

Effiziente Simulation von Gateoxiddefekten auf Gatterebene mit Transistorlevel-Genauigkeit

Hagen Sämrow, Claas Cornelius, Philipp Gorski, Jakob Salzmann, Dirk Timmermann

Universität Rostock, Institut für Angewandte Mikroelektronik und Datentechnik, Rostock
saemrow@trinamic.com, claas.cornelius@uni-rostock.de

Kurzreferat

Die fortschreitende Skalierung integrierter Schaltungen führt dazu, dass Verschleißerscheinungen wie Gateoxiddefekte immer mehr in den Fokus von Schaltungsdesignern geraten, da sie ein Ausmaß erreichen, das die Lebensdauer dieser Schaltungen signifikant begrenzt. Um dieser Entwicklung Rechnung zu tragen, verwenden viele Tools, die oberhalb der Transistorebene angesiedelt sind, allerdings nur pauschale Erhöhungen der Verzögerungszeit, um den Verschleiß zu modellieren. Dieser Beitrag stellt daher einen Simulator vor, der Gateoxiddefekte ausgehend von Transistormodellen auf die Gatterebene überträgt. Damit wird bereits nach der Synthese der Gatternetzliste eine genaue Abschätzung möglich, inwieweit Verschleißeffekte die Schaltung mit der Zeit beeinflussen. Darüber hinaus bietet der Simulator verschiedene Einstellungen der Fehlerverteilung und erstellt sowohl Berichte über Spannungspegel aller Netze als auch Gatterbibliotheken, mit denen das Zeitverhalten analysiert werden kann. Als Ergebnis zeigt dieser Beitrag wie mit dem vorgestellten Ansatz effiziente Simulationen von Gateoxiddefekten auf Gatterebene durchführbar sind. Damit kann das sich verändernde Verhalten (aufgrund von Alterung und Verschleiß) eines Designs über die Zeit modelliert werden. Die Ergebnisse belegen außerdem, dass die Genauigkeit der berechneten Werte die Qualität von komplexeren Transistormodellen erreicht.

1. Stand der Technik

Die Verzögerungszeit integrierter Schaltungen konnte durch die Skalierung der Technologieparameter in der Halbleitertechnik in den letzten Jahrzehnten erheblich gesenkt werden. Im Gegensatz dazu werden Ausbeute und Zuverlässigkeit mit jeder fortschreitenden Verkleinerung negativ beeinflusst. So führt die Reduzierung der physikalischen Dimensionen auf dem Chip auch zu verkleinerten Design- und Prozessfehlerspannen, was in einer erhöhten Fehleranfälligkeit resultiert [1].

Die Verschlechterung der Ausbeute war schon immer ein kritischer Aspekt der Herstellung integrierter Schaltungen und früher vor allem ein Sachverhalt, dessen Verbesserung sich ausschließlich auf die Produktion beschränkte. Allerdings sind mit steigender Komplexität der Schaltungen auch Designer gezwungen, die Ausbeute beim Entwurf zu berücksichtigen, beispielsweise im Design For Manufacturing (DFM) bzw. Design For Yield (DFY). Des Weiteren sind heutzutage zunehmend zuverlässigkeitsrelevante Effekte zu beachten, die nicht die Ausbeute sondern den fehlerfreien Betrieb der Schaltung betreffen. Zum Beispiel permanente Fehler, die erst nach einer bestimmten Betriebszeit auftreten, senken die Lebensdauer integrierter Schaltungen mittlerweile soweit, dass ohne adäquate Gegenmaßnahmen die vom Kunden erwarteten Betriebslaufzeiten nicht mehr gewährleistet werden können. Zu diesen Verschleißerscheinungen zählt das Gateoxid Breakdown (GOB), dessen Auswirkungen mit der Skalierung ansteigen [2]. So werden Techniken zur Reduzierung dieser Störungen immer wichtiger, sowohl mittels Anpassung der Technologie als auch des Designs. Dazu müssen allerdings auch die Auswirkungen zur Designzeit abschätzbar werden.

Eine Art des Gateoxid Breakdowns ist das Time-Dependent Dielectric Breakdown (TDDB), bei dem das Gateoxid des Transistors mit der Zeit immer weiter geschwächt wird. Dadurch wird das Zeitverhalten der Schaltung zunehmend gestört [8], was dann schließlich in einem kompletten Fehlverhalten münden kann, wenn mehrere defekte Transistoren das Timing des Designs durcheinander bringen [3]. Beim TDDB wird im Allgemeinen zwischen soft und hard Breakdown unterschieden, wobei Letzterer den endgültigen Ausfall der Transistorfunktion bezeichnet. Der soft Breakdown ist der Beginn der Störung im Gate, wobei zunächst nur die Stromkennlinien nichtlinear verändert werden, da das Dielektrikum des Transistors nur punktuell durchbrochen wird [4],[5]. Bereits ein kleiner Durchbruch beschleunigt jedoch den Gateoxiddefekt immer weiter (d.h. der Widerstand zwischen Gate und Bulk nimmt ab), womit es irgendwann zum hard Breakdown kommen kann. Weitere Gateoxiddefekte, die auch vorwiegend das Zeitverhalten stören, sind Hot-Carrier Effekte (HC) und die Negative Bias Temperature Instability (NBTI), die die Schwellspannung von p-MOSFETs erhöht [2],[11].

Bevor der soft Breakdown wissenschaftlich nachgewiesen war, basierten Zuverlässigkeits-simulatoren auf der Annahme, dass Gateoxiddefekte zu unmittelbaren Funktionsstörungen führen. Die Berkeley Reliability Tools (BERT) nutzen eine Komponente, den Circuit Oxide Reliability Simulator (CORS), um die Wahrscheinlichkeit von Fehlern in integrierten Schaltungen aufgrund von Gateoxiddefekten vorherzusagen. Diese Vorhersage basiert auf Knotenspannungen in Spice-Simulationen und einem $(1/E)$ -Modell für die Fehlerverteilung [7]. Weitere Simulatoren, die Verschleißaspekte adressieren, wurden für NBTI und HC erstellt. Der in [9] vorgestellte Ansatz generiert gatterspezifische Parameter, wie die Verzögerungszeit, mittels Transistorlevel-Simulationen. Zusammen mit probabilistischen Fehlerverteilungen ermöglichen die Gatterparameter Analysen zur Verzögerungszeit des Designs. Der HC-Simulator generiert auch Daten über die Veränderung der Verzögerungszeit der gesamten Schaltung aufgrund von HC Effekten. Dieser Ansatz ist vergleichbar mit einem probabilistischen Simulator, der die Performance aufgrund von Parameterschwankungen nach der Produktion modelliert [10]. In modernen Analysetools existieren mehrere Ansätze. Zum einen werden auf Transistorebene Verschleißeffekte mittels Parameter modifiziert, womit der Stromfluss im Bauteil verändert wird [12]. Zum anderen wird auf der Gatterebene agiert, indem man die Gatterverzögerungszeit pauschal um definierte Prozentpunkte vergrößert [13]. Eine analytische Methode zur Vorhersage von fehlerhaften, größeren Designs wurde in [14] vorgestellt. Hierbei wurden hard Breakdowns als ursächlich für mögliche Störungen der Schaltung herausgestellt. Umstände, die zu kritischen Spannungsdifferenzen ($V_{HIGH} - V_{LOW} > \text{Grenzwert}$) einzelner Netze führen und somit zu Funktionsfehlern, wurden mittels modifizierter Weibull-Wahrscheinlichkeitsverteilungen der Transistoren dargestellt. Daraus entwickelten die Autoren eine Vorhersage für Fehler des gesamten Designs, sowie Dimensionierungsregeln einzelner Gatter, um funktionalen Fehlern entgegen zu wirken.

