A Temperature-Aware Time-Dependent Dielectric Breakdown Analysis Framework

Dimitris Bekiaris, Antonis Papanikolaou, Christos Papameletis, Dimitrios Soudris, George Economakos, and Kiamal Pekmestzi

Microprocessors and Digital Systems Lab, National Technical University of Athens
157 80, Zografou, Athens, Greece
{mpekiaris,antonis,xristos86,dsoudris,
geconom,pekmes}@microlab.ntua.gr

Abstract. The shrinking of interconnect width and thickness, due to technology scaling, along with the integration of low-k dielectrics, reveal novel reliability wear-out mechanisms, progressively affecting the performance of complex systems. These phenomena progressively deteriorate the electrical characteristics and therefore the delay of interconnects, leading to violations in timing-critical paths. This work estimates the timing impact of Time-Dependent Dielectric Breakdown (TDDB) between wires of the same layer, considering temperature variations. The proposed framework is evaluated on a Leon3 MP-SoC design, implemented at a 45nm CMOS technology. The results evaluate the system's performance drift due to TDDB, considering different physical implementation scenarios.

Keywords: Reliability, Time-Dependent Dielectric Breakdown, Inter-Metal Dielectric Leakage, Timing.

1 Introduction

The current trend of CMOS technology scaling aggressively reduces the physical dimensions of devices and interconnects leading simultaneously to contiguous effects, which form novel threats regarding the reliability of modern integrated circuits. The shrinking of channel length of transistors incurs an exponential growth of sub-threshold leakage, which increases power density and creates hot spots in congested areas of the chip. The reduction of gate oxide thickness in technology nodes beyond 65nm enhances the gate tunneling current, resulting in Negative-Bias Temperature Instability (NBTI) in PMOS transistors due to the gradual rise of threshold voltage.

Similar effects of a progressive impact also appear in interconnection structures. They are caused by the shrinking of geometrical dimensions and the saturation of the operating voltage at around 1V, in sub-micron technologies [1]. The reduction of wires width and thickness increases current density, while the smaller pitch and spacing enhances the electrical field between interconnects of the same metal layer. Thus, Back-End-of-Line (BEOL) reliability phenomena like Electro-Migration (EM), Stress Migration (SM) and Time-Dependent Dielectric Breakdown (TDDB) start to gain in

R. van Leuken and G. Sicard (Eds.): PATMOS 2010, LNCS 6448, pp. 73–83, 2011.
© Springer-Verlag Berlin Heidelberg 2011

significance with technology scaling and they progressively degrade the electrical characteristics and structure of affected interconnects.

The recent move from silica-based to porous, low-k dielectrics between copper lines in the interconnect stack comes along with the advent of nanoscale technologies and has further aggravated the potential TDDB problems. Copper tends to "leak" into the dielectrics and create conductive paths between wires of the same metal layer, leading to breakdowns in the dielectric and leakage current between wires. Moreover, the evolution of this leakage current is not abrupt. It seems to be a rather smooth function of operating time until the magnitude of the current is large enough to create an electrical short between wires which affects the functionality of the circuit.

In this paper, we present an analysis flow that can capture the impact of Time Dependent Dielectric Breakdown of the low-k dielectrics of the interconnect stack on the delay of the individual wires and, furthermore, propagate this impact to the timing of the entire chip. Hence, we can estimate when the chip will present timing violations due to reliability problems on the interconnects.

The rest of the paper starts by presenting the related work in the literature and continues with the model used for the TDDB estimations. Section 4 presents the proposed reliability analysis framework and Section 5 demonstrates the experimental results, based on the application of this framework on layouts of an MP-SoC platform. Finally, a discussion on the results and also hints for future work conclude the paper.

2 Related Work

Time-Dependent Dielectric Breakdown of the low-k dielectrics has been identified as a potential reliability threat by many independent researchers since the decision to move from aluminum to copper wires for standard CMOS processes [2][3][4]. Significant effort is being invested at the process technology development level, in order to determine the process steps and materials that can alleviate this phenomenon [5][6]. Up to now, however, no solution at the level of process technology seems to solve the problem completely. Hence, TDDB must be taken into account at the design stage as a potential threat, not only for the reliability of interconnects, but also for the circuit's performance, as the flow of inter-metal leakage through the dielectric increases the wire delay and possibly design's critical time delay drift over time.

