Automated Insertion of Twin Gates to improve Reliability concerning Gate Oxide Breakdown

Hagen Saemrow*[a], Claas Cornelius[a], Frank Sill[b], Anreas Tockhorn[a], Dirk Timmermann[a]
[a]Dept. of Electrical Engineering, University of Rostock, Rostock, Germany;
[b]Dept. of Electrical Engineering, Federal University of Minas Gerais, Belo Horizonte, Brazil

ABSTRACT

Scaling device dimensions towards atomic scales leads to increased reliability and yield concerns which considerably affects the work of integrated circuit designers. Furthermore, the complexity of integrated systems increases which leads to a demand for tool assisted reliability insertion during the design process. Lots of research efforts have focused on soft-errors and system-level approaches. However, only few low-level solutions have been published to enhance lifetime reliability. Investigations in this field have reached an up to 200 % increased reliability concerning gate oxide breakdown if so called Twin Gates have been inserted. This contribution comprehensively presents algorithms to implement these redundant cells automatically during logic synthesis. Besides the placement in the whole design process, approaches are provided to insert Twin Gates correctly considering timing and area issues.

Keywords: Design flow, integrated circuit design, redundant systems, reliability, gate oxide breakdown

1. INTRODUCTION

The performance of integrated circuits was improved significantly in the past decades due to aggressive scaling of technology parameters. On the other hand, as we are approaching the limits of nanotechnology yield, reliability and power concerns arise. For instance, design and process error margins due to material defects and imperfections are reduced which has direct impact on product yield and lifetime reliability. Unfortunately, full functional system tests are not feasible due to the rising complexity of integrated systems. Hence, tool assisted insertion of reliability mechanisms into the design flow will be one of the key priorities in the future [5].

Increasing yield has ever been a crucial aspect for manufacturing of integrated circuits. Up to now, mostly manufacturing engineers have focused on yield enhancements, but against the background of nanotechnology, circuit designers are becoming more aware of the capabilities for improvement during system design. This part of the design process is usually known as Design For Manufacturability (DFM), respectively as Design For Yield (DFY) and is implemented in commercial layout products. Used in the layout phase of the design process, this step comprises layout modifications and, amongst others, preparations for static reconfiguration. Layout modifications are rearrangements of system parts to optimize the design concerning yield. It consists of e.g. via duplication or substitution, the minimization or the widening of interconnects and the reassignment of layers [6]. Static reconfiguration is often utilized in memory manufacturing where defective parts are disconnected from and spare parts are connected to the system by using laser fuses [7]. Further on, additional test features – like Design For Test (DFT) scan chain registers or JTAG boundary scan controllers – will more and more be inserted to be able to test the design after manufacturing.

Besides reliability issues at the beginning of a circuit’s lifetime, miniaturization leads to an increased sensitivity of transistors and interconnects to different kinds of failures during system operation. Effects which affect lifetime reliability can be separated into two categories: transient failures and permanent failures. Transient defects contain soft errors, defects due to environmental variations and operational effects like crosstalk. Soft errors are caused by radiation events which can change the amount of electrical charge on internal nodes. Recently, a lot of techniques for soft-error resilience and crosstalk avoidance have been published. But to the best of the authors’ knowledge, only crosstalk avoidance by calculating signal integrity is fully integrated into commercial automated layout tools. The published solutions vary from RC-filters or additional capacitances to harden internal vulnerable nodes [8] to higher level approaches like error-correcting [17] or error-detecting codes and watchdog processors [18]. A promising low level approach, which increases reliability concerning soft errors by more than a magnitude, has been published by Mitra [9].

*hagen.saemrow@uni-rostock.de; phone ++493814987278; fax ++493814981187251; uni-rostock.de

Thereby, latches are duplicated and the outputs are stabilized by a Muller-C-Element. Because of the existing duplicated latches in the scan path, the area overhead is minimal and an integration of this technique into existing CAD tools could easily be implemented.

In contrast to transient failures which disappear again, permanent failures will harm integrated circuits permanently from the point in time they occur. Causes which affect lifetime reliability are e.g. Time-Dependent Dielectric Breakdown (TDDB), electromigration (EM) or thermal cycling [1]. The impact of these adverse effects on integrated circuits increases with every new technology generation because of the non-ideal scaling of power supply and an increased transistor count and power density. Moreover, adaptive processing (e.g. DFS, DVS) further worsens the degree of these defect mechanisms and therefore the probability of circuit malfunctions. As far as we know, from all lifetime reliability issues, only electromigration is examined by commercial CAD tools by analyzing the EM-vulnerability of power rails after place and route. As a result, the dimensions of the power lines are adapted if violations occur. Furthermore, integrated circuits have the possibility to a self check at certain points in time to respond to errors during operation if Built-In-Self-Tests (BIST) are inserted.

