frequency and STA for all benchmarks. We generated the results for 5 pattern sets including 50 and 500 SDD, and 100, 1000, and 10000 PDLY patterns. As it can be seen in this table, with increasing pattern count, the error is reduced. Increasing SDD pattern count from 50 to 500, achieves an error as low as 3.96%. Furthermore, increasing PDLY targeted path count from 100 to 10000 improves the error down to 1.85%. Therefore, depending on the time invested on testing during production, the accuracy of performance prediction can be improved.

The measured error for SDD and PDLY means that in order to make sure the performance prediction is accurate enough, a margin should be added on top of the estimated performance. If the inaccuracy of performance prediction is predictable, it is possible to come up with a safe margin. Fig. 2 illustrates the average standard deviation of the error plotted versus the average error measured using SDD and PDLY for all the circuits in the ISCAS'99 benchmarks. The plotted measurements are represented by the size of the test pattern set (reflected by the size of the circle in the plot). The figure shows that the larger the size of the used test pattern set, the more predictable the performance prediction will be. Therefore, depending on the time invested on testing during production, the accuracy of performance prediction can be improved. PDLY100 shows the worst prediction among all test pattern sets, which means that the size of this pattern set is not enough for performance prediction purposes. More test patterns should be taken into account to increase prediction accuracy as PDLY10000 shows the best performance prediction among all test pattern sets. In general, we can conclude that path delay test patterns are more suitable for performance prediction, however, sufficient pattern set sizes should be taken into account.

IV. CONCLUSIONS

In this paper, we proposed an innovative test flow for adaptive voltage scaling using delay testing during production. Since delay testing is performed to explicitly test for actual chip performance, the expensive effort of correlating PMB responses to chip performance during the characterization stage of manufacturing is not needed anymore, which reduces the cost and time to market dramatically. We compared two approaches (SDD or PDLY) based on their accuracy as performance predictor and how many pattern counts is sufficient for accurate voltage adaptation. We performed simulations using industrial grade 28nm FD-SOI library for 2 SDD pattern sets including 50 and 500 patterns, and 100, 1000, and 10000 PDLY targeted paths. The results show that Increasing SDD pattern count from 50 to 500, achieving an error as low as 3.96%. Furthermore, increasing PDLY targeted path count from 100 to 10000 improves the error down to 1.85%.

REFERENCES

[1] B. Kruseman, A. Majhi, and G. Gronthoud, On Performance Testing with Path Delay Patterns, in VTS, 2007.
[2] N. B. Zain Ali, et al., *Dynamic Voltage Scaling Aware Delay Fault Testing*, in ETS 2006.
[3] M. Sauer, et al., *On the Quality of Test Vectors for Post-Silicon Characterization*, in ETS 2012.
[4] P. Das, et al., *On Generating Vectors for Accurate Post-Silicon Delay Characterization*, in ATS 2011.

6

IMPACT OF TECHNOLOGY SCALING ON DELAY TESTING FOR LOW-COST AVS

SUMMARY
In this chapter, we compare three types of delay test patterns, namely TF, SDD, and PDLY test patterns in terms of effectiveness for performance estimation during production. Based on the simulation results of ISCAS'99 benchmarks with 28 nm FD-SOI library, using delay test patterns result in an error of 5.33% for TF testing, error of 3.96% for SDD testing, and an error as low as 1.85% using PDLY testing. Accordingly, PDLY patterns have the capacity to achieve the lowest error in performance estimation, followed by SDD patterns and finally TF patterns.

In addition, we also investigate the impact of technology scaling on the accuracy of delay testing for performance estimation during production. The results show that the 65 nm technology node exhibits the same trends identified for the 28 nm technology node, namely that PDLY is the most accurate, while TF is the least accurate performance estimator.

This chapter is based on the following paper.

Zandrahimi, M.; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Impact of Technology Scaling on Delay Testing for Low-Cost AVS*, submitted to the Journal of Electronic Testing.