This has been implicitly understood also by the process technology people, who have started working on modeling the impact of TDDB on the electrical properties of interconnects [7][8][9]. Although the design community has not yet taken up any of these models to evaluate the impact of TDDB at the level of an entire system, recent works present methodologies and tools estimating the system's performance drift over time [10][11], based on the extrapolation of accelerated inter-metal leakage measurements to normal, operating conditions. In this work, we take a step further on these approaches, by exploring the impact of different place-and-route styles on system's timing degradation due to TDDB, while considering the entire layout's temperature profile, which is of course dependent on the application.

3 Time-Dependent Dielectric Breakdown Mechanism

Time-Dependent Dielectric Breakdown (TDDB) of inter-metal dielectrics refers to the progressive destruction of the material insulating interconnects of the same metal layer, leading to the formation of "leaky" paths and therefore increasing the time required for charging and discharging of wire capacitances. This mechanism is similar to the ones appearing in gate oxide structures and parallel plate capacitors of high-k dielectrics, also used in DRAMs.

However, TDDB becomes more significant for interconnects with the advent of low-k porous dielectric materials, mostly used in the sub-micron manufacturing processes to reduce interconnect delay, while improving crosstalk and minimizing interconnect power dissipation. These gains come hand in hand with worse reliability characteristics, due to the porous nature of the specific type of dielectrics. The gradual breakdown of low-k materials is aggravated as far as the electric field between neighboring wires rises, wire pitch scales down and the operating voltage saturates around 1V [1]. Hence, the inter-metal electric field is growing stronger with technology scaling and comprises the main reason for the formation of conductive paths through the dielectric, along with imperfections appearing in the interconnects.

These defects appear in the low-k materials used in current nanometer technology processes and their formation is mainly due to the dominating dielectric deposition methods, performed during the manufacturing process. Therefore, considering, in accelerated conditions of voltage and temperature, an electric field lower than 6 MV/cm, which is a usual stress value for low-k metal-insulator-metal structures [7], free charges (holes) are trapped into the areas of the dielectric where these defects exist. The number of trapped holes rises progressively, until a critical value is reached. Then, the flow of inter-metal leakage becomes significantly stronger, leading to the dielectric's breakdown and finally resulting into a short-circuit.

The TDDB mechanism can be modeled either by the Schottky or by the Frenkel-Poole emission, both of which have similar mathematical expressions of inter-metal leakage current density [7] and are exponentially dependent on temperature. However, mainly because of the nature of the specific wear-out and of the recent shifting of the interconnect technology on low-k dielectrics, there is little convergence on a specific model. Therefore, a common practice for the estimation of inter-metal leakage in operating conditions deals with the extrapolation of leakage measurements from experimental data, where wires are stressed for a certain number of hours under high voltage and temperature, resulting in strong electric fields.

The extrapolation approach has been also adopted in this work, where the wires have been stressed for about one hour. The leakage in operating conditions is extracted by performing linear extrapolation from the experimental measurements and the derived values formed the basis for the estimation of the delay impact of TDDB on individual interconnects. In the proposed reliability analysis framework, presented in the following section, we demonstrate how we use the information from the inter-metal leakage characterization libraries in stress conditions, in order to guide the estimation of additional delay in wires due to TDDB. This was a necessary step for the development of the proposed reliability framework, which predicts the design's performance drift over time due to TDDB and therefore the shortening of system's operating lifetime, under the required performance.

4 The Proposed Interconnect Reliability Framework

The proposed reliability analysis flow, which captures the impact of TDDB in interconnects of low-k dielectrics on a design's timing, is illustrated in Fig. 1. Even though its structure is generic enough, we have customized this instance of flow to capture the impact of TDDB on the delay of interconnects.

The flow of Fig. 1 takes four main inputs: (i) the layout of the circuit which includes all the geometrical information of the interconnect stack, (ii) the timing constraints of the design, (iii) the standard-cell technology libraries, which include the information about the timing of the cells in the design's post-layout netlist and the dimensions of cells and interconnects, and (iv) the layout's power profile, which is needed to extract the temperature profile and establish the actual temperature on each net.