In contrast to transient defects, only little effort has been made so far on techniques to enhance lifetime reliability in the presence of permanent failures. A high level approach which reduces the effects of lifetime issues at run-time has been published in [10] where a dynamic system management adapts the operating conditions in response to an observed hardware usage to stay within a given reliability target. Another very different approach has been made in [11] where it has been assumed that device failures cannot be prevented but have to be resolved. Therefore, redundant transistors are inserted randomly into the design to increase yield as regards stuck-open transistors. This idea has been extended in [12] where the redundant transistors are inserted only at those instances that are most vulnerable to TDDB which increases not just the yield but also lifetime reliability concerning gate oxide breakdown. This approach has been transferred to higher levels of IC-design at our institute. Investigations have shown an increased improvement due to the duplication of logic gates (so called Twin Gates) in contrast to duplications at transistor level. Fortunately, this approach is predestinated to be integrated into automated synthesis tools which will be presented in the following.

Our approach to increase lifetime reliability concerning gate oxide breakdown and the breakdown mechanism itself will be explained in chapter 2 of this contribution. Subsequently, algorithms for the integration of Twin Gates into logic synthesis are presented in chapter 3. Lastly, section 4 will conclude the paper.

2. TWIN GATES TO INCREASE RELIABILITY CONCERNING GATE OXIDE BREAKDOWN

2.1 Gate Oxide Breakdown

One of the fundamental components for reliability, performance and power consumption is the gate oxide which is the dielectric isolation between the transistor input and the conducting channel. The thickness of gate oxide comprises only a few atomic layers (<20 Å) in current technologies. Due to this fact, non-ideal scaling of the supply voltage and increased electric fields, the gate oxide has become highly vulnerable to breakdown mechanisms causing transistor defects and logical malfunctions.

The point of time a conducting path is generated between the gate and the substrate is called gate oxide breakdown [2] whereas the cause can originate from two different situations. Firstly, sudden damage occurs due to extreme overvoltage and leads to non-isolating gate oxide, e.g. caused by Electro-Static Discharge (ESD). Secondly, a rather slow destruction over time is also possible, called Time-Dependent Dielectric Breakdown (TDDB). Thereby, charge traps start to form in the gate oxide during operation which causes an autocatalytic loop of events: overlapping charge traps form a conducting path between gate and substrate which leads to increased current flow as well as heat dissipation. This again causes thermal damage and, hence, more charge traps. This positive feedback loop results in an accelerated breakdown and finally in a defect transistor [3][4]. Depending on the I-V characteristics of the defect transistor, the breakdown is distinguished into resistor-like hard breakdowns and soft breakdowns. The conductance of the soft breakdown is strongly non-linear and more limited concerning the amount of current flow [19]. In most cases a hard breakdown is the final result of a soft breakdown.

Until the end of the last century, it was widely assumed that a gate oxide breakdown ─ in any case considered as a hard breakdown ─ is a catastrophic failure caused by a massive short through the gate oxide which leads to a logical malfunction of the transistor. But besides the identification of the soft breakdown mode, it has been shown in the meantime that gate oxide breakdowns not necessarily result in logic malfunctions of the transistor. Rather than a logic failure, an affected transistor and its associated logic cell suffer from a modified delay [18]. Certainly, the whole circuit fails if the timing between the cells is no longer balanced due to delay failures of multiple gate oxide breakdowns.

To be able to simulate the mechanisms of gate oxide breakdown, we chose an equivalent circuit model which is depicted in figure 1 [14]. There, two transistors were added in parallel. Modifying the widths *w1*, *w2* and *w3* of all three transistors will result in an appropriate representation of the current flow at the drain, source and gate for a transistor with a punctual gate oxide breakdown on a certain location of the transistor gate. The resistor *R* connecting the net of the drains of the two additional transistors with the net of every gate of all three transistors simply represents the resistance of the short between gate and bulk of the transistor. With this model it is possible to simulate a gate oxide breakdown on any location from gate to substrate with any resistance of the breakdown path.

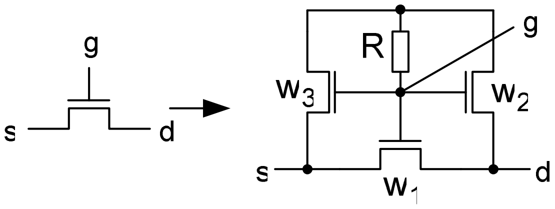

Figure 1 Circuit model for a gate oxide breakdown between the gate (g) and the conducting channel:s - source; d - drain

2.2 Twin Gates

The elementary approach of the work is to enhance the reliability of integrated systems in the design phase by adding redundant components. Unlike most expectations that redundant transistors provide the most fine-grained and, thus, the best results, the most reliable designs are the designs which are duplicated at gate level (Twin Gates). Figure 2 depicts the Twin Gate strategy (b) and schematics of the basic design (a) and a design with transistor duplication (c) to show the differences. Twin Gates are duplicated gates which in- and outputs are connected, whereas at transistor level, every transistor is duplicated and the duplicated transistor is connected to the nets of the original one.