Ziel dieser Arbeit ist es, Verschleißerscheinungen aufgrund von Gateoxiddefekten auf Gatterebene zu simulieren. Der hier vorgestellte GOB-Simulator unterscheidet sich von vorhergehenden Arbeiten in der Art, dass Gattern nicht pauschal, sondern individuell nach definierten Kriterien eine Fehlerwahrscheinlichkeit zugeordnet wird. Damit ist es möglich, Schwachpunkte innerhalb eines Designs aufzuzeigen. Des Weiteren wird neben der funktionellen Analyse auch die Veränderung der Verzögerungszeit gatterweise ermöglicht, um den Einfluss von hard und soft Breakdowns über die Zeit untersuchen zu können. Um dies zu erreichen, wurden Transistormodelle von Defekten mit einem Spice-Simulator analysiert, die als Datenbasis für den GOB-Simulator dienen. Dieser Simulator operiert dann auf Gatterebene und erstellt Spannungspegelreports über die Netze eines vorgegebenen Designs. Außerdem wird eine Gatterbibliothek generiert, damit das Zeit- und Stromaufnahmeverhalten auch mit Standard-Designtools analysiert werden kann. Damit ist es

möglich Einflüsse durch Alterung und Verschleiß auf Spannungspegel, das statische Zeitverhalten und die Stromaufnahme innerhalb des Designs nachzuvollziehen.

Da Gateoxiddefekte die Stromkurven der Transistoren verändern und damit ihr Zeitverhalten, war es nahe liegend, das Current-Source Modell (CSM) für die Gatterbibliotheken zu verwenden [15]. Im Gegensatz zum Non-Linear Delay Modell (NLDM), bei dem das Zeitverhalten mittels Flanken- und Verzögerungstabelle dargestellt wird, besteht der Tabellenoutput beim CSM aus Stromkurven (mit mehreren Zeit/Strom-Paaren), die nichtlineare Stromquellen darstellen (Treibermodell). Daraus kann mit Hilfe eines Receiver-Modells (Eingangskapazität) unter Einbeziehung des Leitungsmodells die Spannungskurve und damit die Verzögerungszeit sowie die Flanken einzelner Gatter generiert werden. Als Input dienen dementsprechend die Eingangsflanke und die Ausgangskapazität. Der Spannungswert zu einem bestimmten Zeitpunkt wird anhand der Spannung des vorherigen Zeitpunktes, des Stromflusses zwischen beiden Zeitpunkten und der Lastkapazität ermittelt. In Abbildung 1 sind beiden Varianten gegenübergestellt. Der größere Speicherbedarf des CSM wird mit der flexibleren Handhabung aufgewogen, da z.B. nichtlineare Skalierungen der Versorgungsspannung und der Temperatur unterstützt werden [16].

In Kapitel 2 wird der Ablauf der Gattersynthese dargestellt und wie sich der GOB-Simulator in den Ablauf einfügt. Danach werden die grundlegenden Transistorsimulationen erläutert. Kapitel 4 befasst sich dann mit der Funktionsweise des programmierten GOB-Simulators. Ergebnisse von Testläufen und Vergleiche mit reinen Spice-Simulationen werden in Kapitel 5 präsentiert bevor abschließend eine Zusammenfassung diesen Beitrag beendet.

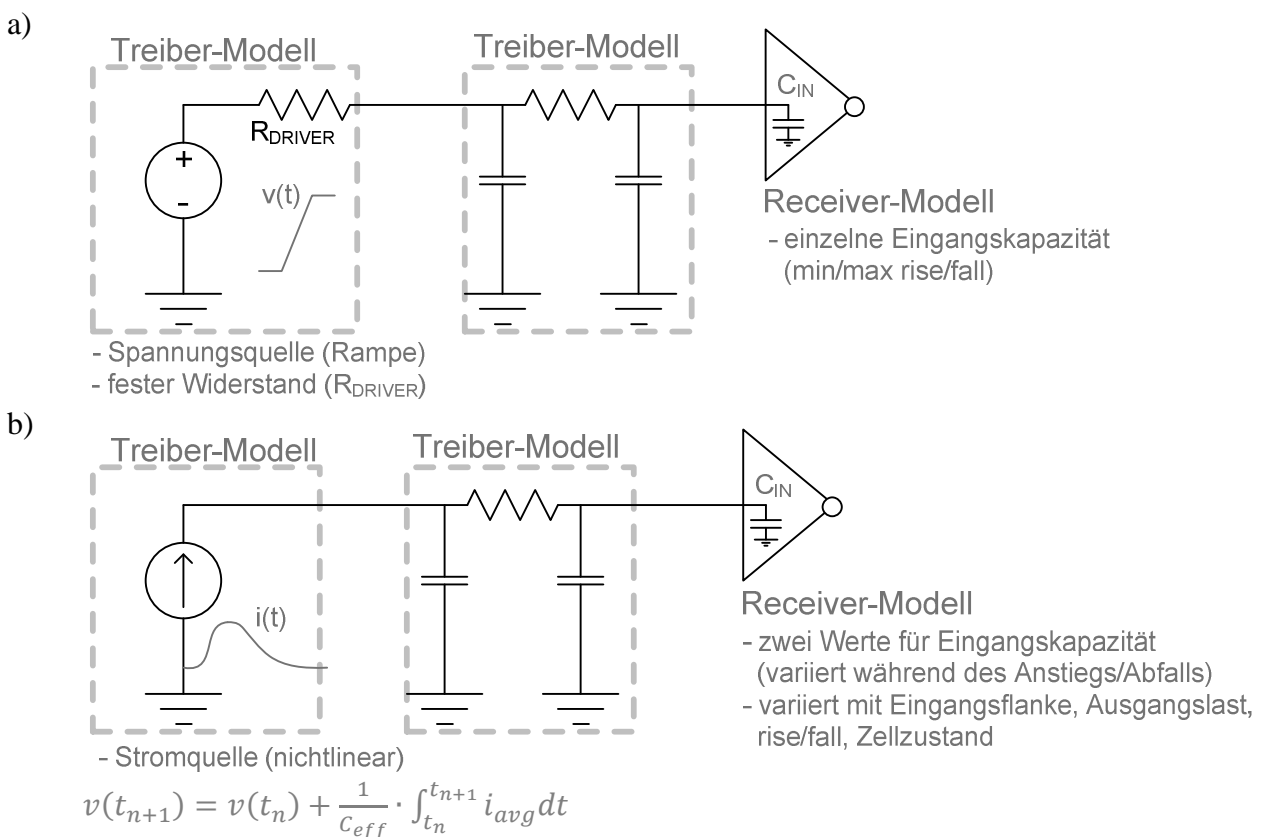


Abbildung 1 Modelle zur Berechnung der Spannungen einzelner Netze

a) Non-Linear Delay Modell(NLDM)

b) Composite Current Source Modell (CCS)