The first steps of the flow estimate the temperature and timing profile of the layout. For the temperature estimation, we used HotSpot [12], an open-source academic tool that produces the thermal map of the chip, by taking as inputs the floorplan of the target design and the power consumption of the floorplan's units. The power profile required for the temperature estimation is obtained via power analysis of the post-layout Verilog netlist in Synopsys PrimeTime PX, using an activity trace obtained through logic simulation, based on a testbench of a real application, in ModelSim.

Static Timing Analysis (STA) is performed on the design's post-layout Verilog netlist, using the SoC Encounter Timing System (ETS) tool, which finds the most timing-critical paths in the design. In our framework, we extract the nets from the 50 most timing-critical paths. These nets are the "key" interconnects, as they belong to timing paths susceptible to suffer from TDDB. These paths have a minimal slack (less than 2ns) and thus, a delay overhead due to TDDB may lead to timing violations.

After these nets are identified, their geometrical properties are extracted, including the dimensions of the wires themselves and of their neighbors, as well as the spacing between them. This is performed through a Tcl script, which is executed in the SoC Encounter's environment and reads the layout's database based on the SoC Encounter's Database Access [13] command set. Hence, the script extracts the wires of the nets for the examined critical path, as well as their length, width and thickness, and finds the neighboring wires of the same metal layer, along with their physical dimensions and the distance between them and the wires of the examined net. All this information, which will be used in the additional delay computation due to TDDB, is dumped to an output file, named as *wire.report* in our toolchain.

After extracting the physical information about the examined nets' wires, the next step is to estimate the impact that TDDB is expected to have on the delay of these wires individually, based on the model outlined in the previous section, and to annotate the generated delay overhead due to TDDB on the design's Standard Delay Format (SDF) file. Finally, the additional delay of each wire is taken into account in a chip-level timing analysis, in order to estimate the impact of TDDB on the timing of the entire layout, in a similar way as in the second step.