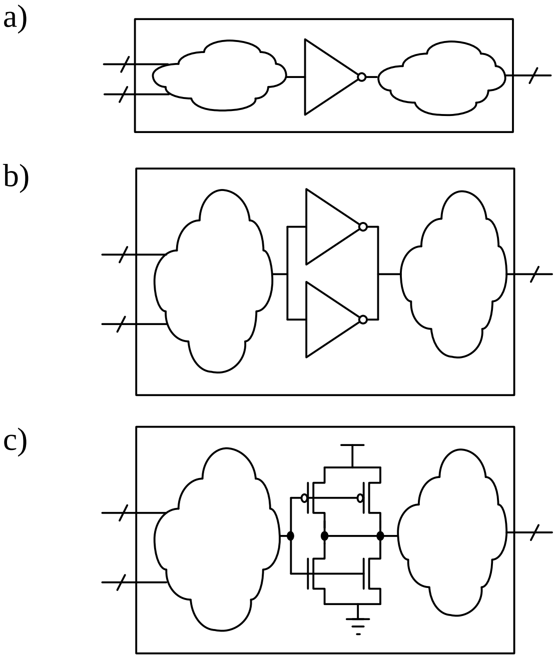

Figure 2 Duplication strategies: a) Basic design
b) Gate duplication (Twin Gates)
c) Transistor duplication

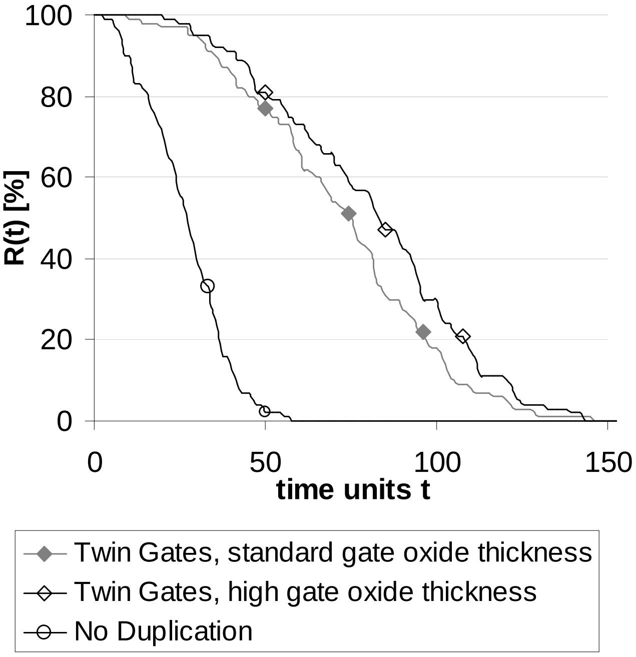

Figure 3 Reliability enhancements due to Twin Gates

The cause for the better results of Twin Gates in contrast to transistor duplication is found in the transistor stacks due to the fact that both implementations only differ in the duplication of transistor stacks. Due to the defect at a transistor in the stack, their connected nets at drain and source are charged to a voltage amount which is related to the degree of the gate oxide breakdown when the inputs switch. As a result, the current flow from drain to source at the transistor is less related to a flawless stack, because of the lower drain-source- and gate-source-voltage. This leads finally to a longer fall or rise time for a defect stack because of the slower discharge of its output capacitance in contrast to a flawless stack. If a defect occurs in a duplicated stack, one of both Twin Gates stacks will remain unaffected and is able to discharge/charge the output faster. At transistor duplication on the other hand, a defect would also affect the voltage levels of the nodes of the duplicated stack due to the cross links. Therefore, stacks duplicated with Twin Gates are slightly faster than transistor doubled stacks. Lastly, this leads to better reliability results for Twin Gates designs because they meet the timing constraints longer in the presence of multiple defects.

Further improvements are achieved by inserting gates whose transistors have a thicker gate oxide than the original ones. Figure 3 shows the reliability improvements of both Twin Gate strategies in contrast to the reliability of a basic Wallace multiplier. The reliability *R(t)* of a system represents the probability of a system to perform as desired until time *t*. The curves represent the enhancements with a duplication rate of 100 % and constant failure rate. Hence, the reliability improvements were made at the cost of doubled power consumption and doubled area, but with an equal or a slightly increased delay. Beyond lifetime reliability enhancements, Twin Gates would increase also yield as regards stuck-open transistors due to the additional gates which implicitly replace defect gates.

