

Verification of Safe Operating Area (SOA) Constraints in Analog Circuits

Udo Sobe¹, Karl-Heinz Rooch¹, Dietmar Mörtl¹, Achim Graupner¹, André Lerch² and Michael Pronath³

¹ZMD AG

Grenzstraße 28

01109 Dresden

{Udo.Sobe, Karl-Heinz.Rooch,

Dietmar.Moertl, Achim.Graupner}@zmd.de

²X-FAB Semiconductor Foundries AG

Haarbergstraße 67

99097 Erfurt

Andre.Lerch@xfab.com

³MunEDA GmbH

Stefan-George-Ring 29

81929 München

Michael.Pronath@muneda.com

Abstract

The modeling, simulation and their effects of aging (e.g. negative bias temperature instability (NBTI), hot carrier stress (HC)) have been a topic of research for nearly two decades. These reliability models and corresponding simulation techniques are not yet widely available in current process development kits (PDK) and design flows. In general, process documentation contains information about safe operating areas (SOA) such as maximum current and voltage stress of the devices. How this information can be used with current tools for design, verification and optimization today is presented in this paper.

1 Introduction

The challenging lifetime standards of today's circuits for automotive and industrial applications are in the range of about 10.000 hours and about 100.000 hours respectively. Therefore to ensure such a lifetime for high voltage and temperature applications the design development is using conservative techniques for design for reliability (DfR) like safe operating areas (SOA). Such DfR methods have to be continuously available for three circuit development phases: design, verification and optimization. Here the support by the process development kits (PDK) of the FABs and their adaptation to the design flows by special departments inside a design center is important to make such methods and tools useful for circuit development.

So, it is obvious that device reliability is becoming more important within circuit design. Physical constraints limit the performance of devices. Effects like gate oxide breakthrough, negative bias temperature instability (NBTI) and hot carrier stress (HC) restrict the reliability of devices if rules are not considered.

A definition of SOA for designers is interesting to be conform to DfR. This area is defined as the voltage and current or power conditions over which the device can be expected to operate without self-damage or degradation of device parameters such as V_{th} does not lead to parametric failures during lifetime.

Substrate current itself is an undesirable concomitant of device operation and can refer to the amount of additional electrons, disturbing the proper device function by certain effects like HC, change of local substrate voltage or Latch up.

The following section explains the motivation regarding SOA technique. In addition, the importance of the substrate current is discussed.

An overview of concepts using SOA technique is given in the third section. Performance measuring, model encapsulation, device checking and constraint matrix concepts are compared in reference to design, verification and optimization of analog and mixed-signal circuits. Here we discuss especially the specific of verification and optimization. Additionally we give an estimation about the effort of setup, application and results analysis.

How the device checking feature of the Cadence Analog Design Environment (ADE) as well as the model encapsulation can be used for verification of process, voltage and temperature (PVT) corners is described in the fourth section. Here we show the application in our in-house tool `zmdAnalyser`.

2 SOA Technique

2.1 Physical SOA Motivation

Particularly automotive applications demand high chip device reliability under extreme operating conditions. We understand design for automotive applications to assure defined functionality and chip performance within a wide temperature range from -40°C up to 150°C .

Thus, one of our first intentions, with regards to device reliability, was concentrated on substrate current, which can be used as an integral monitor of process quality [1]. In this context we worked on design methods to additionally eliminate minority carriers in order to suppress uncontrolled substrate currents [2]. Later on, we optimized the design of these structures to realize an absolute prevention against latch up of CMOS structures under automotive conditions [3]. As a result, the design measure developed allows trigger currents of parasitic four-layer bipolar structures as high as 400mA at

150°C. Such parasitic bipolar structures cannot be avoided within a standard CMOS technology.

Furthermore, it is known that the intrinsic carrier density within silicon [4] doubles for every eleven temperature degrees. Another point of view is demonstrated by A. M. Abo [5]. By analyzing the breakdown and degradation phenomenon in MOS devices, he was able to show that relative terminal potential determines device lifetime. Taking into account the heating-up caused by high power density [6] at electric field maximums, beyond SOA criteria and substrate current it is ultimately the high operating temperature, which poses a design challenge.

2.2 SOA and Bulk Current

Bulk current of MOS devices is sensitive to technological changes. It can be measured by the help of a simple test configuration which is illustrated in Figure 1a. The qualitative characteristic of the bulk current I_B vs. gate voltage V_G for fixed source V_S and drain V_D voltages is shown in Figure 1b.