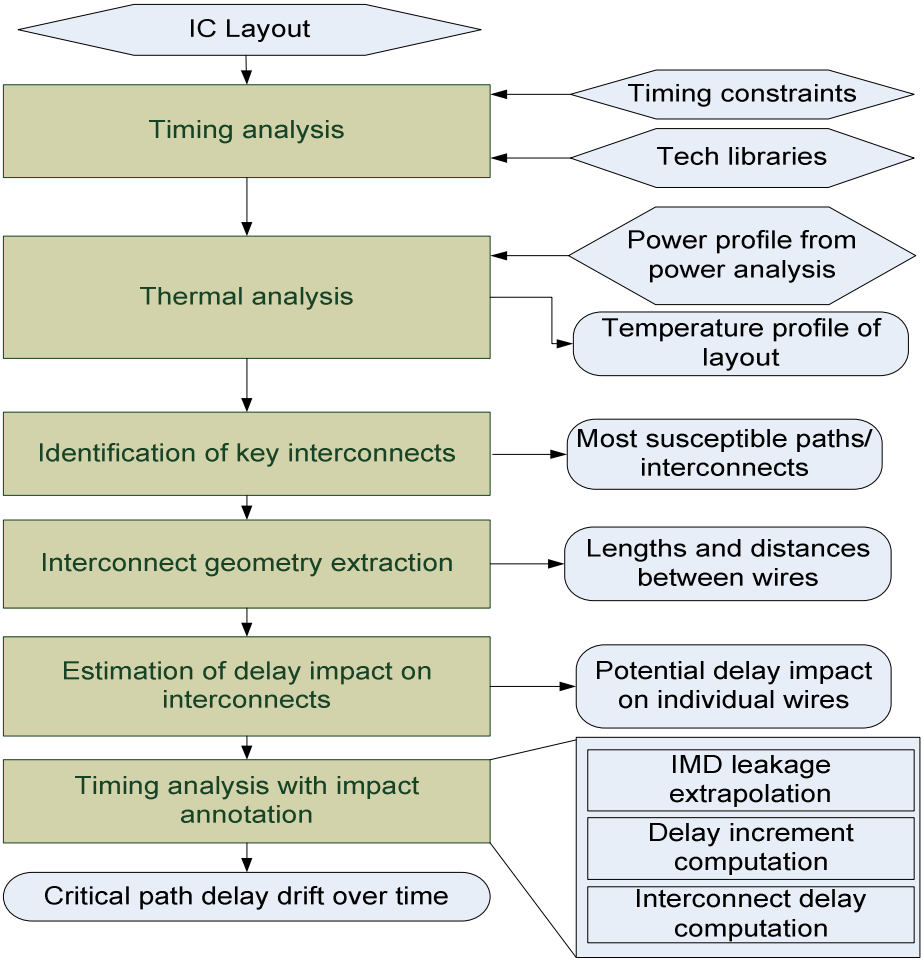

Fig. 1. The proposed temperature-aware interconnect reliability framework

4.1 Estimation of Delay Impact on Interconnects

For each of these wires identified in the 50 most timing-critical paths, our flow estimates the delay overhead due to TDDB, based on pre-computed inter-metal leakage (IMD) look-up table libraries, given in operating and accelerated conditions. This delay, computed for each of the nets of the examined path, is annotated to the Standard Delay Format (SDF) file of the design, to update the specific net delay with the new value. The computation of the additional delay due to TDDB is performed in three steps, as it is shown in Fig. 1. It is performed through a Matlab script, based on the information of wire extraction for the nets of the examined path, while taking into account the proper temperature, depending on the units from which the specific path comes through. The final SDF file, including the new net delays, is then back-annotated, along with the post-layout netlist, to the static timing analyzer of ETS, to

evaluate the impact of TDDB on the design's performance. The analytic description of the three steps required for the TDDB impact annotation is given below:

Step 1 – IMD Leakage Extrapolation: Based on the neighboring wire information, a Matlab script performs the additional delay overhead computation due to inter-metal leakage and annotates the shifted delay to the SDF file of the design, for the TDDB timing impact evaluation. The additional delay calculation is divided into two steps. At first, the script reads all the wires of the examined net from *wire.report*, as well as their neighboring wires, and obtains IMD leakage from accelerated to operating conditions by performing linear extrapolation, based on experimental look-up table libraries. These libraries contain IMD leakage information after having stressed the wires for up to one hour and in conditions of 35V, 40V and 45V of voltage, under temperatures of 323K, 398K and 448K respectively.

Step 2 – Delay Increment Computation: The extrapolated leakage is used to estimate the additional delay on the net due to TDDB, based on another look-up table library, which provides the delay increment ratio for charging or discharging a wire, depending on the inter-metal leakage between two adjacent wires of varying length, spacing and overlap. For the construction of such a library, we simulated the behavior of two neighboring wires in Synopsys HSPICE, in order to find the ratio of delay increment of charging and discharging a wire due to IMD leakage, for various possible adjacent wire patterns. This library was created once and it is used in all the conducted experiments, as long as the on-the-fly extraction and simulation of adjacent wire patterns for all the timing paths of each layout would be time-consuming.

In the conducted experiments, the wire length ranges between 10um and 600um, in order to include wire patterns with length equal or greater than those met in the layouts of our case study. The spacing's range is between 0.06um and 0.5um, covering the range defined in the design rules of the 45nm standard-cell library used for the implementation of the layouts. Moreover, in order to measure the delay of wires which are not totally overlapped, we simulated wire patterns where the starting point of the neighboring wire was not equal to the one of the wire for which the delay was measured. Hence, the neighboring wire's starting point was ranging from zero (total overlap of wires) to 75% of the target wire's length (smallest overlap of wires). In order to simulate the inter-metal leakage in HSPICE, we used current sources distributed across the target wire at each R-C (Resistance – Capacitance) segment.

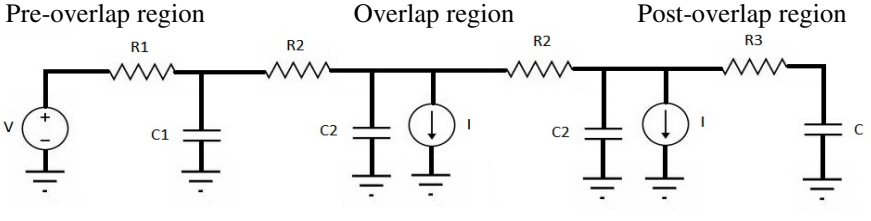

Fig. 2. The distributed RC model, simulating inter-metal leakage in HSPICE

The total leakage current for each wire in our simulations is dependent on the wire's length and varies between 0 and 50uA, in order to cover a wide enough range. The

value of each current source, given in uA, depends on the overlap length between the target wire and its adjacent one. In our approach, it is computed by dividing the total leakage current for each wire with the number of R-C stages corresponding to the overlap length between the target wire and the adjacent. In Fig. 2, we demonstrate an example of an equivalent distributed R-C model of a wire that has two of the four R-C stages overlapping with its neighboring one (Overlap region), at the same metal layer.