2. Ablauf der Gattersynthese mit Berücksichtigung von GOB-Simulationen

Da der Ansatz dieser Arbeit auf der logischen Synthese von Standardgattern fußt, wird diese kurz erläutert. Auf Basis einer gegebenen Gatterbibliothek und spezifischer Vorgaben bezüglich Verzögerungszeit, Stromaufnahme und Fläche (design constraints) wird eine Design-Beschreibung (die in der Regel als Register-Transfer-Logik (RTL) vorliegt) auf konkrete Standardgatter der Bibliothek übertragen, um die logische Funktion auf Gatter abzubilden. Die Gatterbibliothek beinhaltet neben der logischen Funktion auch Angaben zum statischen Zeitverhalten, zur Stromaufnahme und zum Flächenverbrauch der einzelnen Gattertypen. Während und nach der Erstellung des Designs erfolgt eine statische Verzögerungszeitanalyse (Static Timing Analysis – STA), um zu gewährleisten, dass Vorgaben bezüglich des Zeitverhaltens eingehalten werden. Wie in Kapitel 1 erwähnt, wird dazu in dieser Arbeit das Current-Source Modell (CSM) verwendet, im speziellen das Composite Current Source (CCS) Timing-Modell, da mit Tools von Synopsys™ gearbeitet wurde. Im Allgemeinen erfolgen nach der Synthese erneut funktionale Simulationen mit vordefinierten Eingangsmustern, um zu testen, ob die logische Funktion korrekt auf die Gatter

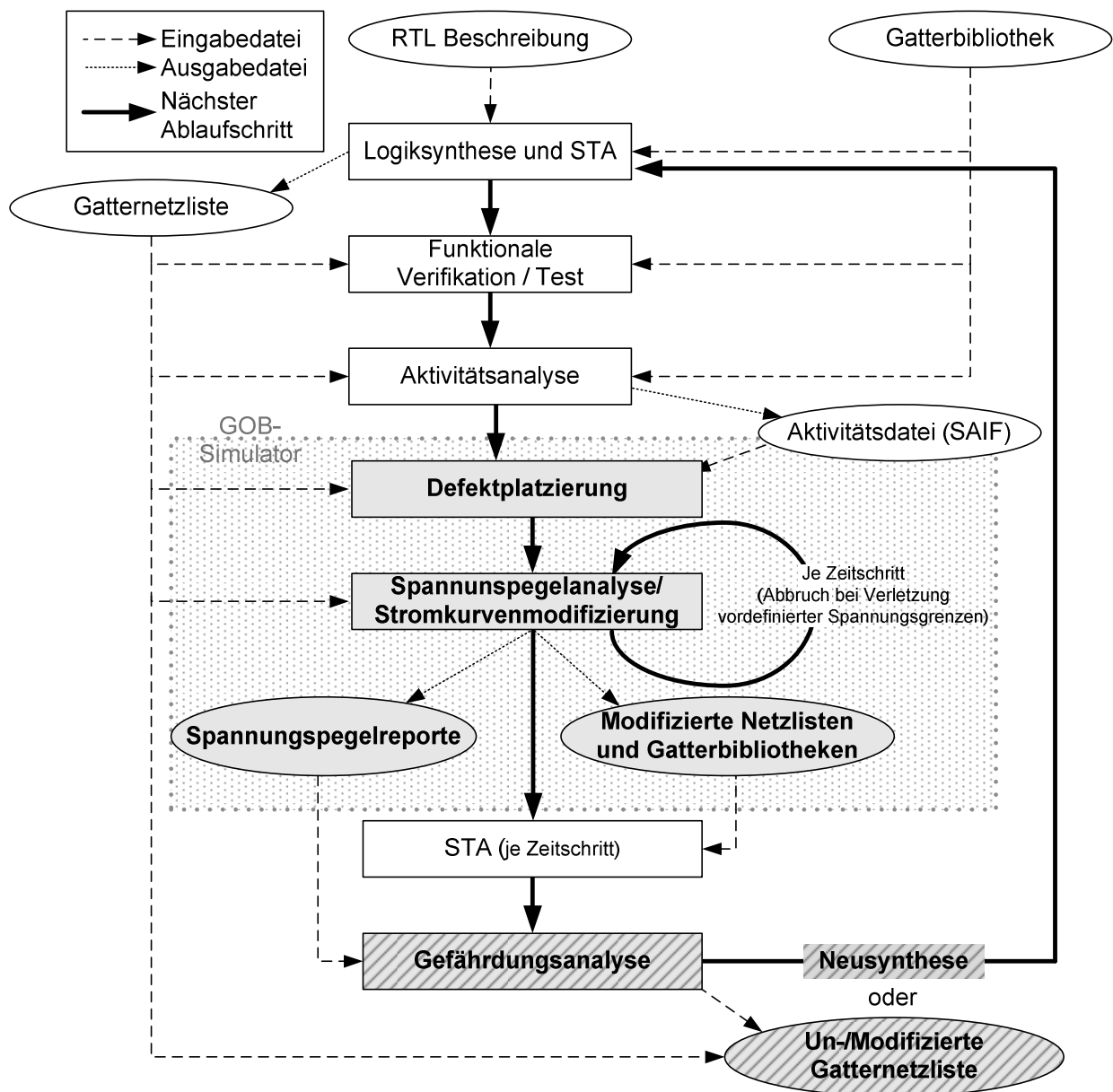


Abbildung 2 Kompletter Syntheseablauf mit Originalablauf und den vorgestellten Erweiterungen (GOB-Simulator, Gefährdungsanalyse)

abgebildet wurde. Diese Simulationen werden außerdem dazu verwendet, Aktivitätsdateien (Switching Activity Interchange Format – SAIF-Dateien) zu erstellen, mit denen konkretere Angaben zur dynamischen Stromaufnahme berechnet werden können, da diese Dateien Informationen über den logischen Status (0, 1, X) einzelner Netze über den gesamten Simulationslauf erfassen. Schließlich folgen dann weitere Schritte (Layout, DRC, LVS ...), um das Design in den Fabrikationsprozess zu übergeben.