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

Mahroo Zandrahimi · Philippe Debaud · Armand Castillejo · Zaid Al-Ars

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, error of 3.96% for small delay defect testing, and an error as low as 1.85% using path delay testing. 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

Mahroo Zandrahimi
Delft University of Technology, Mekelweg 4, 2628CD, Delft, Netherlands
E-mail: m.zandrahimi@tudelft.nl

Philippe Debaud
STMicroelectronics, 12 Rue Jules Horowitz, 38019 Grenoble, France
E-mail: philippe.debaud@st.com

Armand Castillejo
STMicroelectronics, 12 Rue Jules Horowitz, 38019 Grenoble, France
E-mail: armand.castillejo@st.com

Zaid Al-Ars
Delft University of Technology, Mekelweg 4, 2628CD, Delft, Netherlands
E-mail: z.al-ars@tudelft.nl

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 [1]. 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

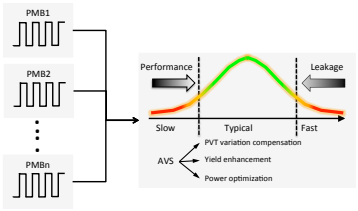

Fig. 1 Implementation of AVS power optimization using PMBs

optimal voltage is adapted exclusively to each operating point of each manufactured chip. PMBs 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 [2–7]. 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 [1].

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 [8] and [9], propose techniques for generating optimal set of delay test patterns during the characterization process, which guarantees 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 to expose systematic timing errors that are likely to affect a large fraction of manufactured chips so it 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 [10] 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 [12] 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 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. [14] 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. [13] 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 transition fault testing (TF), path delay testing (PDLY), and single delay defect testing (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. Online AVS approaches show better power efficiency, however, from industrial point of view, these approaches are considered very intrusive and risky. we back up this statement with the following reasons. First, in terms of planning and implementation effort, the fact that an extensive modification in hardware is needed makes the implementation of online AVS approaches very expensive. Moreover, for some sensitive parts of the design, such as CPU and GPU, which should operate at high frequencies, implementing online AVS approaches is quite risky since it affects planning, routing, timing convergence, area, and time to market. In contrast, 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 intra-die variations. Consequently, offline AVS approaches seem more manageable due to the fact that they can be considered as an incremental solution for existing devices and the amount of hardware modification imposed to the design is very low.

As discussed earlier, offline AVS techniques which are currently being used 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 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 Figure 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 Figure 2(3)).

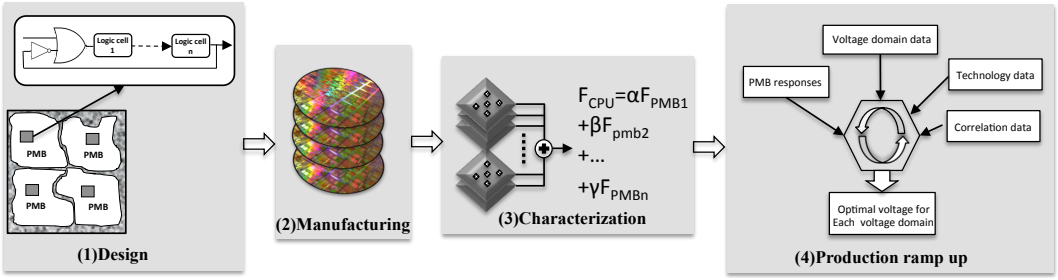

Fig. 2 AVS implementation in different levels of the design and manufacturing process

– **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.
– **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.
– **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.
– **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 [15], 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

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.

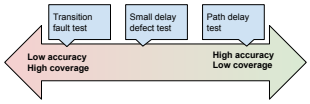

Fig. 3 Tradeoff in accuracy and coverage between different types of delay testing types

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 Figure 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. However, automatic test pattern generation (ATPG) algorithms are based on heuristics like SCOAP [17], 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 [18]. 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 (TF), small delay defects (SDD) and path delay testing (PDLY) [16]. As shown in Figure 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.

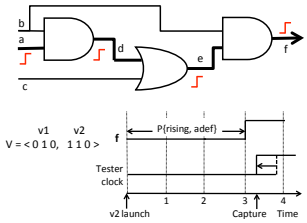

Fig. 4 An example of performance prediction using path delay testing

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{rising, adef} in this figure (the highlighted path) is one of the critical paths of the circuit reported by STA. The path delay test pattern needed to propagate the rising transition from input a to output f is the vector pair $V =< 010, 110 >$. 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 that 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.