***Step 3 – Interconnect Delay computation*:** Thus, based on the delay increment ratios extracted from the simulations of wires, we constructed a look-up table library, including the wirelength of which the delay is computed, along with the neighboring wire's length, the wires' spacing and the starting point of the neighboring wire, all given in um. Based on this library, namely *TDDB_LUT.lib* in our flow, as well as on the extrapolated leakage from accelerated to operating conditions of the first step, we now performed a linear interpolation through Matlab, to compute the delay ratio for each wire of the examined net in the current timing path. It must be noted that only wires of length longer than 10um are considered in the additional delay calculation script, in correspondence with the range of wire lengths included in *TDDB_LUT.lib*.

In the linear extrapolation method performed in Matlab, we derive the wire of each net in the examined path of the design, by reading the *wire.report*, which contains the physical dimensions information about the net's wires and their neighboring ones in the same metal stack, from the initial layout extraction step. The arguments passed for the extrapolation are the wire's width, thickness and length, as well as the starting point of each neighboring wire, its length and the distance between them, all obtained from the layout extraction. Hence, in order to find the additional delay for the specific wire, considering the neighboring ones from the layout, we perform a linear interpolation between these values and those of the wire patterns simulated in HSPICE, for which we have already computed the delay increment ratios and dumped them in *TDDB_LUT.lib*, as it is mentioned above. The delay overhead for individual wires is shown in Fig. 3, as a function of the wires' length and distance (right figure), as well as of temperature and operation time, given in years (left figure).

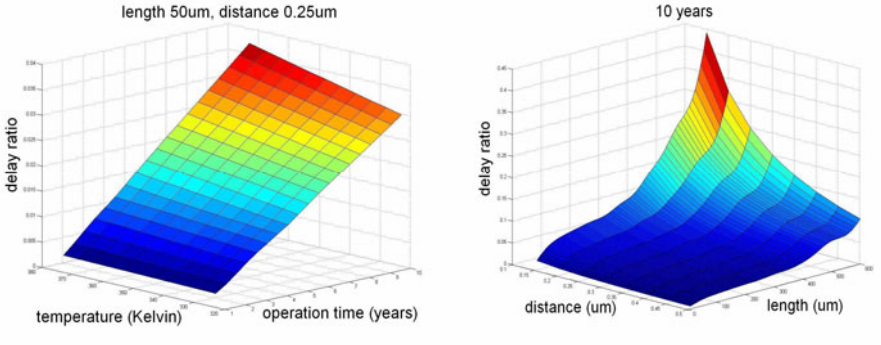

Fig. 3. Delay impact on a wire due to TDDB depending on: temperature (left) and wire length and distance (right)

The calculated delay ratio is then multiplied with the quotient of the target wire's length and the total net length and the result is added to the initial wire's delay due to IMD leakage, which is of course zero. Thus, the additional delay on the specific wire due to TDDB is computed. The same process is performed for all the wires of the examined net in the current path of the design. The total additional delay of the whole net due to TDDB is the weighted summation of the all net's wire delays, where the weights are computed by dividing the wire's length to the total net's length. The updated net's delay is then annotated into the design's SDF file, by finding the specific net and adding the extra delay to the existing one. Thus, the produced SDF, containing the delay overhead from all nets of the path is then annotated to ETS to evaluate the total impact of TDDB on the design's performance. The aforementioned process is continuously followed for all the selected register-to-register paths in the design, while it is applicable to any other design with a reasonable amount of gates.

5 Evaluation of the TDDB Framework to a LEON3-Based MP-SoC

The presented TDDB analysis flow is applied to an MP-SoC design, based on two LEON3 SPARC processor cores, both attached on the AMBA Advanced High Performance bus (AHB). Each processor has seven pipeline stages, while the internal caches include 2 sets of 4K bytes.