Wie in Abbildung 2 ersichtlich, fügt sich unser GOB-Simulator (grau unterlegte Komponenten) nach der erstmaligen Synthese des Designs in den Designablauf, also wenn Netzliste und Aktivitätsdatei erstellt worden sind. In der Abbildung, die eine zusammenfassende Übersicht über den erweiterten Syntheseflow gibt, sind die einzelnen Stationen als Rechtecke dargestellt, Dateien als Ellipsen. Der GOB-Simulator, welcher in Kapitel 4 näher erläutert wird, führt eine Art Schwachstellenanalyse durch, um potentielle Probleme aufzuzeigen, die durch Verschleißerscheinungen hervorgerufen werden können. Dabei werden für einen definierten Zeitraum der Verschleißphase Spannungspegelberichte und CCS-Bibliotheken generiert. Basierend auf diesen Ergebnissen kann die Einhaltung logischer Spannungspegel kontrolliert und das statische Zeitverhalten analysiert werden.

Wenn mehrere GOB-Simulationsläufe durchgeführt worden sind, ist es außerdem möglich eine Art statistisches Zeitverhalten der Schaltung zu erstellen, wobei dann kritische Pfade sichtbar werden, die ohne Schwachstellenanalyse nicht bekannt wären. Das genauere Ergebnis kann durch den Designer genutzt werden, um das Design neu zu synthetisieren oder bestimmte Stellen im Design zielgerichtet zu verändern. Mögliche Ansätze sind dabei entweder auf der RTL-Ebene oder innerhalb der Gatternetzliste, in dem man z. B. Treiberstärken oder andere Gattertypen (High-, Low- V_{TH}) einsetzt, die die Schaltung besser gegenüber Verschleißproblemen durch Gateoxiddefekte schützen. Diese und weitere Reaktionen (schräg schraffiert in Abbildung 2) auf die Ergebnisse des GOB-Simulators sind aber nicht Bestandteil dieser Arbeit und werden deshalb nicht näher erläutert.

3. Transistorlevel Simulationen als Ausgangspunkt

Ähnlich der grundlegenden Simulationen für den Aufbau von CCS-Datenbanken wurde eine Auswahl an Gattern mit Gateoxiddefekten simuliert, die als Basis für den GOB-Simulator dienen.

Transistormodell für Gateoxid-Breakdown

Als Schadensmodell wurde das Gateoxiddefektmodell von Renovell verwendet [17], welches in Abbildung 3 dargestellt ist. Der punktuelle Defekt an einer bestimmten Stelle des Gateoxides wird mit Hilfe von zwei parallel geschalteten Transistoren nachgebildet, indem der Stromfluss (I_{GS} , I_{DS}) des schadhafte Transistors mit Transistorbreite w_0 durch Anpassen der Modell-Transistorbreiten w_1 , w_2 und w_3 modelliert wird. Neben der Position des Defekts stellt der hinzugefügte Widerstand R_{BD} zwischen den Drainanschlüssen und den Gates der Transistoren eine Möglichkeit dar die Größe des Stromflusses durch das Gateoxid anhand eines Parameters von einem geringen (bei hochohmigen Defekten) bis zu einem starken Schaden ($R_{BD} \rightarrow 0$) zu verändern. Damit ist es

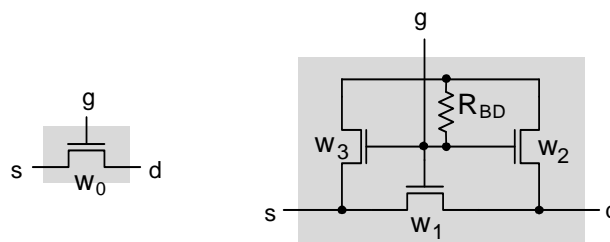


Abbildung 3 Modell eines Gateoxiddefektes nach Renovell (g-Gate; s-Source; d-Drain)

möglich, Gateoxiddefekte an einer beliebigen horizontalen und vertikalen Stelle zwischen Gate- und Bulkanschluss mit entsprechend frei wählbarem Defektwiderstand R_{DEF} abzubilden. Dabei ergibt sich R_{DEF} aus R_{BD} und den Widerständen der Transistoren, welche je nach R_{BD} und Pegel am Gate g einen größeren oder nur minimalen Einfluss auf R_{DEF} haben.

Spice-Simulationen

Um nun Daten für eine CCS-Modellierung zu erhalten, wurden etliche Spice-Simulationen mit grundlegenden Gattern (wie beispielsweise Inverter, 2/3-fach NAND, 2/3-fach NOR, welche zusammengefasst auch mehrstufige Gatter darstellen können) in verschiedenen Treiberstärken durchgeführt, um relevante Daten für die Gatterlevel-Simulationen zu erhalten. Ausgehend vom Single-Input-Switching Verhalten für die Gatterlevel-Bibliothek, wurde jeweils ein Transistor des Gatters als defekt deklariert und dessen Eingang geschaltet. Die resultierende Stromkurve am Ausgang (die essentiell für die CCS-Modellierung ist), sowie die Stromwerte am Eingang zu bestimmten Spannungswerten (welche zur Anpassung der Stromkurve im GOB-Simulator notwendig sind) wurden gespeichert. Des Weiteren sind die An-/Abstiegszeiten der Flanken am Eingang (zur Berechnung der Eingangskapazität) als auch am Ausgang zur Bestimmung spezifischer Parameter im GOB-Simulator von Relevanz. Alle Simulationen wurden mit verschiedenen Parametern (wie Treiberstärke, Spannungswerte und Kapazitäten am Eingang, Lastkapazität) und unterschiedlichen Defektwiderständen ausgeführt, was zu mehreren zehntausend Parameterkombinationen pro Eingang eines Gatters führt.

4. Ablauf eines GOB-Simulationslaufes

Voraussetzung für den Ablauf des GOB-Simulators sind die vorhergehenden Spice-Simulationen, aus denen eine Datenbank erstellt wurde, mit der der GOB-Simulator das Verhalten eines gealterten Schaltkreises, bestehend aus dem verfügbaren Set an Gattern, nachbilden kann. So ist es zum einen möglich, Simulationen auf Gatterebene durchzuführen, die zeitabhängige Spannungsverläufe an einzelnen Netzen aufzeigen. Zum anderen können auch modifizierte Netzlisten und Gatterbibliotheken erstellt werden, die dann vom Synthese-/Analysetool genutzt werden können, um Zeit- und Stromaufnahmeberichte anzufertigen bzw. die Synthese oder die Implementierung an bestimmten Stellen im Design anzupassen.

Ein Durchlauf des GOB-Simulators gliedert sich in eine Defektplatzierung und eine Analyse-schleife, bei der für bestimmte Zeitschritte die genannten Netzlisten und Gatterbibliotheken erzeugt werden, sowie die Spannungspegel aller Netze berechnet werden. Die dritte Phase der Simulation ist dann die Anwendung des STA-Tools. Vor diesen Schritten wurde einmalig mit Hilfe der Spice-Datenbank eine originäre Bibliothek erstellt, die das ursprüngliche Zeitverhalten (CCS-Stromkurven), die Eingangskapazitäten und die originale Stromaufnahme für alle Gatter abbildet bei denen noch keine Verschleißerscheinungen auftreten. Im Folgenden werden die Defektplatzierung und die Analyseschleife näher erläutert.