To decrease the overhead of power and area, fewer gates should be duplicated. Implementation algorithms to choose gates to reach the highest improvement for a given area overhead are presented in the next chapter.

3. INTEGRATION OF THE TWIN GATE STRATEGY INTO THE DESIGN FLOW

3.1 Basic design flow

The design process from a register-transfer-level (RTL) description to a GDSII file for manufacturing is basically classified into three main steps:

1. Logic synthesis process which parses, translates and optimizes an RTL design into discrete logic gates
2. Place and Route design stages which consists amongst others of floorplanning, placement of standard cells and routing of the cells
3. Layout which provides the different layers for manufacturing and which is usually automated by place and route tools for wiring etc.

Implementing Twin Gates is related to the logic synthesis because of the duplication of logic gates. Therefore, the logic synthesis is explained more in detail in the following. On the basis of a given gate library and constraints regarding timing, power and area, the compiler maps the RTL description to a gate netlist whose cell descriptions include static timing, power and area parameters besides the logic function. During and after compilation, a static timing analysis will usually check if timing constraints are met. Further on, functional tests are possible by implementing the netlist into a test environment which checks by simulations if the output pattern is generated correctly after feeding the design with a predefined input pattern. If the test fails or timing and other constraints are not met, the design process has to start again from the beginning or in the worst case the RTL design has to be changed. It is also possible, to analyze the dynamic power consumption by extending the test which then generates a switching activity file that stores the logic status (low, unknown or high level) of every net of the design for the whole simulation time. With this switching activity file it is possible to calculate the dynamic power the design would consume.

3.2 Extended logic synthesis to implement Twin Gates automatically

Besides the possibility to double every gate, it is more reasonable to double only a part of the design to restrict the area and power overhead. Hence, an estimation of the failure probability of every gate is necessary. This requires additional steps in the design process which are attached to the basic design flow after the generation of the switching activity file. Due to the dependence of TDDB on the absolute voltage level at the gate input [2] this switching activity file is used to classify all gates depending on their failure probability related to TDDB. After this vulnerability analysis the insertion of Twin gates follows only at the most vulnerable instances.

The whole extended logic design process is depicted in figure 4. Here, the process steps are bordered by rectangles, whereas necessary files are encircled by ellipses. These files are required to execute the design steps and some of them are generated during the process. Firstly, the logic synthesis will be processed which uses the RTL design and the gate library as input files. The generated gate netlist is used for the following steps. As explained in chapter 3.1, the functional test and the switching analysis serve as verification for the functional correctness and the creation of the switching activity file for the power analysis. The next two steps (bold text) are additionally inserted to gain a reliability improved netlist as regards TDDB. Both design steps will be explained in detail in the next chapters.

3.3 Vulnerability analysis

As introduced in chapter 3.2, the probability of a defect due to TDDB is related to the electric field which is applied to the transistor gate [2]. For instance, if the gate voltage of a NMOSFET is $V_G = 0$, the probability for a TDDB defect is zero because there is no electric field which is applied to the gate. On the other hand, if $V_G = V_{DD}$ – the power supply voltage – the probability is the highest possible due to the permanent electric field which is applied to the transistor gate. The same applies for PMOSFET, but with inverse voltage levels, the crucial case is $V_G = 0$. Due to time dependence of TDDB the probability of a defect increases with the time the transistor is in the critical case.

As a consequence, the probability of a transistor to fail is directly proportional to the amount of time a cell is in a certain logic state. Due to the switching activity file which provides the probability of logic states for every net of the design, two approaches obtaining a sorted list as regards defect probabilities are conceivable:

a) Sorting of the gates only due to their logic states

b) Sorting of the gates due to the defect probabilities of their transistors

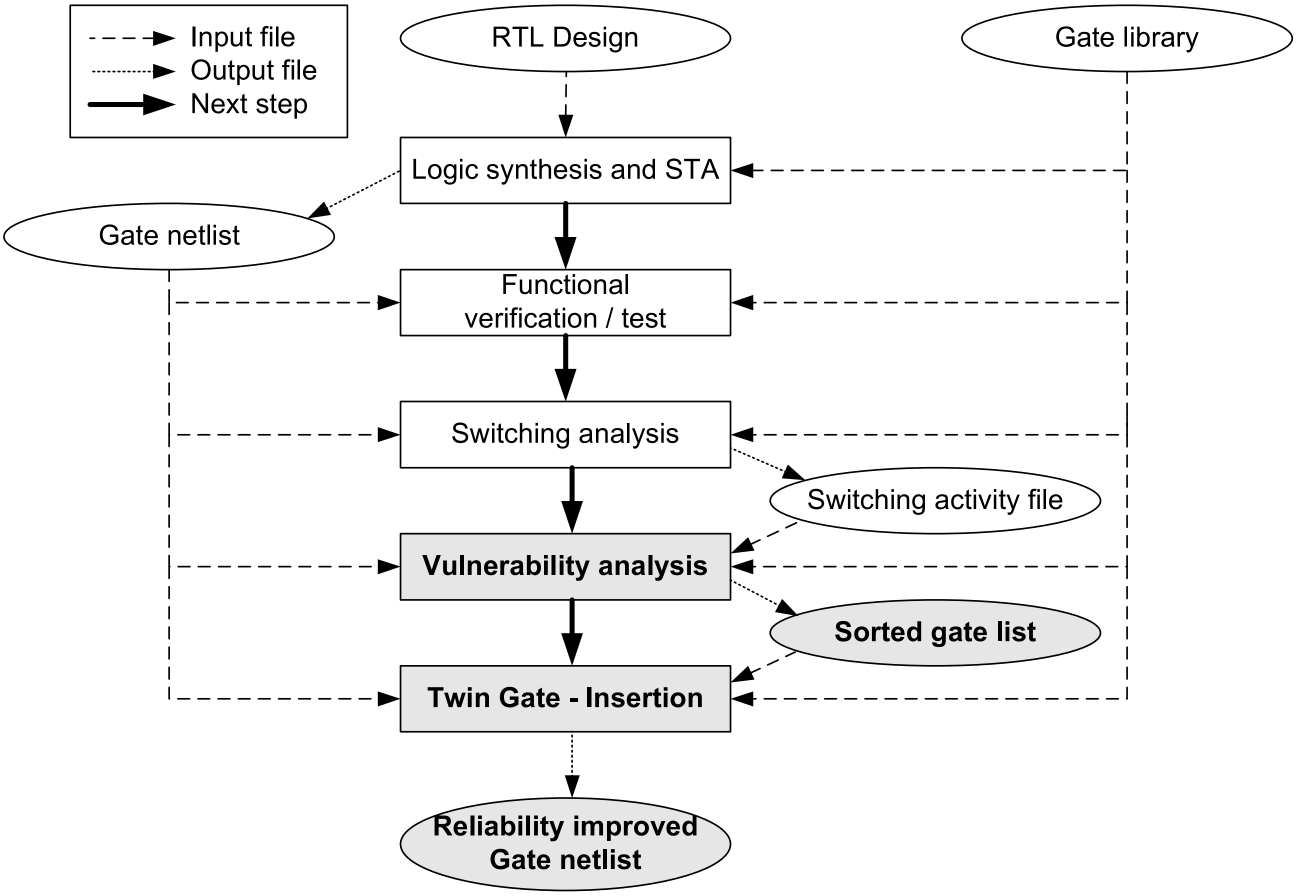

Figure 4 Design process steps, extended version (bold text) to enhance reliability as regards TDDB

The differences between both approaches are depicted in figure 5. The first case is easier to implement but not that accurate due to the disregard of internal transistors which are not directly connected to the input nets of a cell. In this case, the sorting of the gates are effected firstly by the logic state probability attached to the gate, and subordinated by the number and type of the transistors. For instance, if two gates have the same logic state probability with three-quarter of the operating time at logic state one, the cell with more NMOSFET transistors would be identified as more vulnerable.

In the second case, the logic state probability and therefore the defect probability of every gate would be assigned downwards to the transistors the cells comprise. The defect probability of internal transistors would be calculated with the internal connections of the gate. Next, every transistor of the design will be sorted according to its defect probability. Again, there are two possibilities to order the gates. First, every gate is sorted by the transistor which appears first in the sorted transistor list. Second, every gate is assigned with a cumulative defect probability – calculated with the probability of every transistor the gate consists of. With this information, a sorted list of gates will be generated.

If transistors with different gate oxide thicknesses are used the defect probabilities have to be adjusted due to the differences. As a widely used assumption, it is assumed that with a 0.1 nm thicker gate oxide the failure rate decreases by one magnitude [2].