The design's RTL description is given in parameterized VHDL, configured via the Gaisler Research automated tools [14]. It is synthesized in Synopsys Design Compiler based on the TSMC 45nm standard-cell library (0.9V, 25 C) and at a clock period constraint of 2ns, resulting in about 30K gates. The floorplanning and the place-and-route steps are implemented in Cadence SoC Encounter, while ETS is employed for Static Timing Analysis. The post-layout Verilog netlist simulation is performed in ModelSim [15], where we obtained switching activity from a matrix multiplication application, running in both processors, as well as from an MP-SoC benchmark initializing the two cores and the system's peripherals, included in the Gaisler's suite. The power analysis is performed in PrimeTime PX, by annotating the .vcd (Value Change Dump) file with the design's activity, derived from ModelSim's framework.

In the proposed case study, we explore how the impact of TDDB on the performance of a LEON3-based MP-SoC design may change, by selecting different placement and routing scenarios, considering the gate-level netlist obtained from synthesis. The dependence of inter-metal leakage on length and distance of wires motivated us to look at different place-and-route strategies favoring either timing or congestion, to find out which scenario minimizes the timing impact of TDDB.

5.1 Experimental Results and Discussion

The main parameters that affect TDDB on the interconnect dielectrics are temperature, wire length and distance between adjacent wires. Regarding temperature, it is mostly affected by the switching activity of the designs. In our LEON3 layouts, which were implemented based on five different place-and-route scenarios, we observed minor temperature differences for two application benchmarks of different computational effort, mentioned above. This is due to the similar power traces

extracted from power analysis for the two application benchmarks executed, as well as to the fact that we have been based on the same floorplan, in order to implement the different placement and routing strategies.

On the other hand, interconnect stack geometrical parameters, like lengths and distances, are mainly impacted by how the circuit is placed-and-routed. In principle, a timing-optimized placement and routing approach will tend to lead to shorter wires, while a congestion-oriented physical implementation strategy will tend to result into longer wires due to coarser placement, as well as to the detouring of wires during routing, to avoid the formation of over-congested areas. Therefore, it is likely that such a strategy will incur larger distances between wires in the same metal layer.

However, the results depicted in Fig. 4 indicate that when placement is congestion-aware (CPl-NR & CPl-CR), the delay overhead due to TDDB is very high. Such a placement scenario will spread out the standard cells and inevitably lead to longer wires at the routing stage, compared to the timing-driven approach and irrespective of the routing strategy. At the other extreme, timing-aware placement and routing seems to result into the minimum delay impact, because the wire lengths are minimal.

Combining these remarks with those of Fig. 3 (delay impact of a wire due to TDDB vs length vs distance vs wire length), we can draw interesting conclusions. Even though at the individual wire level the distance between wires is the most critical parameter for the delay impact of TDDB, at the entire chip level, wire length is the only important parameter for our LEON3-based layouts. Howevesr. in the presented case-study, the different routing strategies, favoring timing or congestion, tend to leave the distances between wires almost unaffected. Hence, a timing-optimal placement and routing approach will also lead to the best layout for TDDB. Since timing is usually the major design spec, the resulting layouts will be optimal for TDDB, while selecting a totally timing-driven place-and-route approach.

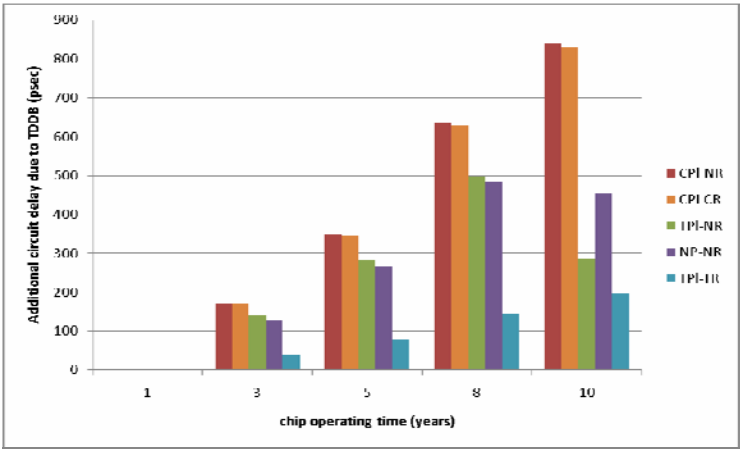

Fig. 4. Chip-level timing overhead due to TDDB for different layout styles (C: congestion-aware, T: timing-aware, N: normal, Pl: placement, R: routing)