Einfügen von Defekten

Ausgehend von der Übernahme der Daten einer vorhandenen, zu analysierenden Netzliste und einer dazugehörigen Aktivitätsdatei werden intern Gatter- und Netzlisten erstellt, die jede Instanz der Netzliste umfassen. Dann werden Defekte innerhalb der Netzliste platziert. Dies erfolgt quasi auf Transistorebene, indem jedem Transistor eine Verteilungsfunktion zugewiesen wird, dessen Verlauf über verschiedene Parameter definierbar ist. Zunächst stehen vier grundlegende Wahrscheinlichkeitsverteilungen zur Verfügung: Gleich-, Normal-, Exponential- und Weibullverteilung. Da die Normal- und die Exponentialverteilung auch mit Hilfe der Weibullverteilung dargestellt werden können und weil die Weibullverteilung für

Verschleißerscheinungen sehr verbreitet ist, ist sie als default voreingestellt. Ausgehend von der Weibullgrundgleichung [18] ergibt sich für die kumulative Verteilungsgleichung $F(t)$:

$$F(t) = 1 - e^{-\left(\frac{t}{ttf}\right)^\beta} \quad (1)$$

wobei β der Weibullfaktor, t die Zeit und ttf (ttf – time-to-failure) die charakteristische Lebensdauer (bei einer Ausfallwahrscheinlichkeit von 63 %) ist.

Folgende Parameter beeinflussen die Lebensdauer ttf entscheidend [18]:

- Gatefläche des Transistors: $ttf \propto \left(\frac{1}{W \cdot L}\right)^{1/\beta}$ (2)

- Gateoxiddicke des Transistors T_{OX} : $ttf \propto T_{OX}$ (3)

- Temperatur T : $ttf \propto e^{\left(\frac{c}{T} + \frac{d}{T^2}\right)}$ (4)

- Spannung über dem Gate V_{GS} : $ttf \propto V_{GS}^{a+bT}$ (5)

wobei W der Transistorgatebreite, L der Transistorgatelänge, sowie a , b , c und d technologieabhängigen Parametern entsprechen. Da die GS-Spannung über die Zeit entweder V_{DD} oder V_{SS} entspricht, kann man die Lebensdauer abhängig von der Wahrscheinlichkeit P_{ON} darstellen, also wie oft der Transistor durchgeschaltet ist (angegeben als relative Größe). Diese Daten kann man der Aktivitätsdatei entnehmen. Die Lebensdauer wird entsprechend der aktuellen Gegebenheiten jedes einzelnen Transistors angepasst (z.B. an die Größe $W \cdot L$). Danach wird $F(t)$ ein Zufallswert zwischen 0 und 1 zugeordnet. Aus Gleichung (1) ist es dann möglich einen Zeitpunkt $t = t_{BD}$ für jeden Transistor zuzuordnen, bei dem dieser einen Defekt für den einzelnen Analyselauf ausbildet. Die Veränderung des Widerstandes R_{BD} folgt dann dem exponentiellen Anstieg des Stromes I_{BD} durch den Defekt über die Zeit t [20]:

$$I_{BD} = I_{t=t_{BD}} \cdot e^{\left(\frac{t-t_{BD}}{G}\right)} \quad (6)$$

mit G als Wachstumszeit, die sich aus der Lebensdauer und technologischen Parametern errechnet. Damit ist es aufgrund der gewählten CCS-Modellierung möglich jedem Transistor und damit jedem Gatter einen individuellen Defektverlauf zuzuordnen.

Wiederholter Analyseablauf

Auf die Platzierung der Defekte folgt ein Berechnungsverlauf, der für jeden zu analysierenden Zeitpunkt wiederholt wird. Dabei verläuft sowohl die Analyse zeitlich sequentiell als auch die Bearbeitung der Netzliste, da Veränderungen der Treiberzelle sich auf die nachfolgenden Gatter auswirken. Ausgehend von den Designeingängen werden für jedes Gatter (und jeden Eingang) die Stromkurven und Spannungspegel am Ausgang berechnet. Wenn das gesamte Design durchlaufen wurde und kein Spannungspegeleinbruch über voreingestellte Grenzen festgestellt wurde, werden die Netzliste und die Gatterbibliothek für diesen Zeitpunkt erstellt. Beim Übergang zum nächsten Zeitpunkt werden Defekte der vorherigen Stufe übernommen und dem Zeitpunkt entsprechend modifiziert. Gegebenenfalls werden neue Defekte hinzugefügt.

Die einzelne Gatteranalyse erfolgt folgendermaßen. Als erstes wird geprüft, ob das treibende Gatter schon analysiert wurde. Falls nicht, werden die Berechnungen für das aktuelle Gatter später ausgeführt. Der zweite Schritt ist die Berechnung der minimalen und maximalen Pegel (V_{MIN} , V_{MAX}), die am Ausgang erreicht werden können, gefolgt von der Modifizierung der Stromkurven. Sollten mehrere Inputpins vorhanden sein, werden die schlechtesten Werte der Pegelberechnung für den Ausgang angenommen. Die Pegel am Ausgang sind abhängig von den maximalen Inputpegeln, vom eigenen Defekt und von Schäden an Transistoren der Gatter, die über das Ausgangsnetz mit dem aktuellen Gatter verknüpft sind. Der Minimalpegel V_{MIN} , der idealerweise V_{SS} entsprechen

Tabelle 1 Auswirkungen bestimmter Umgebungsparameter auf die minimalen und maximalen Spannungspegel; mit negativem (-), positivem (+) oder keinem Einfluss (o)

Auswirkungen auf das Ausgangsnetz:	V_{MIN}	V_{MAX}
Minimaler Eingangspegel $> V_{SS}$	(o)	sinkt (-)
Maximaler Eingangspegel $< V_{DD}$	steigt (-)	(o)
Eigener Defekt	steigt (-)	sinkt (-)
Defekt eines p-MOSFET am Ausgang	steigt (-)	steigt (+)
Defekt eines n-MOSFET am Ausgang	sinkt (+)	sinkt (-)

sollte, wird anhand der fallende Flanke des aktuellen Gatters berechnet. Negativ beeinflusst wird er durch folgende Parameter:

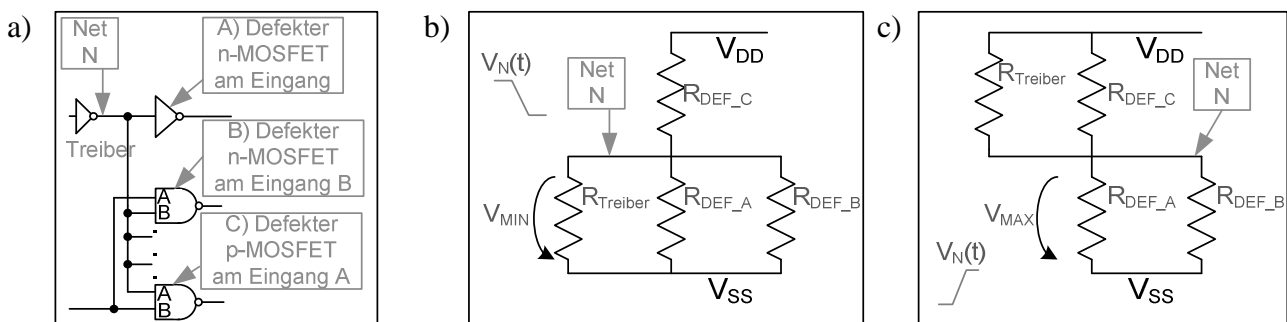
- ein Maximalpegel am Eingang, mit $V_{MAX} < V_{DD}$, d.h. n-MOSFET nicht komplett offen
- ein Defekt am n-MOSFET oder p-MOSFET des aktuellen Gatters und Pins
- p-MOSFET – Schäden nachfolgender Gatter, da sie das Netz mit V_{DD} verbinden können

Nachfolgende n-MOSFET-Defekte ziehen den Pegel wieder Richtung V_{SS} , was den Minimalpegel positiv beeinflussen würde. Allerdings sind die Auswirkungen nicht so groß wie beim Defekt eines vergleichbaren p-MOSFET, da der Widerstand des Defektes aufgrund des fast geschlossenen Transistors sehr groß ist. Diese Effekte kehren sich bei der Berechnung von V_{MAX} um, da hier die steigende Flanke entscheidend ist. Hierbei ist dann am Eingang auch der Minimalpegel des Vorgängers entscheidend. Tabelle 1 zeigt die positiven und negativen Effekte auf, die bei mehreren Defekten an einem Netz auftreten können, wie es in Abbildung 4 a) dargestellt ist. Abbildung 4 b) und c) zeigen vereinfacht, wie diese Defekte die Pegel im statischen Fall beeinflussen, in der Art eines elektrischen Ersatzschaltbildes, wobei die Schäden am Gate durch Widerstände ersetzt wurden. Da die Daten auch in Form von Strömen zu einer bestimmten Spannung vorliegen, ergeben sich folgende Gleichungen für die Ströme, um V_{MIN} und V_{MAX} zu berechnen:

$$I_{TREIBER}(V_{MIN}) = \sum_{\forall p\text{-MOSFET-Defekte}} I_{DEF}(V_{MIN}) - \sum_{\forall n\text{-MOSFET-Defekte}} I_{DEF}(V_{MIN}) \quad (7)$$

$$I_{TREIBER}(V_{MAX}) = \sum_{\forall n\text{-MOSFET-Defekte}} I_{DEF}(V_{MAX}) - \sum_{\forall p\text{-MOSFET-Defekte}} I_{DEF}(V_{MAX}) \quad (8)$$

Die Ströme I_{MIN} und I_{MAX} , die zu diesen Spannungszuständen fließen, werden ebenfalls errechnet, genauso wie bei Spannungen zwischen diesen Pegeln. Anhand dieser Spannungs-/Strompaare wird die ursprüngliche Stromkurve des Gatters am Ausgang modifiziert. Dies ist notwendig, da die Auf-/Entladung der Ausgangskapazität, was durch die Stromkurve dargestellt wird, durch die Defekte verlangsamt/beschleunigt wird, da der Auf-/Entladestrom durch die Gateschäden verringert beziehungsweise erhöht wird. Die Einflussparameter sind dabei entsprechend denen der Pegel V_{MIN} und V_{MAX} . Außerdem stellen I_{MIN} und I_{MAX} statische Leckströme dar, die auf den untersuchten Netzen fließen, wenn das Gatter nicht schaltet. All diese Effekte werden bei der Erstellung der



*Abbildung 4 a) Netz (N) mit mehreren defekten Transistoren
Auswirkung auf die b) minimal und c) maximal erreichbaren Spannungspegel*

Gatterbibliothek berücksichtigt, wobei Gatter mit modifizierten Stromkurven als neue Gatter aufgenommen werden. Deshalb wird auch die Gatternetzliste des Designs modifiziert, damit diese Zellen bei der späteren STA Berücksichtigung finden. Dementsprechend wird ein Gatter also auch dann als neues, defektes Gatter aufgenommen, wenn es nur mit schadhafte Gattern am Ausgang verbunden ist, da dessen Stromkurve beeinflusst wird. Die Gatterbibliothek und -Netzliste wird nur dann erstellt, wenn die Pegel V_{MIN} und V_{MAX} bestimmte vordefinierte Grenzpegel nicht über- bzw. unterschreiten damit noch ein ausreichender Spannungshub vorhanden ist. Wenn dies nicht der Fall sein sollte, wird das Design als defekt deklariert.

5. Vergleich der Resultate des GOB-Simulators im Vergleich zu Spice-Simulationen

Um die Ergebnisdateien des GOB-Simulators (Spannungspegelreport, Gatternetzlisten/-bibliotheken) zu testen, wurden sie gegen die Spice-Simulationen getestet. Aufgrund der Komplexität der Berechnungen bestand das erste Design aus Invertern in einer zehnstufigen Reihe (Test_1). Die Tests 2 bis 5 umfassen jeweils Inverter gleicher Treiberstärke, die in einem achtstufigen Baum angeordnet wurden. Des Weiteren wurden zwei-Eingangs NAND- und NOR-Gatter in den Tests 6 bis 8 mit Invertern in einem sechsstufigen Baum zusammengefasst. Den Abschluss bildeten dreifach NAND- und NOR-Gatter, die zusammen mit Invertern jeweils in einem achtstufigen Baum simuliert wurden (Tests 9 bis 11). Bei den Tests 6 bis 11 wurden teilweise Schaltungseingänge konstant gehalten während andere ständig schalteten, damit die Ausgänge aller Gatter immer schalteten. Die Zeitskala ist in relative, dimensionslose Zeitpunkte unterteilt, wobei ein GOB zu jedem Zeitpunkt $t > 0$ auftreten kann.

Abbildung 5 a) und b) zeigen die Ergebnisse für den Vergleich der mit dem GOB-Simulator errechneten Minimal-/Maximalpegel in Bezug zu den mit Spice simulierten Ergebnissen. Dargestellt sind die Ergebnisse der Testdesigns 5 (a) und 8 (b), welche exemplarisch für alle Testdesigns anzusehen sind. Um die Vielzahl der Simulationsläufe gut gegeneinander zu vergleichen, wurden alle Abweichungen in Streuungsquartilen zusammengefasst, einmal für „alle“ Simulationen über die Gesamtheit der Zeitpunkte und dann jeweils für bestimmte Zeitschritte, wobei ein weibullverteilter Verschleiß angenommen wurde, so dass mit größer werdenden