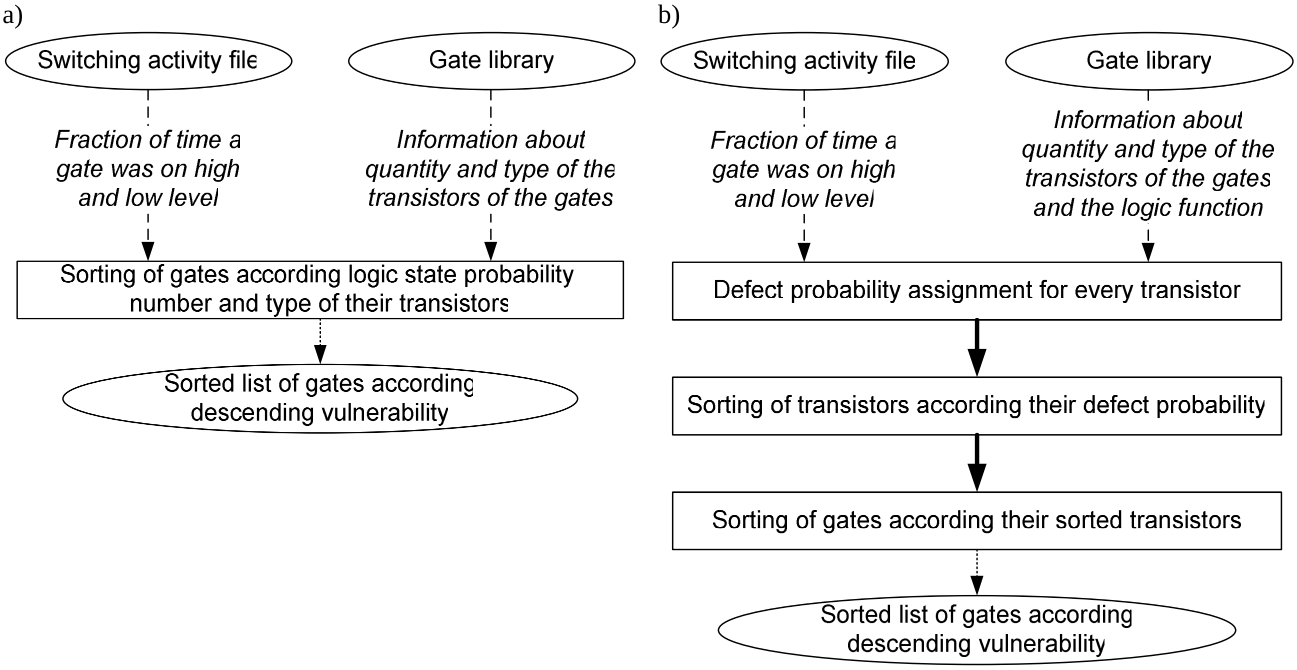
a)
Switching activity file
Gate library
Fraction of time a gate was on high and low level
Information about quantity and type of the transistors of the gates
Sorting of gates according logic state probability number and type of their transistors
Sorted list of gates according descending vulnerability
b)
Switching activity file
Gate library
Fraction of time a gate was on high and low level
Information about quantity and type of the transistors of the gates and the logic function
Defect probability assignment for every transistor
Sorting of transistors according their defect probability
Sorting of gates according their sorted transistors
Sorted list of gates according descending vulnerability

Figure 5 Vulnerability analysis: Sorting of the gates due to a) their logic states b) the defect probabilites of their transistors

3.4 Twin Gates Insertion

After the generation of the sorted list of gates according to the descending defect probability, a fraction of cells is doubled. The amount of doubled gates is related to the area overhead the design is allowed to have (critical area). The duplication of a gate leads to a local change of the load capacitance and driver strength and therefore a change of the timing behavior of the whole design. Whereas the load capacitance of the previous gates increases and thus, their propagation delay increases accordingly, the propagation delay of the doubled gate decreases because of the doubled driver strength. There are several possibilities to insert the Twin Gates. The simplest and quickest algorithm would be an insertion of Twin Gates until the critical area is reached. A problem of this approach occurs if the STA after the insertion fails. Without a detailed timing analysis it is then not possible to detect which of the Twin Gates has to be removed from the extended netlist.

Responding to this issue figure 6 depicts a basic algorithm, how timing and area constraints are considered during the insertion process. Firstly, due to the sorted gate list the cell with the highest priority (= highest defect probability) is chosen for duplication. The design is extended with a Twin Gate that has the same function as the original one and whose transistors have the same widths as the transistors of the original gate. To improve the reliability further, transistors with thicker gate oxide could be inserted. However, these gates would affect the timing behavior of the whole design more than a completely doubled gate due to the reduced driving capability of the combination of both gates because transistors with thicker gate oxide are slower. Next, with the information about timing and area, a STA and an area check is following. If the design meets the timing constraints, the gate netlist is extended by the Twin Gates. Furthermore, the two previous design steps are repeated, if the design area has not reached the critical area. Therefore, the duplicated gate will be removed from the sorted gate list and the next gate is the highest priority cell. If the critical area is reached, the algorithm ends with the extended gate net list as output.