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 (Vmin), 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

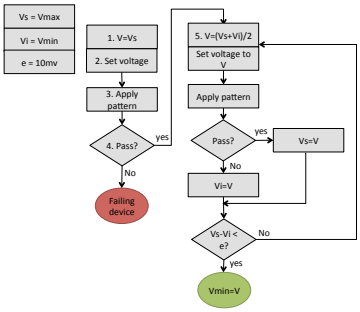

Fig. 5 Proposed flow to identify AVS voltages using delay testing

2. Set supply voltage to Vmax and wait for stabilization
3. Apply delay fault at speed test
4. If the chip fails the test, discard it, otherwise,
5. compute new values and do a binary search to find Vmin.

Conversion from Vmin to Fmax might be required depending on either performance estimation is done for yield enhancement or power optimization. "e" is an arbitrary value which is up to 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 [20]. 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. In order to investigate if such correlation exits 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 [21] libraries to compare the delay fault maximum frequency versus the critical paths of ISCAS'99 benchmarks [22]

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

using SYNOPSYS tools [23]. 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, Primetime 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. Also, we note that identifying the most critical part of the application is not possible in most cases.

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 FD-SOI library to extract the critical timing of bench-

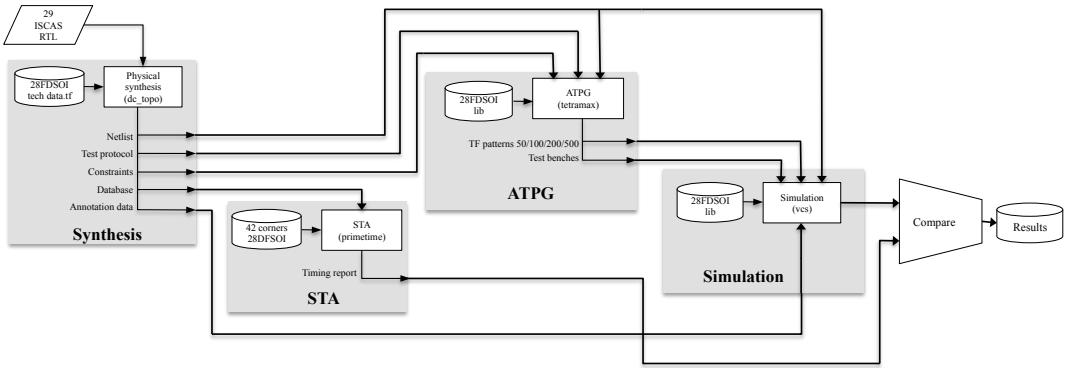

Fig. 6 Simulation flow for comparing delay testing frequency vs. STA for the 29 ISCAS'99 circuits

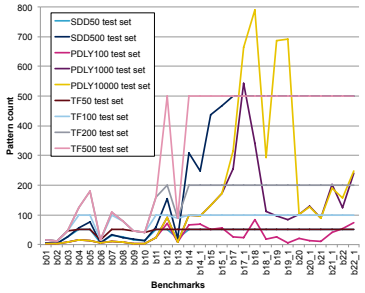

Fig. 7 Number of test patterns generated for each ISCAS'99 design targeting TFs, SDDs and PDLYs

marks 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. 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 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:

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

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:

$$error = \max_{all\ corners} [error_{corner}] \quad (2)$$

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

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

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

Table 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, 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 un-

Table 2 Error and standard deviation (SD) of error for TF versus STA

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%**

Table 3 Error and SD of error for SDD versus STA

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%**

certainty, 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.

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.

Figure 9 shows the average SD_{error} plotted versus the average *error* measured for the two technology nodes using each pattern set for all the circuits in the ISCAS'99 benchmarks. The size of each plotted measurement circle in the figure 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-