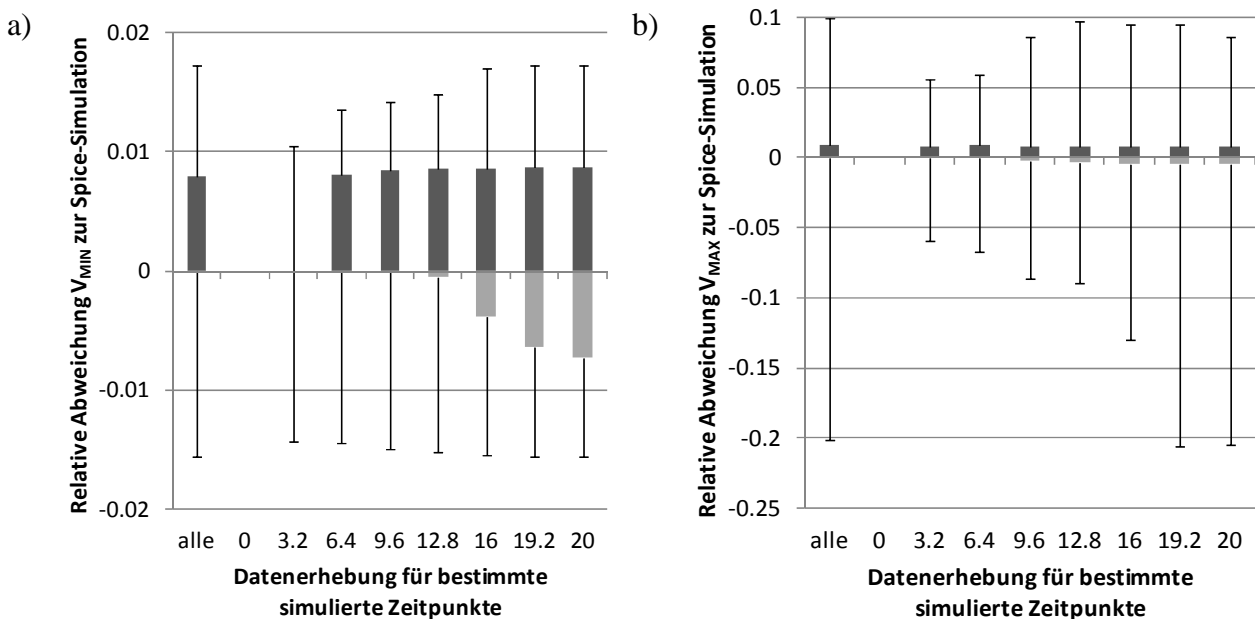


Abbildung 5 Streuung der Abweichungen der mit dem GOB-Simulator errechneten Minimal- und Maximalpegel V_{MIN} (a) und V_{MAX} (b) von den Spice-simulierten Resultaten – bezogen auf $(V_{MAX} - V_{MIN})$ der Spice-Simulationen

Zeitschritten die Anzahl und das Ausmaß der Gateoxiddefekte zunimmt. Der Median befindet sich durchgängig bei 0 ± 0.01 . Zudem ist ersichtlich dass ein Großteil der errechneten Pegel nur minimal von den simulierten abweicht. Zum Beispiel ist in Abbildung 5 a) zu sehen, dass 50 % der Werte (Interquartilsabstand IQR) zwischen $Q_{.25}$ (unteres Ende des grauen Balkens) und $Q_{.75}$ (oberes Ende des schwarzen Balkens) weniger als 1 % von den Spice-simulierten Werten abweichen. Und selbst die Maximalwerte liegen im Bereich von $\pm 2\%$. In den Testdesigns 6 bis 11 – exemplarisch in Abbildung 5 b) mit den Maximalpegeln des Testdesigns 8 – sind die Abweichungen etwas stärker, allerdings befindet sich auch hier der Großteil in einem Bereich von $\pm 2\%$. Die sehr starken Maximalabweichungen von +10 % bis -20 % entstehen zum größten Teil aufgrund der Tatsache, dass der GOB-Simulator bei Überschreiten eines bestimmten Limits den Wert von $(V_{MAX} - V_{MIN})/2$ annimmt, obwohl der Spice-Simulator den Pegel noch gerade so innerhalb der Grenze errechnet. Dies ist allerdings in sehr seltenen Fällen geschehen, was aus Abbildung 6 ersichtlich wird, wo die durchschnittlichen Abweichungen aller Designs für alle Simulationen zum Zeitpunkt $t = 9.6$ als Beispiel dargestellt sind. Nur bei einem Testdesign liegt sie außerhalb von $\pm 1\%$, so dass man festhalten kann, dass die Spannungspegel sehr exakt vom GOB-Simulator errechnet werden.

Ergebnisse zum Zeitverhalten sind in Abbildung 7 dargestellt. Hier werden die Durchschnittswerte für spezifische Zeitpunkte über alle Simulationsläufe des Designs test_2 aufgezeigt. Zu sehen ist die immer vorhandene Abweichung der STA (mit Synopsys™ DesignCompiler™ errechnet) von den Spice-Simulationen, da die Tools vom schlechtesten Fall (worst case) ausgehen und das Schalten der nächsten Stufe später als mit Spice erfolgt, wo schon mit Überschreiten der Schwellspannung am Eingang der nächsten Stufe das Schalten dieser Stufe beginnt. Allerdings ist ebenso gut zu erkennen, dass die Analyse der erstellten Gatterbibliotheken, dem Verlauf der sich verschlechternden Verzögerungszeitkurve der Spice-Simulationen folgt, was gut den Verschleiß der Schaltung widerspiegelt.

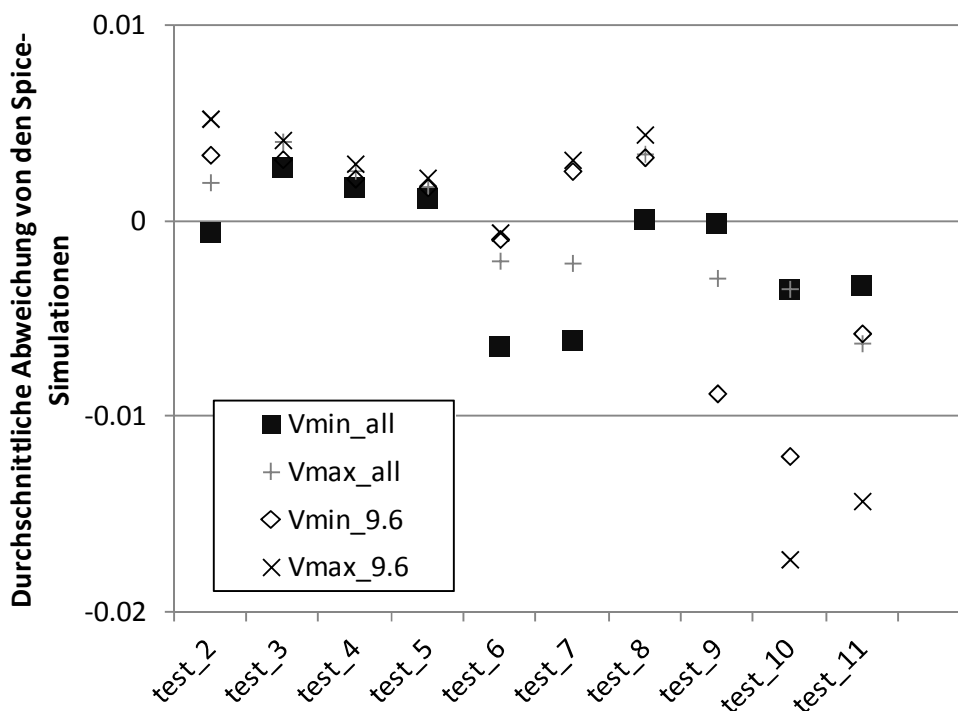


Abbildung 6 Durchschnittliche Abweichungen der berechneten Spannungspegel von den mit Spice simulierten Spannungspegeln einzelner Netze