Because of the high priority of a cell which is rejected due to a failed STA, it could be worth to retry an insertion after certain additional Twin Gates because of the changed timing behavior then. Another approach for a failed STA is a recursive algorithm which duplicates preceding gates to compensate their increased load capacitance.

In the following design steps – like place and route and layout phase – the Twin Gates should be placed near their original gates because keeping the timing situation for both gates equally is a key element for the insertion of Twin Gates. A wider gap between both cells could result in length differences of the interconnects which the gates are linked to. Furthermore, the impact of within-die variations would affect both gates differently because of the distance on the die. Both issues would disturb the timing which is essential for Twin Gates due to the connected outputs. The larger the switching difference is between both outputs, the higher is the possibility that the output net is in an indefinite state for a longer time as expected and the more short current circuits would be generated.

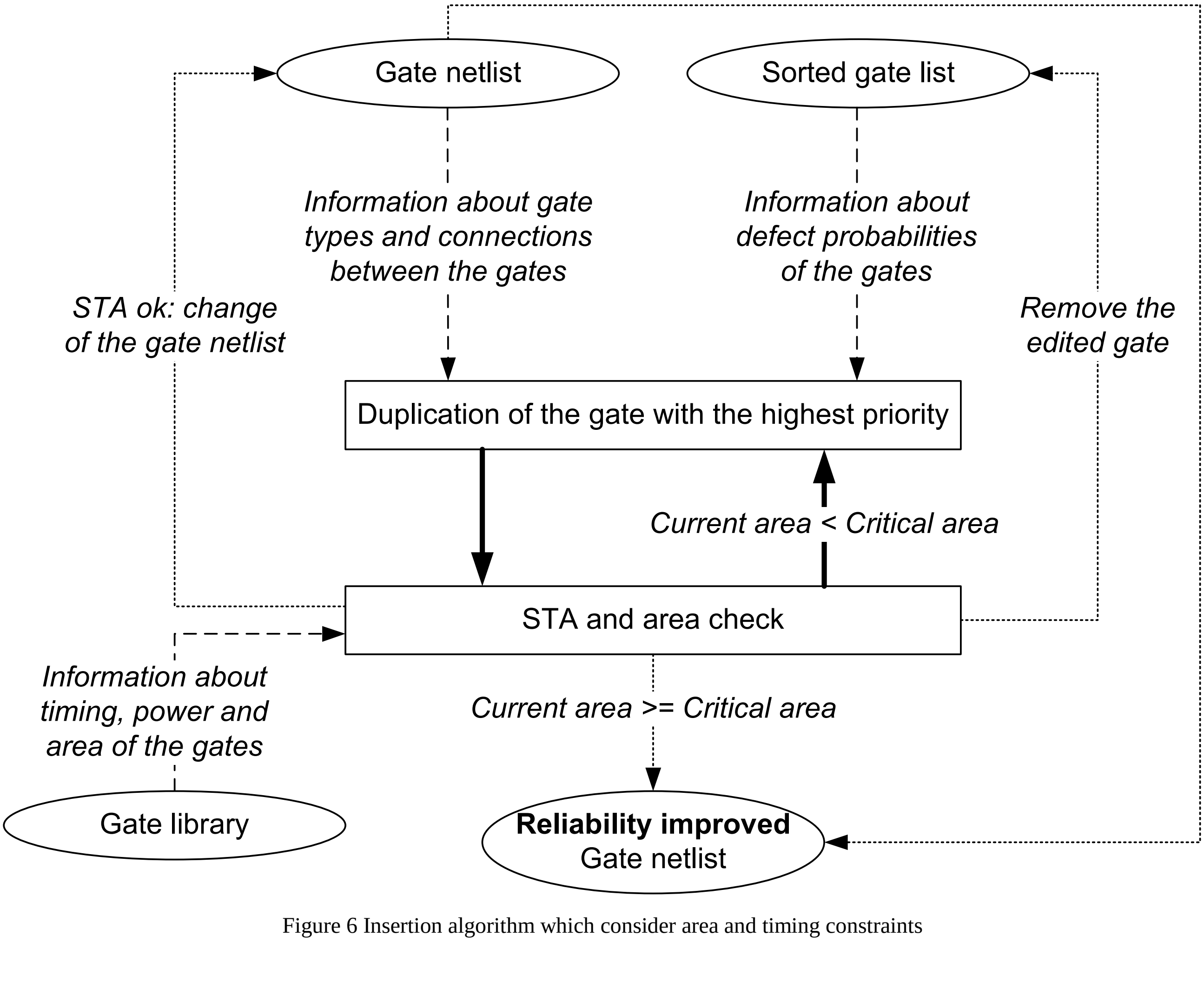
Gate netlist
Sorted gate list
Information about gate types and connections between the gates
Information about defect probabilities of the gates
STA ok: change of the gate netlist
Remove the edited gate
Duplication of the gate with the highest priority
Current area < Critical area
STA and area check
Information about timing, power and area of the gates
Current area >= Critical area
Gate library
Reliability improved
Gate netlist