This does not imply that designers need not to worry about TDDB, however. Timing-optimal layouts tend to have minimal slack between the data arrival time and the

required time, so that the designs can run at the highest possible clock frequency. Even after 3 years of operation the layout we used has incurred a critical path delay overhead of about 40ps, which might be enough to cause a timing violation. There is a trade-off between actual clock frequency and operating lifetime of the chip. If enough timing slack is left for TDDB tolerance, the expected operating lifetime will be longer, while the design's operating frequency and consequently performance will degrade, and vice versa.

6 Conclusion and Hints for Future Work

In this work, we introduced a reliability analysis framework that estimates the impact of Time-Dependent Dielectric Breakdown on the system's performance, considering an MP-SoC design implemented with a nanometer CMOS technology with different place-and-route strategies. The proposed flow captures the timing violations induced by the inter-metal leakage of low-k interconnects of the examined paths and predicts the gradual performance degradation for each implementation scenario, considering the layout's temperature profile, based on a specific application. Future work may be focused on the frameworks' automation, as well as on the selection of paths, depending on the temperature of design units, the congestion and the length of wires.

References

1. ITRS 2005 public reports (2005), http://public.itrs.net
2. Chen, F., et al.: Critical low-k reliability issues for advanced CMOS technologies. In: Proc. of the 2009 IRPS Symposium, Montreal, Canada, May 26-30, pp. 464–475 (2009)
3. Nitta, S., et al.: Copper BEOL interconnects for silicon CMOS logic technology. In: Davis, J.A., Meindl, J.D. (eds.) Interconnect Technology and Design for Gigascale Integration. Springer, Heidelberg (2003)
4. Gonella, R.: Key reliability issues for copper integration in damascene architecture. Journal of Microelectronic Engineering 55(1-4), 245–255 (2001)
5. Tan, T.L., Gan, C.L., Du, A.Y., Cheng, C.K., Gambino, J.P.: Dielectric degradation mechanism for copper interconnects capped with CoWP. Applied Physics, Letter. 92, 201916 (2008)
6. Takeda, K.-i., Ryuzaki, D., Mine, T., Hinode, K., Yoneyama, R.: Copper-induced dielectric breakdown in silicon oxide deposited by plasma-enhanced chemical vapor deposition using trimethoxysilane. Journal of Applied Physics 94(2572) (2003)
7. Chen, F., et al.: Line-edge roughness and spacing effect on low-k TDDB characteristics. In: Proceedings of the 2008 International Reliability Physics Symposium (IRPS), April 27-May 1, pp. 132–138 (2008)
8. Chen, F., Shinosky, M.: Addressing Cu/Low-k Dielectric TDDB Reliability Challenges for Advanced CMOS Technologies. IEEE Transactions on Electron Devices 56(1), 2–12 (2009)
9. Li, Y.: Low-k dielectric reliability in copper interconnects, PhD Dissertation, Katholieke Universiteit Leuven (2007)
10. Guo, J., et al.: A Tool Flow for Predicting System-Level Timing Failures due to Interconnect Reliability Degradation. In: Proc. of the 2008 GLSVLSI International Symposium, Orlando, Florida, USA, May 4-6, pp. 291–296 (2008)

11. Guo, J., et al.: The Analysis of system level timing failures due to interconnect reliability degradation. IEEE Transactions on Device and Material Reliability (2009)
12. Huang, W., Ghosh, S., Velusamy, S., Sankaranarayanan, K., Skadron, K., Stan, M.R., Brown, C.L.: HotSpot: a compact thermal modeling methodology for early-stage VLSI design. IEEE Transactions on VLSI Systems 14(5) (May 2006)
13. Cadence SoC Encounter Database Access command reference, http://www.cadence.com
14. Aeroflex Gaisler Research, http://www.gaisler.com
15. Mentor Graphics ModelSim, http://www.model.com