Table 4 Error and SD of error for PDLY versus STA

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%**

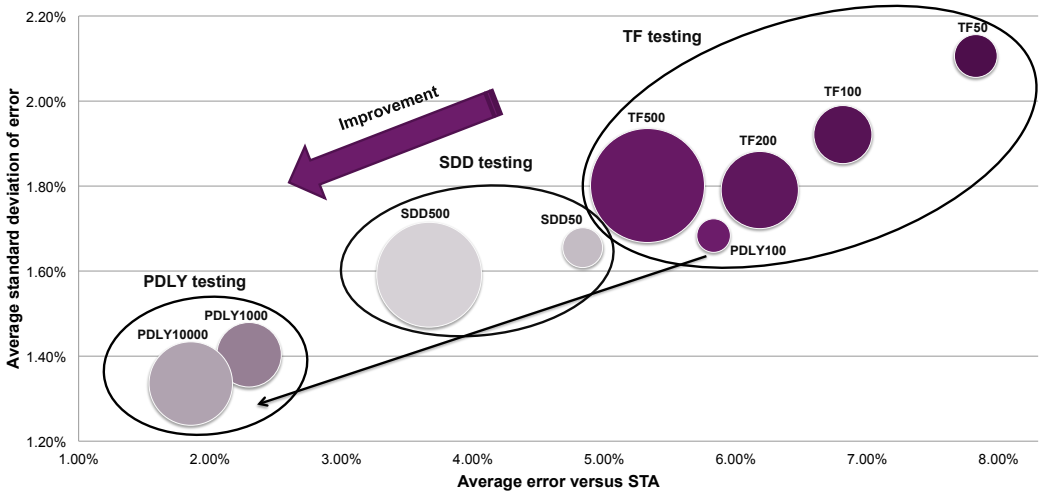

Fig. 8 Average error vs average standard deviation of error for all different test pattern types and test set sizes, in 28nm technology node. TF testing with 50, 100, 200, and 500 pattern sets, SDD testing with 50 and 500 pattern sets, and PDLY testing with 100, 1000, and 10000 pattern sets. The size of the bubble represents the average size of the pattern set used for all benchmarks.

ure 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, 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 Conclusions

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 representating 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.

References

1. M. Zandrahimi and Z. Al-Ars, *A Survey on Low-power Techniques for Single and Multicore Systems*, in ICCASA, pp. 69-74, 2014.
2. T. Chan and A.B. Kahng, *Tunable Sensors for Process-Aware Voltage Scaling*, in ICCAD, pp. 7-14, 2012.

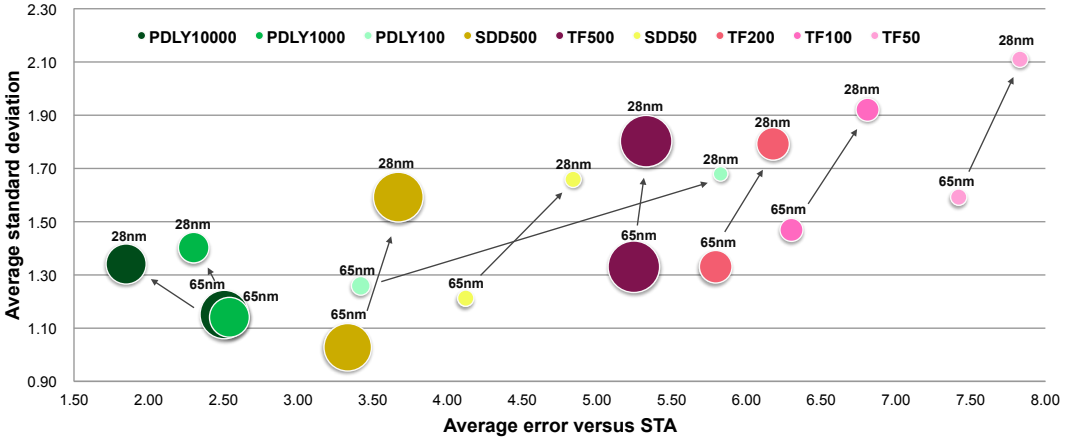

Fig. 9 Impact of technology scaling on average error and standard deviation of different delay test approaches for 65nm and 28nm

