

Multi Domain Behavioral Models of Smart-Power ICs for Design Integration in Automotive Applications

Dieter Metzner, Jürgen Schäfer, Chihao Xu

Infineon Technologies AG
P.O. Box 800949 , D-81609 München, Germany

dieter.metzner@infineon.com, juergen.schaefer@infineon.com, chihao.xu@infineon.com

A. Summary:

A modeling methodology for behavioral models of Smart Power ICs is presented. These components typically consist of a few hundred to thousand transistors, which can be separated into digital (logic) and analog (electro-thermal) model equations. The preferred simulation tool in the automotive industry is presently SABER (MAST® language). The resulting models enable system level simulations of the mechatronic application in a few minutes CPU time, compared to hours with a transistor level simulation of the IC alone.

B. Introduction:

There are 3 incentives to develop of behavioral models for automotive applications Smart Power Ics' suppliers:

- **Simulations of the end application**, in most cases mechatronic systems, such as light modules, injector valves, ignition systems, window, wiper or door locks. Questions to the simulation here are typically: Characterization of mechanical, thermal and electrical stress on the components, cooling effort, losses, EMI, failure modes, an so on. I.e. application support for customers by component suppliers.

Such a simulation helps system designers to understand the dynamic behavior, limits, potentials and possible failures of the component in the system. It also allows the dimensioning of the components and system optimisation.

- **Concept verification** in an early phase of the chip development for the manufacturer. These include feasibility studies, worst case simulations, sensitivity on manufacturing processes as well as SOA considerations.

- **Application support** for the technical marketing of components, and component selection in the design-in phase. The supplier gets better feeling of the requirements and processes in complex applications.

A powerful language in the dialogue for specification and product definition is created.

C. Example:16-fold Low Side Switch

Fig.1 shows a typical application of the TLE6240GP, a 16-fold lowside switch designed for engine management applications:

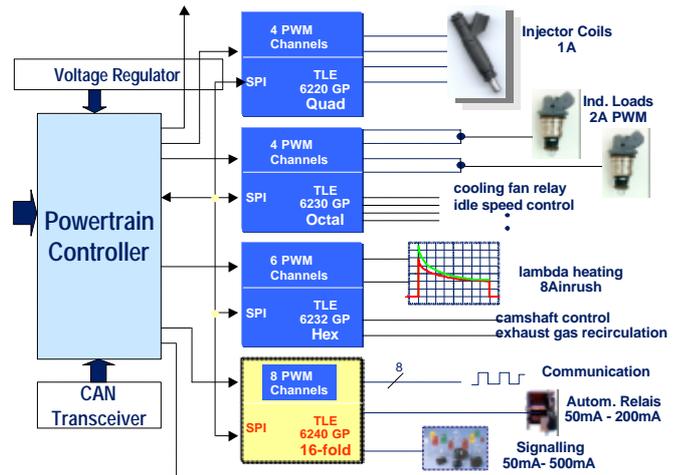


Figure 1: Powertrain application

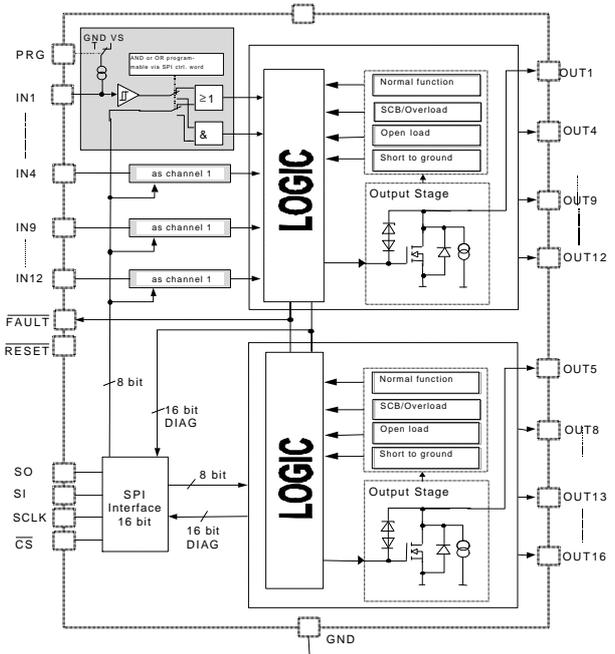


Figure 2: Functional structure of the TLE6240

As shown in Fig. 2 this IC is a typical example of an application for a mixed mode analog/digital design. The analog power output channels are controlled partly by parallel analog input pins and partly by a serial digital input (Serial Parallel Interface, SPI). Internal Proc-