A tolerance tube can be defined to distinguish between specified or problematic technology conditions. Important properties are the maximum bulk current I_{Bmax} at the related gate voltage V_{Gimax} and the shape of the characteristic. These parameters are shifted outside of the specified tube if for instance the tolerance of implantation or oxide thickness are violated.

The accuracy of device models (e.g. BSIM3v3) is not focused at the range of bulk current measurement. The marked area in Figure 1b shows the range of reduced accuracy of the bulk current.

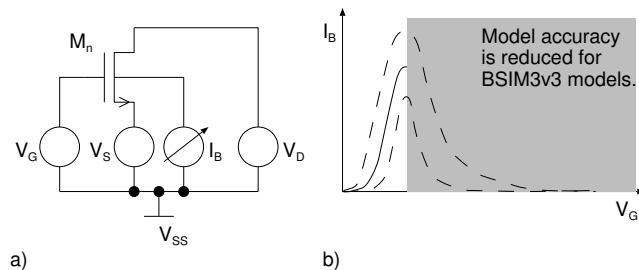


Figure 1: Test structure to measure bulk current. a) Measurement circuit. b) Schematic representation of the bulk current with tolerance tube and area of reduced model accuracy.

Analysis by Abba [1] on bulk current is more likely to show that intermediate gate voltages can be more critical. From the bell shaped curve it can be seen that the maximum bulk current usually can be found at gate voltages V_G of about 2/3 of the drain voltage V_D .

Basically, the substrate current is a result of impact ionization by the electrons flowing from the source to the drain. In this connection electron-hole pairs are generated and the holes flow to the substrate contact, hence defining the substrate current. The electrons, here the minority carriers, predominantly flow to the drain.

Unfortunately operating conditions with higher substrate current partly correspond to “safe operating area”. The

above mentioned facts emphasize remarking bulk current in a SOA concept.

3 Application of SOA

Constraints on circuit design to reduce device stress are based on current density. The documentation of current PDKs give the analog designer information about

1. Maximum voltage difference
2. Maximum path current
3. Maximum power consumption
4. Safe operating area

for every device [7]. SOA diagrams give additional parameters in general about special devices (see Fig. 3). Such rules are used during circuit design using power and high voltage devices.

The check of SOA limits has to be available within the complete flow. This means it should be supported in the following tools:

1. Design environment (e.g. Cadence’s Analog Design Environment)
2. Verification tool (e.g. ZMD in-house: zmdAnalyser)
3. Optimization tool (e.g. MunEDA’s WiCkeD)

Table 1 compares some methods to support the SOA technique. There is no method available which is supported by all tools.

General performance measurements can be used for data extraction of performance parameters (e.g. bandwidth) and of SOA parameters like maximum of drain source voltage of transistor M_1 , too. Because the definition effort by using calculator expressions is high, only few SOA parameters for selected devices are checked by this method in general.

Two other possibilities for SOA checking are the model encapsulation via SOA checker and Spectre assertion feature. Model encapsulation of every device by a SOA checker which is connected in parallel to the respective device (cf. Fig. 2) is common. A high description language (HDL) is used to define and to measure the properties. The advantage to use common HDLs like VerilogA is that the implementation is supported by different simulators. Other HDL can be used [8] but are focused to special applications (e.g. mechanical systems) and simulators (e.g. AMS).

The results are submitted into a log file and can be analyzed. Advantage of this method is the simulator independent implementation. But the results analysis, which is based on inspection of a log file, depends on the respective implementation of the scripts. There are no standards for such a log file. Some PDKs like [9] deliver additional analysis tools to filter and manage the SOA violations. These tools supports the design phase and are limited suitable for verification and optimization.

Spectre assertion is coupled to the Spectre simulator. A graphical user interface which is called Device Checking is a feature of the ADE and is utilizing Spectre assertion [10].