3. T. Chan, et al., *DDRO: A Novel Performance Monitoring Methodology Based on Design-Dependent Ring Oscillators*, in ISQED, pp. 633-640, 2012.
4. A. Drake, et al., *A Distributed Critical-Path Timing Monitor for a 65nm High-Performance Microprocessor*, in ISSCC, pp. 398-399, 2007.
5. TD. Burd, et al., *A dynamic voltage scaled microprocessor system*, in ISSCC, pp. 294-295, 2000.
6. J. Kim and M.A. Horowitz, *An efficient digital sliding controller for adaptive power-supply regulation*, in IJSSC, vol. 37, no. 5, pp. 639-647, 2002.
7. Q. Liu and S.S. Sapatnekar, *Capturing Post-Silicon Variations Using a Representative Critical Path*, in TCAD, vol. 29, no. 2, pp. 211-222, 2010.
8. M. Sauer, et al., *On the Quality of Test Vectors for Post-Silicon Characterization*, in ETS 2012.
9. P. Das, et al., *On Generating Vectors for Accurate Post-Silicon Delay Characterization*, in ATS 2011.
10. J.B. Brockman and S.W. Director, *Predictive Subset Testing: Optimizing IC Parametric Performance Testing for Quality, Cost, and Yield*, IEEE trans. on Semiconductor Manufacturing, 1989.
11. J. Lee, et al., *IC Performance Prediction for Test Cost Reduction*, IEEE International Symposium on Semiconductor Manufacturing Conference, 1999.
12. G. Li Zhang, et al., *EffiTest: Efficient Delay Test and Statistical Prediction for Configuring Post-silicon Tunable Buffers*, in DAC 2016.
13. N. B. Zain Ali, et al., *Dynamic Voltage Scaling Aware Delay Fault Testing*, in ETS 2006.
14. K. N. Shim and J. Hu, *A low Overhead Built-In Delay Testing with Voltage and Frequency Adaptation for Variation Resilience*, in DFT 2012.
15. M. Zandrahimi, Z. Al-Ars, P. Debaud, and A. Castillejo, *Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation*, in DATE, 2016.
16. M. Tehranipoor et al., Test and Diagnosis for Small-Delay Defects, in Springer Science+Business Media, LLC, 2011.
17. L.H. Goldstein, E.L. Thigpen, SCOAP: Sandia Controllability/Observability Analysis Program, in DAC, 1980.
18. B. Kruseman, A. Majhi, and G. Gronthoud, On Performance Testing with Path Delay Patterns, in VTS, 2007.
19. M. Zandrahimi, P. Debaud, A. Castillejo, and Z. Al-Ars, *Using Transition Fault Test Patterns for Cost Effective Offline Performance Estimation*, in DTIS, 2017.
20. P. Pant and E. Skeels, *Hardware Hooks for Transition Scan Characterization*, ITC 2011.
21. http://www.st.com/content/st_com/en/about/innovation—technology/FD-SOI.html
22. http://www.cad.polito.it/downloads/tools/itc99.html
23. http://www.synopsys.com/tools/pages/default.aspx

7

SUMMARY AND CONCLUSIONS

In this thesis, we introduced the idea of using various types of delay test patterns for offline AVS as an alternative for PMBs. We showed that our proposed techniques improve the effectiveness of voltage estimation, reduce cost and time to market significantly, and unlike PMBs, can be used for general logic as well. Besides that, we also presented an analysis of the impact of technology scaling on the effectiveness of delay testing approaches using two technology nodes: 28 nm and 65 nm. Below, we discuss the conclusions that we drew from Chapters 2 to 6.

CHAPTER 2

In this chapter, we gave a survey of different low power methods for single and multicore systems. Based on the system model we proposed, we presented a classification of power reduction techniques. Three main classes have been discussed: techniques which aim to reduce power either during fabrication/design or runtime in multicore tiles, runtime power reduction techniques for interconnects, and adaptive voltage scaling techniques to dynamically manage power during run-time. In addition, a number of design and manufacturing issues (such as process and temperature variations) have been taken into consideration. This chapter also discussed a number of examples for each of the classes and presented the published power reduction numbers reported by their respective papers. A summary of these numbers has been listed along with the trade-offs in performance and/or area overhead incurred as a result. The main conclusions are the following.

1. As technology scaling enters the nanometer regime, CMOS devices are facing many problems such as increased leakage currents, large parameter variations, as well as low reliability and yield. Therefore, in order to avoid encountering a stall in future growth of computing performance, high performance microprocessors had to enter the multicore era.

2. The growth in the number of cores causes super-linear growth in non-core area and power; accordingly, the power dissipation problem did not disappear in the

new multicore era. Therefore, we still need research and development of much more power-efficient computing systems at various levels of abstraction.

