Software-based Infield Wearout Monitoring for Synchronous Digital Chips

Bardia Zandian and Murali Annavaram
Electrical Engineering Department, University of Southern California
Los Angeles, CA, USA
{bzandian,annavara}@usc.edu

I. INTRODUCTION

A. Background and Motivation

Electronic circuits of all sizes are increasingly used in every aspect of our lives and our dependence on their reliable operation has significantly increased in the past three decades. A mere 7 to 10 year life expectancy, conventionally guaranteed for integrated circuits used in consumer electronics, is not going to be enough for meeting demands of future widespread applications for integrated circuit. In the past decade, reliability and expected lifetime of circuits manufactured using nano-scale fabrication technologies have been decreasing. Efforts to continue with CMOS scaling while improving performance and controlling power consumption have resulted in decreased reliability and lifetime of circuits fabricated [1-3]. Increase in transistor density, higher operation temperature, higher current density, and higher electric field stress all result in reduced lifetime reliability. Infield wearout or aging of circuits is a result of multiple electro-physical phenomena which change the electrical properties of devices and interconnects in the circuit. Electromigration, Negative Bias Temperature Instability (NBTI), Positive Bias Temperature Instability (PBTI), Hot Carrier Injection (HCI), Time-dependent Dielectric Breakdown (TDDB), stress migration, and thermal cycling are the main contributors to circuit wearout and are happening at a higher rate in smaller nano-scale technologies [3, 4]. Initial manifestation of wearout is gradual timing degradation of transistors and interconnects. Timing guardbands are added to synchronous digital circuits at design time in anticipation of this degradation of signal paths. Excessive levels of wearout result in hard faults in the circuits due the device or interconnect failure (e.g. stuck-at-0 or stuck–at-1). Both the trend in demand for a longer expected lifetime of electronics and the trend of higher rates of wearout in more scaled technologies would require designer to increase, already high, frequency and voltage guardbands. Increased guardbanding results in lost opportunities for higher performance and power efficiency.

In presence of this high amount of unreliability many fault tolerance frameworks have been proposed to be used in addition to implementation of conservative guardbands. Conventional fault tolerance solutions detect a hard error and diagnose the source of the error and then disable the faulty structures [5, 6]. In the case of wearout there is a more effective approach than conventional error detection and that is wearout monitoring followed by preventive error avoidance. Wearout state of the system can be monitored at runtime and error avoidance techniques can be used to postpone unset of these failures. Only when all preventive measures to postpone wearout induced failures have been exhausted then the system will fall back on error tolerance.

This work was supported by NSF grants CAREER-0954211, CCF-0834798.

B. Software-based Wearout Monitoring

Our proposed framework uses software to measure the amount of wearout of a chip during its infield operation. This framework does not require any hardware modifications to the fabricated integrated circuit and only uses software to initiate wearout measurement tests to collect information regarding the wearout state of different parts of the chip. This would enable use of this framework with many of the chips already fabricated and being used infield without the need for hardware modifications. Hardware-based frameworks for wearout monitoring, such as [7, 8], require changes to the way the circuit is implemented and are not usable with chips already fabricated. Furthermore, for many small application specific integrated circuits (ASIC) the overhead of hardware-based wearout monitoring enhancements are too high. This software-based framework can be used for small synchronous digital chips which don't have internal memory elements. Small digital signal processing (DSP) chips, microcontrollers, and microprocessors are examples of this category of chips.

The chip being monitored is periodically switched into a test mode in which instead of executing normal user inputs it would run wearout measurement tests. During a test phase the chip is operated at an elevated clock frequency while a group of inputs are fed into the chip. The test inputs are part of a test program running on the chip and the adjustment to the operation frequency during the test phase can be either done by utilizing already in place variable frequency and voltage knobs used for Dynamic Voltage and Frequency Scaling (DVFS) or can be provided by adding circuit board level frequency/voltage controls. The test frequency at which each of the input patterns fails to produce correct expected output is logged. As the chip ages more the software keeps track of the amount of degradation the paths being tested are suffering from. This is the gradual increase in timing delay of the circuit paths tested which would be observed as failure of test inputs at lower frequencies from the test frequency range. When the wearout level of any path being tested gets close to a preset threshold (set based on the amount of guardband the chip has), a red flag is triggered, marking critical levels of wearout. This notification regarding critical level of wearout can be either used to signal end users that their chip requires replacement in the near future or can be

used by automated adaptive frameworks to change how the chip is used (e.g. reduce its operation frequency or increase its operation voltage). We believe that this proactive framework of dealing with wearout has significant advantages compared to conventional solution which wait for critical levels of wearout to cause errors and only then react to the critical wearout state. Although the overhead of testing using software is higher than hardware-based frameworks, the software-based approach can be easily applied to chips which don't have built-in wearout monitoring hardware enhancements (i.e. a large percentage of chip currently in use or will be fabricated in the future).

II. FRAMEWORK

This wearout measurement framework measures initial manifestation of wearout which is increase in the delay of circuit timing paths. A group of inputs are fed into the chip which is operated at its nominal operation frequency (f_0) and supply voltage (V_0). The signal outputs from this new chip (i.e. not wornout) are recorded and labeled as the correct expected output from execution of the workload. This workload used for testing is selected so that it would sensitize the most timing critical paths in the chip. Details of test workload selection are described later.