essing and serial output of error status signals is accomplished by digital circuitry, while power supply-, gate drive-, and sensing functions are analog.

The features of the TLE6240 include:

- 16 low side channels 0.4Ω to 1.0Ω
- 8 PWM channels (parallel input) and 8 channels controlled only by SPI
- overtemperature protection
- shortcircuit protection (current limitation)
- overvoltage protection
- status feedback via SPI (openload, short-circuit, short to ground)

The process used is a Infineon Smart Power Technology based on junction isolation, rated to 60 Volts

Fig.3 illustrates the chip layout where the power output transistors (red stripes) can be well distinguished from the digital parts (center) of the circuit.

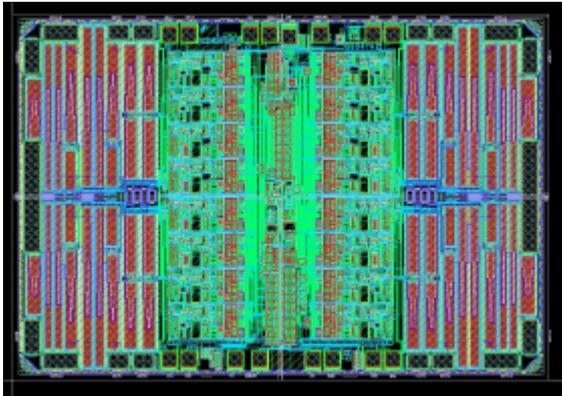


Figure 3 Physical Layout

D. Modeling methodology

Because the Model must cover many potential applications, a hybrid modeling concept is used with two different approaches:

- modules which description is based on the relevant physical equations:
 - I) exact representation of the switching behavior (Stress, EMI, timing analysis)
 - II) self heating effects (SOA, thermal design)
 - behavioral parts, such as
 - III) protective and diagnostic functions
 - IV) gate drive characteristics
 - V) input and status output characteristics
- have to be implemented.

At last, many "infrastructure" components ,e.g. supply concept and current mirrors can be eliminated in a behavioral model.

Since the switching behavior of a DMOS is strongly nonlinear, the accurate description of MOS Capacitances, is essential (Miller Capacitance) for realistic results like EMI analysis.

DC- characteristics as R_{dson} depend not only of the intrinsic MOS Channel, but might also be dominated by drift regions and metalization effects for power devices.

For SOA considerations, effects of impact ionization (avalanche breakdown) have to be included. In the semiconductor models, many of the parameters are coupled to the thermal system which consists of lumped elements.

Protective functions generally consist of feedback loops implemented by an analog circuit design. Here, the challenge is to reduce the complexity by behavioral descriptions containing controlled voltage or current sources.

While in a physical modeling approach all important dependencies on operating parameters as supply voltage or chip temperature are automatically included, the behavioral model parts have to be described by empirical equations. In this simplification process, a clear understanding of the circuit is crucial for a successful simulation (As many components as necessary, as few as possible).

To get a good tradeoff between accuracy and simulation time the circuit parts based on thresholds and logic functions must be represented by digital models, e.g. comparators, detection thresholds, processing of diagnosis information into protective actions, diagnosis output or interfaces such as the SPI (serial parallel interface).

Since many elements are used repeatedly in both domains (design and modeling), the structure of the design concept is converted to a modular modeling approach. Thus a behavioral model of a specific block can be used in a Top Down method for circuit and concept design.

This demonstrates that a top down design concept using behavioural models is much more efficient than with a classical transistor level simulation.