Table 1: Comparison of the methods. State: Cadence IC5.1.41, WiCkeD 5.2; Legend: ✓ applicable, - high, o medium, + low; Abbreviation: **Calc** Calculator, **n.a.** not available, **VD** Violation Display, **WV** Waveform Viewer

Criteria		Performance Measuring	Model Encapsulation	Device Checking	Constraint Matrix
Basis		Calculator expression	HDL script	Spectre Assertion	WiCkeD constraining
Definition	Single device	✓	✓	✓	✓
	Device type	✓	✓	✓	✓
	Structure	✓			✓
Tools	Spectre	✓	✓	✓	
	Spectre MC	✓	✓	✓	
	Spectre Verilog	✓	✓	✓	
	AMS Spectre	✓	✓		
	WiCkeD	✓	✓	✓	✓
Application	Design	✓	✓	✓	
	Verification	✓	✓	✓	
	Optimization	✓			✓
Analysis	Results format	Cadence data base	log file	Cadence data base	WiCkeD data base
	Manual	ADE WF/Calc	log file	ADE VD/Cals	WiCkeD
	Automatic	Calc/Ocean/Skill	log parser	Calc/Ocean/Skill	WiCkeD internal
Effort (Design/PDK)	Setup	-/n.a.	-/+	o/+	-/o
	Application	-/n.a.	+/+	+/+	+/+
	Analysis	-/n.a.	-/+	o/+	+/+

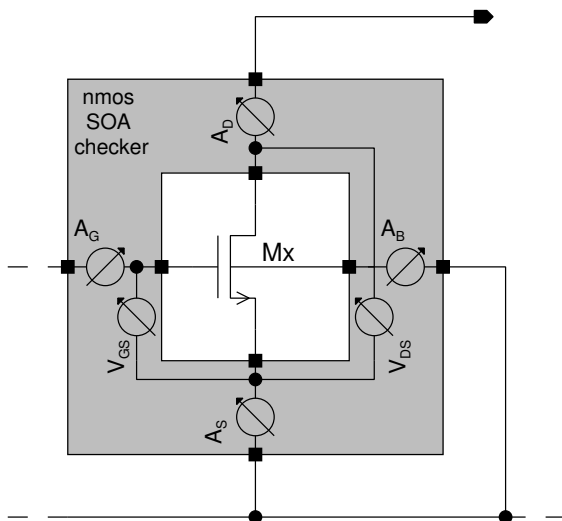


Figure 2: The principle of model encapsulation of the device models to check SOA rules in a circuit. The designer can instantiate the devices as usual. No design changes are necessary because the PDK realizes a comfortable implementation.

The Device Checking capability is integrated in the ADE and allows comfortable configuration of so-called checks. Every check contains rules, limits and options where SOAs can be defined. Advantage of this feature is the easy config-

uration via graphical user interface (GUI). Therefore a designer can define or add new checks if the PDK does not provide SOA checking tools or the provided SOA checks are not sufficient respectively.

A recently established method by ZMD [11] together with MunEDA [12] is based on the utilization of a SOA constraints matrix (cf. Fig. 3 and 4), taking into account SOA constraints during circuit optimization [13, 14].

WiCkeD [12] supports structure constraints and performance constraints which are considered during the analysis and optimization. The mathematical formulation of the rules can be prepared by PDK or designer. In general, a PDK, which supports the WiCkeD tool [15, 16], has to contain templates for structural constraints which allow automatic constraint generation. The designer can add other performance constraints manually. The example in figure 4 shows constraints for a special high voltage device.

Verification need at least the information about existence of SOA rule violations whereas the optimization demands additional details, e.g. to calculate sensitivities. This difference is the reason why the model encapsulation and the device checking is convenient for verification and regression analysis but unsuitable for optimization with a tool. The SOA constraint matrix concept is specific to the algorithm of optimization.

4 Verification

4.1 Using Device Checking

The general way to consider SOA rules with the Device Checker while circuit simulation is split into three steps:

1. Define and enable checks in the ADE Device Checking tool,
2. Carry out simulation as usual and
3. Analyze violations by ADE Violation Display tool.

This flow is convenient for design phase because of the manual inspection with the Violation Display tool. But the verification of circuits need an automated violation measuring for data extraction. Generally, ADE calculator [17] expressions are used to realize performance extraction. Violations can be determined by using the `pv` command [18].

Especially for verification or regression analysis of circuit performances it is sufficient to get information about existence of violation or count of violations. This information can be added beside the performance measurement. Details of violation e.g. type of check, device or value generate a lot of data which are more important for violation analysis and design improvement.

ZMD's in house verification environment `zmd-Analyser` [19] is an extension of the ADE to carry out corner and Monte Carlo analysis. Measurements are used for performance extraction. It is possible to use calculator expressions [17], Ocean commands [18] and additional Skill scripts [20] for data extraction and post processing.