After the first round of testing, the same chip is exercised using the same inputs but is now clocked at a frequency, f_1, which is slightly higher than its nominal operation frequency ($f_1>f_0$) while supply voltage is kept at nominal V_0 value. The output from this chip which is operating at an elevated operation frequency f_1 is compared to the correct expected output. Output values not matching are related to inputs which sensitized longer circuit paths and have failed due to timing violation during operation at above nominal frequency. This type of testing would be repeated with different frequencies ($f_i>\ldots>f_2>f_1>f_0$) to provide a fine-grain measure of delay of each of the circuit paths sensitized.

The same chip can be tested again after it has suffered from more wearout (i.e. after actual in-field usage). The operation frequency at which a specific output fails can provide a measure of the amount of wearout the chip has suffered from. For example, if an input resulted in the correct output at f_0, f_1 ,f_2 , and f_3 but failed at f_4, this tells us that the delay of the path exercised by this input is approximately Delay($@t_1$)=$1/f_3$ at time of the test (t_1). Now let's assume the chip is exposed to wearout and then testing is done at time (t_2) and the same input results in a correct output only at f_0, f_1 but fails at f_2. This tells us that the path exercised has a delay of approximately Delay($@t_2$)=$1/f_1$. Amount of delay difference between Delay($@t_1$) and Delay($@t_2$) provides us with the amount wearout-induced timing degradation the path tested has suffered from between time t_1 and time t_2. It should be also noted that instead of increasing the operation frequency, a method which is based on reducing the operation voltage during the test phase (while frequency is kept at f_0) can also be used for this type of testing.

The slowest, most critical, paths in the chip are the first group of paths which are going to results in timing violations when the chip is critically wornout. Hence, measuring their wearout is required in order to accurately detect wearout level of a chip. Given the implementation details of the chip, timing analysis of the critical paths in the circuit can be performed and a list of test vector which are designed to sensitize the most critical paths of the circuit can be extracted the from gate-level netlist. These test vectors can be used for testing the chip with the framework explained earlier.

If the implementation details of the chip are not available, for reasons such as intellectual property or security restriction, a one-time post-fabrication exhaustive test of the chip can be done in order to provide the wearout monitoring software with a list of test inputs that would sensitize the slowest paths in a chip. This type of exhaustive testing will only be feasible for chips with a small number of input pins, which comprise most of the target market for this type of testing. This exhaustive test needs to be performed only once for each chip implementation (not for each replica of the chip manufactured). Fine-grain multi-frequency testing of chips with all possible input pairs to the chip are done in order to generate a list of the initial delay of all paths in the chip and the input pairs required to sensitize them. Then a list of input pairs which sensitize the most critical paths in the chip are picked to be used for the infield wearout testing.

III. CONCLUSIONS

A software-based framework to measure wearout of synchronous digital chips as they are being utilized by end users is proposed. First, a group of inputs to the chip, selected to sensitize the most critical paths of the circuit, are fed in while the chip is operating at nominal operation frequency and the correct expected outputs for these inputs are stored. Then clock frequency of the chip is increased and the same tests are repeated. Failure of a test to produce correct expected output at the elevated frequency will be used as an indication of the amount of wearout the circuit.

REFERENCES

[1] S. Borkar, "Designing reliable systems from unreliable components: The challenges of transistor variability and degradation," in IEEE Micro, 2005, pp. 10-16.

[2] S. Borkar, "Electronics beyond nano-scale CMOS," in Design Automation Conference, 2006, pp. 807-808.

[3] J. Hicks, D. Bergstrom, M. Hattendorf, J. Jopling, and J. Maiz, "45nm Transistor Reliability," Intel Technology Journal, vol. 12, pp. 131-144, 2008.

[4] C. Kenyon, A. Kornfeld, K. Kuhn, M. Liu, and A. Maheshwari, "Managing Process Variation in Intel's 45nm CMOS Technology," Intel Technology Journal, vol. 12, pp. 93-109, 2008.

[5] D. Sylvester, D. Blaauw, and E. Karl, "ElastIC: An adaptive self-healing architecture for unpredictable silicon," Ieee Design & Test of Computers, vol. 23, pp. 484-490, Nov-Dec 2006.

[6] J. Srinivasan, S. V. Adve, P. Bose, and J. A. Rivers, "Exploiting structural duplication for lifetime reliability enhancement," in 32nd International Symposium on Computer Architecture, Proceedings, 2005, pp. 520-531.

[7] B. Zandian, W. Dweik, S. H. Kang, T. Punihaole, and M. Annavaram, "WearMon: Reliability Monitoring Using Adaptive Critical Path Testing," in Dependable Systems and Networks, 2010, pp. 151-160.

[8] T. H. Kim, R. Persaud, and C. H. Kim, "Silicon odometer: An on-chip reliability monitor for measuring frequency degradation of digital circuits," Ieee Journal of Solid-State Circuits, vol. 43, pp. 874-880, Apr 2008.