3. With the ongoing scaling of CMOS technologies, variations in process, supply voltage, and temperature (PVT) have become a serious concern in integrated circuit design. Therefore, an individual safety margin for each variation source is added on top of the supply voltage to make sure the device works properly in all process and environmental variations. However, this classical worst-case analysis is quite pessimistic and leads to wasting of both power as well performance. To overcome this problem, various adaptive design strategies have been proposed.

CHAPTER 3

In this chapter, we discussed two main categories of performance monitoring methodologies; direct and indirect measurement approaches. We compared them with each other in terms of accuracy, tuning effort, impact on design planning, and implementation risk. Then we focused on performance monitoring methodologies, which measure operation parameters indirectly during production. The challenges that these monitoring methodologies face with decreasing node sizes, in terms of accuracy and effectiveness have been discussed. The following conclusions can be drawn from this chapter.

1. According to the application, we can decide which performance monitoring technique is specifically suitable for the design being investigated. For example, for medical applications accuracy and power efficiency are far more important than the amount of hardware modification and planing effort. Thus, direct measurement approaches, which estimate the operation parameters by monitoring actual critical paths of the circuit during run-time are considered more suitable.

2. For nomadic applications, such as mobile phones, tablets, and gaming consoles, cost and design customization effort are considered more significant than accuracy and effectiveness. Thus, indirect measurement performance monitoring approaches are considered more manageable for these devices. Despite the accuracy and effectiveness of direct measurement performance monitoring approaches, cost versus benefit is not proven since the implementation risk and the impact on design planning is high.

3. In deep sub-micron technologies, indirect measurement approaches are showing limitations to accurately estimate silicon performance, which leads to unnecessary power loss. Based on simulation results on ISCAS'99 benchmarks as well as static timing analysis of a nanometric FD-SOI device, we showed that depending on the design, critical path can change dramatically as a result of PVT variations. Thus, we can conclude that the accuracy and effectiveness of indirect measurement approaches is low. Silicon measurements of the same device show that the required design margin is above 10% of the clock cycle leading to unacceptable waste of power.

CHAPTER 4

In this chapter, we introduced the new concept of using transition fault test patterns for AVS during production, which serves as an alternative to the PMB-based AVS. This approach represents a powerful example of value-added testing, in which TF tests (already used during production) can replace a long and expensive process of PMB characterization, reducing cost and time to market dramatically. Moreover, since transition fault test patterns target all gates and indirectly cover all path-segments, it is a better performance representative than PMBs. As functional patterns are not used anymore, testing approaches could be a solution for general logic, not only for CPU and GPU. The following conclusions can be drawn from this chapter.

1. A case study on real silicon comparing the performance estimation using functional test patterns and the TF-based approach on a 28 nm FD-SOI CPU shows a very close correlation between TF test patterns and functional patterns.

2. A case study on real silicon comparing the accuracy of voltage estimation using PMBs and the TF-based approach on a 28 nm FD-SOI device shows that the PMB approach can only account for 85% of the uncertainty in voltage measurements, while the TF-based approach can account for 99% of that uncertainty.

3. According to simulation results of the 29 ISCAS'99 chips on 42 corners of a 28 nm FD-SOI library, using TF testing for performance estimation ends up with an inaccuracy as low as 5.33% and a standard deviation of 1.8%.

CHAPTER 5

In this chapter, we presented an innovative test flow for adaptive voltage scaling using SDD as well as PDLY test patterns during production. We compared these two approaches based on their accuracy as performance predictors and the number of patterns needed for a sufficiently accurate voltage adaptation. The following conclusions can be drawn from this chapter.

1. Based on the simulations on ISCAS'99 benchmarks using industrial grade 28 nm FD-SOI library for 2 SDD pattern sets including 50 and 500 patterns, increasing SDD pattern count from 50 to 500, achieves an error as low as 3.96%.

2. The same simulation process for 3 PDLY pattern sets including 100, 1000, and 10000 patterns shows that increasing PDLY targeted path count from 100 to 10000 improves the error down to 1.85%.

3. 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.

CHAPTER 6

In this chapter, we compared 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. We also presented an analysis of the impact of technology

scaling on the effectiveness of delay testing approaches using two technology nodes: 28 nm and 65 nm. The following conclusions can be drawn from the chapter.