The support of Skill scripts is the key to realize robust data extraction and post processing. Because violations can exist or not exist the `pv` command can return values or nil respectively. Script 1 shows an example for a Skill script to

check existence of violations. If at least one violation exists one is returned otherwise zero is generated. An example for simple post processing is given in Script 2. As the calculator expressions and Ocean commands does not support violation counting the Skill script was developed.

```
if(pv("violation1" "value"
      ?result "tranViolations-violations")
    then 1
    else 0
)
```

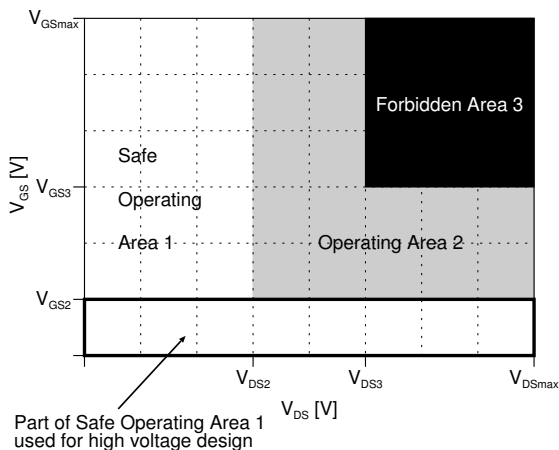
Script 1: Skill script to check violations from device checking.

```
prog(count())
count = 0
while(
  evalstring(
    sprintf(nil "pv(\"violation%L\" \"value\"
      ?result \"tranViolations-violations\")" (count+1)))
    count++
  );while
return(count)
)
```

Script 2: Skill script to count the violations.

The example of verification with device checking is a linear voltage regulator for providing a constant supply voltage of 3.3V together with 12mA supply current. Figure 5 shows a simplified schematic of this circuit, containing about 35 active and 7 passive devices. The design challenge is to handle a wide range of the external supply voltage of about 25V with the magnitude up to 36V and the demanded operating time of about 100.000 hours for industrial applications, hence a strong compliance of the SOA constraints. Most of our attention was directed towards the power consumption and voltage relation with maximum bulk current. For the last one we checked the gate voltage range near two-thirds of the related drain voltage.

As the X-FAB PDK of the 0.35 μ m technology [21] only provides SOA documents in form of tables, the ADE Device Checking capability was used to define SOA rules for design



Legend: Area 1: no degradation (Safe Operating)
 Area 2: life time (DC) is about Xh with Y% degradation
 Area 3: destroys the device
 Note: X and Y are parameters given by PDK

Figure 3: SOA diagram.

	geometrical	electrical
function		$V_{gs} - V_{th} > V_{invmin}$ $V_{ds} - (V_{gs} - V_{th}) > V_{satmin}$
robustness	$L > L_{min}$ $W > W_{min}$ $L * W > A_{min}$	
reliability		$V_{ds} < V_{dsmax}$ $V_{gs} < V_{gs2}$

Figure 4: Constraints for special devices (see Fig. 3).

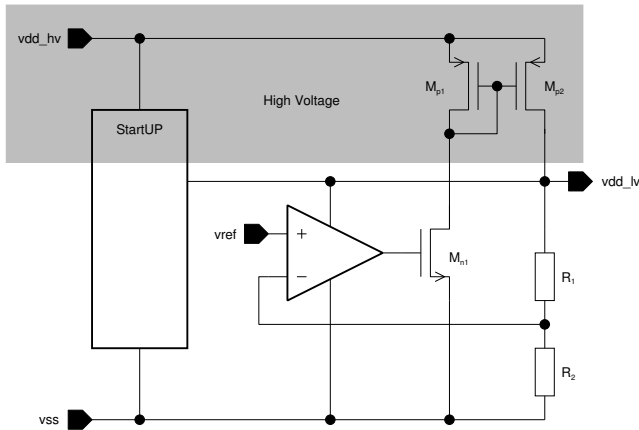


Figure 5: Voltage regulator to supply the circuit. Grey marked is the high voltage part.

and verification. About 10 checks were added for the device types nmos, pmos and high voltage pmos to check about 35 active devices. The setup effort is moderate because about three SOA rules per active device type are needed. Violations were investigated by ADE Violation Display tool during design phase. Finally, the verification of performance parameters and device violations was carried out over PVT corners by using zmdAnalyser. Here, the Scripts 1 and 2 were used to monitor the SOA rules via 2 parameters *deviceViolation* and *deviceViolationCount*. This method generates clear arranged results for the output protocol to recognize problems. Compared to this solution the definition of SOA rules by performance measuring would need about 350 single expressions.