Figure 6 Insertion algorithm which consider area and timing constraints

4. CONCLUSION

This contribution introduced the need for improvements of lifetime reliability. Investigations as regards reliability enhancements lead to the knowledge that the impact of gate oxide breakdown could be reduced by inserting Twin Gates. Additional steps in the design flow were presented with which these redundant logic gates could be inserted automatically into the gate netlist. Extending the logic synthesis, algorithms were identified which choose the gates with highest defect probabilities based on the logic state probability of every gate. Furthermore, timing and area constraints were considered during insertion.

REFERENCES

[1] Srinivasan, J., Adve, S., Bose, P. and Rivers, J., “The Impact of Technology Scaling on Lifetime Reliability”, In Proc. of DSN, 2004.

[2] Stathis, J., “Reliability Limits for the Gate Insulator in CMOS Technology”, In IBM Journal of Research and Development, 2002.

[3] Crook, D., “Method of Determining Reliability Screens for Time Dependent Reliability Breakdown”, In Proc. of Intern. Reliability Physics Symposium, 1979.

[4] Vogel, E. et al., “Reliability of Ultra-Thin Silicon Dioxide Under Combined Substrate Hot Electron and Constant Voltage Tunneling Stress”, In Trans. of Electron Devices, vol. 47, no. 6, 2000.

[5] Semiconductor Industry Association (SIA), “International Technology Roadmap for Semiconductors”, Release 2007, Published on-line: http://www.itrs.net/.

[6] Chiluvuri, V. and Koren, I. “Layout-Synthesis Techniques for Yield Enhancement”, In Trans. of Semic. Manufacturing, vol. 8, no. 2, 1995.#

[7] Chen, Z. and Koren, I., “Techniques for Yield Enhancement of VLSI Adders”, In Proc. of ASAP, 1995.

[8] Omana, M., Rossi, D. and Metra, C., “Latch Susceptibility to Transient Faults and New Hardening Approach”, In Trans. on Computers, vol. 56, no. 9, 2007.

[9] Mitra, S. et al., “Robust System Design with Built-In Soft-Error Resilience“, In Computer, vol. 38, 2005.

[10] Srinivasan, J. et al., “The Case for Lifetime Reliability-Aware Microprocessors”, In Proc. of ISCA, 2004.

[11] Sirisantana, M., Paul, B. and Roy, K., “Enhancing Yield at the End of the Technology Roadmap”, In Trans. of Design&Test of Computers, vol. 21, no. 6, 2004.

[12] Cornelius, C., Sill, F., Saemrow, H., Salzmann, J., Timmermann, D., da Silva, D., „Encountering Gate Oxide Breakdown with Shadow Transistors to Increase Reliability“ , In Proc. of SBCCI, 2008.

[13] Koren, I. and Krisha, C., “Fault-tolerant Systems“, M Kaufmann, 2007.

[14] Renovell, M., Gallière, J., Azaïs, F. and Bertrand, Y., “Modeling the Random Parameters Effects in a Non-Split Model of Gate Oxide Short”, In Journal Electronic Testing, vol. 19, no. 4, 2003.

[15] Johnson, M. C., Somasekhar, D., Roy, K., “Leakage control with efficient use of transistor stacks in single threshold CMOS”, In Proc. of DAC, 1999.

[16] Kaczer, B., et al., “Impact of MOSFET Gate Oxide Breakdown on Digital Circuit Operation and Reliability”, In Transactions on Electron Devices, vol. 49, no. 3, 2002

[17] Spica, M; Mak, T., “Do we need anything more than single bit error correction (ECC)?”, Records of the International Workshop on Memory Technology, Design and Testing, 2004

[18] Mahmood, A; Mc Cluskey, E., “Concurrent Error Detection Using Watchdog Processors – A Survey”, Trans. on Computers, vol. 37, no. 2, 1988

[19] Kaczer, B. et al., “Gate oxide breakdown in FET devices and circuits: From nanoscale physics to system-level reliability”, Elsevier, 2007. Renovell, M., et al., “Modeling the Random Parameters Effects in a Non-Split Model of Gate Oxide Short”, In Journal Electronic Testing, 2003.