1. 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.

2. 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.

3. Based on the results of our analysis of the impact of technology scaling, 65 nm technology node exhibits the same trends identified for the 28 nm technology node, namely that PDLY is the most accurate performance estimation method, while TF is the least accurate performance estimator.

LIST OF PUBLICATIONS

INTERNATIONAL JOURNALS

1. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Impact of Technology Scaling on Delay Testing for Low-Cost AVS*, submitted to the Journal of Electronic Testing.

2. **Zandrahimi, M.**; Zarandi, H. R.; Mottaghi, M. H., *Two Effective Methods to Detect Anomalies in Embedded Systems*, January 2012, Microelectronics Journal, volume 43, issue 1.

INTERNATIONAL CONFERENCES

1. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Cost Effective Adaptive Voltage Scaling Using Path Delay Fault Testing*, East-West Design & Test Symposium (EWDTS 2018), 14-17 September, Kazan, Russia.

2. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *An Industrial Case Study of Low Cost Adaptive Voltage Scaling Using Delay Test Patterns*, Design, Automation and Test in Europe (DATE 2018), 19-23 March 2018, Dresden, Germany.

3. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Industrial Evaluation of Transition Fault Testing for Cost Effective Offline Adaptive Voltage Scaling*, Design, Automation and Test in Europe (DATE 2018), 19-23 March 2018, Dresden, Germany.

4. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Transition Fault Testing for Offline Adaptive Voltage Scaling*, International Test Conference (ITC 2017), 31 October - 2 November 2017, Fort Worth, USA.

5. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Using Transition FaultTest Patterns for Cost Effective Offline Performance Estimation*, 12th International Conference on Design Technology of Integrated Systems in Nanoscale Era (DTIS 2017), 4-6 April 2017, Palma de Mallorca, Spain.

6. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Industrial Approaches for Performance Evaluation Using On-Chip Monitors*, 11th IEEE International Design Test Symposium (IDT 2016), 18-20 December 2016, Hammamet, Tunisia.

7. **Zandrahimi, M.**; Debaud, P.; Castillejo, A.; Al-Ars, Z., *Challenges of Using On-Chip Performance Monitors for Process and Environmental Variation Compensation*, Design, Automation and Test in Europe (DATE 2016), 14-18 March 2016, Dresden, Germany.

8. **Zandrahimi, M.**; Al-Ars, Z., *A Survey on Power Low-Power for Single and Multicore Systems*, International Conference on Context-Aware Systems and Applications (ICCASA 2014), 15-16 October 2014, Dubai, United Arab Emirates.

9. **Zandrahimi, M.**; Zarandi, H. R.; Zarei, A., *A Cache-based Anomaly Detector for Embedded Systems*, Real-Time and Embedded Systems Conference (RTES 2010), 1 November 2010, Singapore.

10. **Zandrahimi, M.**; Zarandi, H. R.; Zarei, A., *A Probabilistic Method to Detect Anomalies in Embedded Systems*, 25th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT 2010), 6-8 October 2010, Kyoto, Japan.

11. **Zandrahimi, M.**; Zarandi, H. R.; Rohani, A., *An analysis of fault effects and propagations in ZPU: The world's smallest 32 bit CPU*,2nd Asia Symposium on Quality Electronic Design (ASQED 2010), 3-4 August 2010, Penang, Malaysia.

12. Rohani, A.; Zarandi, H. R.; **Zandrahimi, M.** , *New Switch Box Architecture for SEU Detection in SRAM-Based FPGAs*, 2nd International Conference on Computer Science and its Applications (CSA 2009), 10-12 December 2009, Jeju, South Korea.

CURRICULUM VITÆ

Mahroo ZANDRAHIMI

Mahroo Zandrahimi was born in Birmingham, UK in 1985. She received her Bachelors in Computer Hardware Engineering at Shahid Beheshti University of Iran, and her Masters in Computer Architecture at Amirkabir University of Technology, both with honors. In April 2013, she started her PhD at the Quantum and Computer Engineering Department of the Delft University of Technology, where she was working under the supervision of Dr. Zaid Al-Ars. She performed her research in collaboration with STMicroelectronics in Grenoble, France, where she was based for a period of 2 years. 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.