Additional to the common SOA constraints like maximum voltages (e.g. V_{gs} , V_{ds} , V_{gb}) and current I_{ds} we detected power consumption of high voltage devices, where the gate-source voltage is close to $2/3$ of the drain-source voltage. The detected devices may be devices with a possible high substrate influence. As a result of this analysis three high voltage FETs fall into such behavior. A first action will be the placement of special guard structures around these three devices to slow down the generated substrate current.

4.2 Invoking Model Encapsulation

As we described earlier in the section, there is no tool which shares the same configuration for SOA purposes. Another way to integrate the SOA functionality into the analog design flow is to modify the device models used by the circuit simulator. Contrary to the method described in [14, 22] where the design engineer creates the equivalent circuit of the primitive device with the integrated SOA checker for the simulation an integration of the checker to the device model will be more comfortable. We could achieve the following benefits with this method:

- The same SOA violations are detected in all tools.
- The setup time is reduced to a simple change of the model path.

- A modification of the schematic view is not necessary.
- The SOA limits can be provided by the semiconductor foundry.

In addition to this requirement the precision of the modified device model must be guaranteed. It must be guaranteed to use the same characterisation for the special SOA model and the normal device model. To achieve this requirement the SOA functionality must be integrated similar to the functionality as shown in Figure 2.

To take a look into the current foundry processes, a PDK from X-FAB Semiconductor Foundries AG was analyzed [21, 23]. The foundry provides a set of models where the SOA functionality is integrated. The SOA investigation circuit is written in a description language and placed ad hoc to the device model. During simulation the SOA investigation circuit checks all SOA limits and writes the violations into a file in a human readable format.

```

procedure(checkSOAViolation(@optional (arg1 "exist"))
/*
Udo.Sobe@zmd.de

Description:
Simple Skill function to check existence of SOA
violations or count SOA violations from a LOG file.

Application:
checkViolation_va_xfab(s_mode)

Arguments:
s_mode=="exist":=>0:no SOA violations exist,
1:at least one SOA violations exist
s_mode=="count":=>count of violations

Examples:
                                checkSOAViolation()
                                checkSOAViolation("count")
*/
prog((count fileName filePort lineString)
;init
count=0
fileName=strcat(openResults() ".././soac.err")
;if file exist then open the file and check every line
if(isFile(fileName) then
    rexCompile("leaving SOA")
    filePort = infile(fileName)
    when (filePort
        while (gets(lineString filePort)
            if(rexExecute(lineString) then count++);if
        ) ;while
    ) ;when
    close(filePort)
);if
;return result
case(arg1
("exist" if(count!=0 then return(1) else return(0)))
("count" return(count))
);case
);prog
);procedure

```

Script 3: Simple Skill function to check existence of SOA violations or count SOA violations from a log file.

An analysis of the model files shows, that the limit for SOA a rectangular shape with the limits of V_{GS2} and V_{DSmax} has been chosen. This area is marked in Figure 3 as part of "Area1". Additional to the V_{GS} and V_{DS} limit, the limits of V_{DB} and V_{SB} are defined and evaluated.

For the time being the model encapsulation integrated inside the PDK is limited to the part of SOA1 focused on high

voltage design (cf. Fig. 3). Nevertheless there can be circuit demands to use the whole SAO1. Additionally it could be wise to take into account operating conditions with higher substrate current.

Additionally effort by a design center is necessary to include the above mentioned human readable log file into an automated verification analysis over PVT corners. As mentioned in section 3 the log file format differs between PDKs. With the help of a parser it is possible to evaluate the existence of SOA violations during verification e.g. by using the zmdAnalyser. Script 3 shows a very simple example, which also supports distribution of simulation jobs, too.

5 Summary and Outlook

In this paper we have shown how reliability constraints of current PDKs can be applied for verification over PVT corners. Model encapsulation is widely used in PDKs. In general the setup is conservative to ensure quality by FAB for a wide field of applications. Furthermore we discussed the ADE design checking technique which is useful to define additional SOA rules.