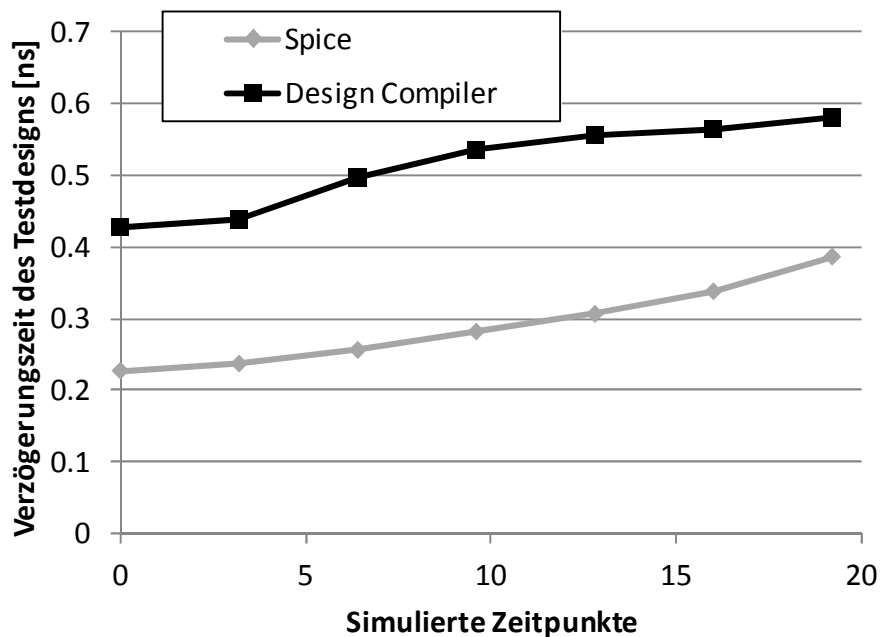


Abbildung 7 Durchschnittliche simulierte Verzögerungszeit (test_2) mit Beginn des Verschleißes bei $t = 0$

Zusammenfassung

Dieser Beitrag identifiziert die Notwendigkeit Verschleißeffekte bereits im Design zu berücksichtigen. Deshalb wurde ein Ansatz vorgestellt, der Simulationen auf Gatterebene einsetzt, um Defekte und deren Einfluss realistisch aufzuzeigen. Die dargestellten Ergebnisse zeigen eindeutig, dass es möglich ist, Modelle aus der Transistorebene in die Gatterebene mittels eines GOB-Simulators zu transferieren und die Simulationen in den Syntheseablauf zu integrieren. Die erzielten Ergebnisse für die Spannungspegel weichen dabei trotz der deutlich höheren Abstraktion im Durchschnitt nur um maximal 2% von den Spice-basierten Transistorsimulationen ab. Des Weiteren werden die Verläufe der Verzögerungszeiten durch die CCS-Bibliotheken sehr gut nachgebildet. Der vorgestellte Ansatz ist nicht allein auf GOB ausgerichtet und bietet somit die Grundlage auch weitere Verschleißeffekte effizient in den Ablauf zu integrieren. Zukünftige Erweiterungen konzentrieren sich darauf die Berechnungskomplexität zu untersuchen und wie anhand der gewonnenen, zusätzlichen Daten identifizierte Schwachstellen eines Designs zielgerichtet verbessert werden können.

Literatur

- [1] G Srinivasan, J., Adve, S., Bose, P. and Rivers, J.: "The Impact of Technology Scaling on Lifetime Reliability", In Proc. of Dependable Systems and Networks, 2004.
- [2] Mc Pherson, J.W.: "Reliability trends with advanced CMOS scaling and the implications for design", In Proc. of Custom Integrated Circuits Conference, 2007.
- [3] Stathis, J.: "Reliability Limits for the Gate Insulator in CMOS Technology", In IBM Journal of Research & Develop, 2002.
- [4] Crook, D.: "Method of Determining Reliability Screens for Time Dependent Reliability Breakdown", In Proc. Reliability Physics, 1979.
- [5] Lombardo, et al.: "Dielectric breakdown mechanisms in gate oxides", In Journal of Applied Physics 98, 2005.

- [6] Kaczer, B. et al.: "Gate Oxide Breakdown in FET devices and circuits: From nanoscale physics to system-level reliability", In *Microelectronics Reliability*, 2007.
- [7] Tu, R. et al.: "Berkeley Reliability Tools – BERT", In *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1993.
- [8] Renovell, M., Gallière, J., Azaïs, F., Bertrand, Y.: "Delay Testing of MOS Transistor with Gate Oxide Short", In *Proc. of ATS*, 2003.
- [9] Saluja, K., et al.: "NBTI degradation: A problem or a scare?", In *Proc. of IEEE International Frequency Control Symposium & Exposition*, 2008.
- [10] Wu, L. et al.: "GLACIER: A Hot Carrier Gate Level Circuit Characterization and Simulation System for VLSI Design", In *Proc. of First International Symposium on Quality of Electronic Design*, 2000.
- [11] Li, P., Stamoulis, G., Hajj: "Probabilistic timing approach to hot-carrier effect estimation", In *Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 1994.
- [12] "Reliability Simulation in Integrated Circuit Design", technical white paper, Cadence™, 2003.
- [13] Synopsys™ DesignCompiler™ Manual, 2011.
- [14] Fang, J., Sapatnekar, S.S.; , "Scalable methods for the analysis and optimization of gate oxide breakdown," In *Proc. of 11th International Symposium on Quality Electronic Design (ISQED)*, 2010.
- [15] Knoth, C. et al.: "Characterization and implementation of nonlinear logic cell models for analog circuit simulation," In *Proc. of Integrated Circuits*, 2009.
- [16] "CCS Timing", technical white paper, Synopsys™, 2006.
- [17] Renovell, et al.: "Modeling the Random Parameters Effects in a Non-Split Model of GOS", In *Journal Electronic Testing*, 2003.
- [18] Koren, I.; Krishna, M.: "Fault-Tolerant Systems," Morgan-Kaufmann, San Francisco, 2007.
- [19] Xiaojun Li: "Deep Submicron CMOS VLSI Circuit Reliability modeling, simulation and design", Dissertation, 2005.
- [20] Lindner, B. et al.: "Growth and scaling of Oxide Conduction after Breakdown", In *Reliability Physics Symposium Proc.*, 2003.