Today, both techniques are directed on the design phase where interactive tasks dominate. Challenges and solutions to automate SOA violation measuring for verification are presented. Here we have presented how PDKs which deliver model encapsulation can be improved by simple enhancements to support such an automation. Both techniques can be combined and complement the SOA based DfR method. Furthermore this feature is the basis for regression analysis of analog and mixed signal circuits.

In addition to the SOA checking we directed our intention on bulk currents to define devices with distinctly high substrate current, hence to be able to define measures against soft degradation of the circuit, too.

The verification with device checking for SOA constraints and substrate current by utilization the zmdAnalyser complements our SOA constraints matrix method for design optimization.

In this context, besides the substrate current and future prediction of long-term reliability via simulation, our main attention is focused on the influence of high temperatures on parameter change.

Acknowledgment

The project described in this report was financed by the "Bundesministerium für Bildung und Forschung" (BMBF), grant ID 01M3178B. The authors are responsible for the contents of this publication.

References

- [1] S. Abbas, "A device and process monitor," in *IEDM*, pp. 404–407, 1974.
- [2] K.-H. Rooch, "Floating short connected p+n+ pieces to guard a device from the influence through substrate current," invention discloser, American Microsystems Inc., 1999.
- [3] K.-H. Rooch and L. Steinbeck, "An investigation of latch-up guard structures for automotive HV-CMOS design," *DASS*, pp. 85–89, 2005.
- [4] K. Fricke and J. Würfl, "Schaltungstechniken für Hochtemperaturlektronik," *Fachbeilage MIKROPERIPHERIK*, vol. 5, no. 6, 1991.
- [5] A. M. Abo, *Design for reliability of Low-Voltage Switched-capacitor Circuits*. PhD thesis, University of California, Berkeley, 1999.
- [6] T. Jährling and K.-H. Rooch, "Temperaturfeste III-V-Halbleiter-Bauelemente," *Fachbeilage MIKROPERIPHERIK*, vol. 5, no. 6, 1991.
- [7] ZMD/ZFOUNDRY, *Reliability of Process C7*, 2006.
- [8] L. Goldgeisser, E. Christen, and Z. Deng, "Modeling safe operating area in hardware description languages," in *DAC '07: Proceedings of the 44th annual conference on Design automation*, (New York, NY, USA), pp. 377–382, ACM, 2007.
- [9] Austriamicrosystems, "Safe operating area check." <http://asic.austriamicrosystems.com/hitkit/hk370/soac/index.html>.
- [10] Cadence, *Virtuoso Analog Design Environment User Guide*, 2007.
- [11] www.zmd.de.
- [12] www.muneda.com.
- [13] U. Sobe, K.-H. Rooch, and M. Pronath, "Robust analog design for automotive applications by design centering," in *GI/GMMITG Fachtagung Zuverlässigkeit und Entwurf*, (München, Germany), pp. 47–54, 2007.
- [14] U. Sobe, K.-H. Rooch, A. Ripp, and M. Pronath, "Robust analog design for automotive applications by design centering with safe operating areas," in *IEEE ISQED*, (San Jose, California), pp. 848–854, 2008.
- [15] MunEDA and X-FAB, "X-FAB adds support for MunEDA WiCkeD tool family to statistical modeling offering to optimize circuit performance, statistical analysis, yield optimization." <http://www.muneda.com/News-44>, 2007.
- [16] V. Boos and H. Wald, "Technology setup for wicked in X-FAB CMOS and BICMOS process for automotive and sensor applications," tech. rep., MunEDA User Group Meeting (MUGM), 2007.
- [17] Cadence, *Waveform Calculator User Guide*, 2005.
- [18] Cadence, *Ocean Reference*, 2007.
- [19] U. Sobe and U. Henniger, "zmdAnalyser: Ein Design Tool von Analog/Mixed Signal Designern," *DASS*, pp. 29–33, 2005.
- [20] Cadence, *SKILL Language Reference*, 2007.
- [21] X-FAB, "XH035 - analog and high voltage options with non volatile memory." www.xfab.com, 2007.
- [22] K.-H. Rooch, U. Sobe, D. Mörtl, and M. Pronath, "Safe operating areas (SOA) constraining concepts for verification and optimization of analog and mixed signal circuits," in *DASS*, (Dresden), 2008.
- [23] www.xfab.com.