E. The TLE6240 Model

The method is illustrated with the behavioral model of the TLE6240 (multichannel lowside switch), which has been divided into the three named approaches:

i) The **analog model** parts cover a wide range of physical domains including the mechanical system, electromechanical parts (motors) and electrical circuit level (PCB) down to the silicon component (transistor level).

Since analog systems are represented by ordinary differential equations, the first and most important part of the modeling process is the creation of **compact or "lumped" models**.

In our case, two domains are relevant:

The **thermal behavior** is normally described by the heat diffusion equation, a partial D.E. used to determine the transient cooling conditions of the power device. Because 3D heat-flow is important (particularly in the case of a monolithic multichannel device), a lumped structure has to be built. To achieve the trade-off between accuracy and complexity (for parameter extraction) a transient finite element simulation is performed which yields the thermal step response at the heat generating cells (self heating) and at the neighbouring influenced elements (cross-coupling).

Then, a comprehensive lumped structure of the main heat paths is defined in a similar way

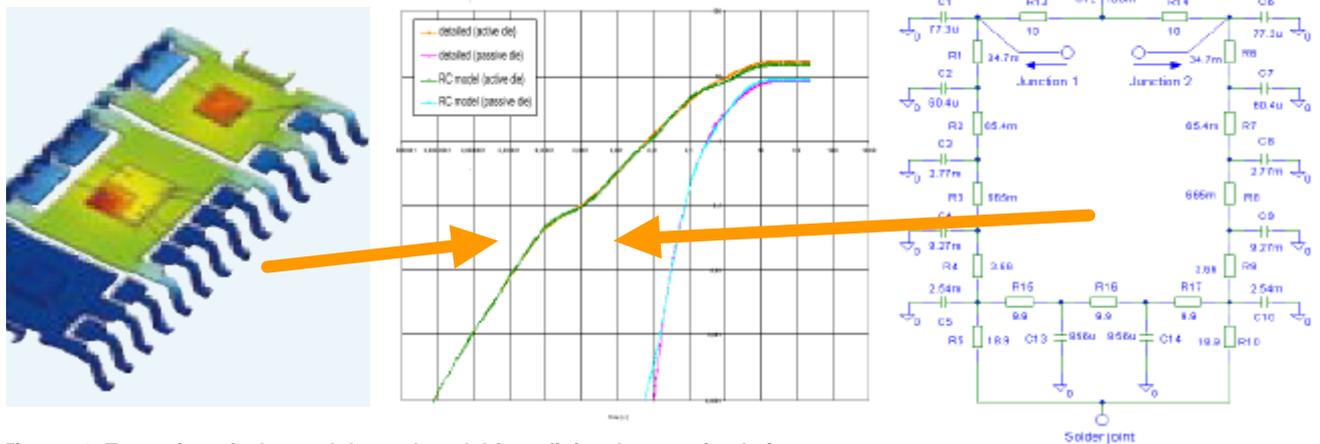


Figure .4: Extraction of a lumped thermal model from finite element simulations

ii) To describe the **Power MOS** output transistor models accurately (correct switching transients and the on-state) In this case we are also using information of finite element software like *Medici* or *Dessis* to optimize the lumped elements.

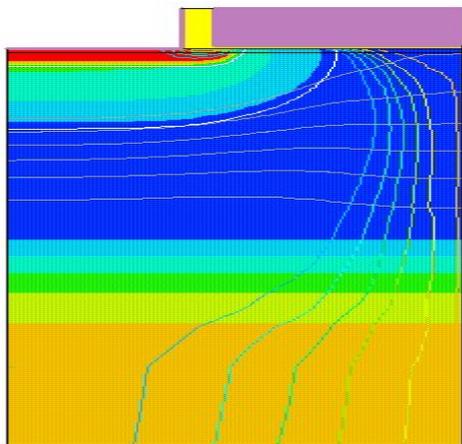


Figure 5a: Potential and current flow lines in a DMOS-Transistor cell

According to Fig. 5a, current paths and potential distributions in Silicon structures which result in nonlinear voltage controlled current

to electrical RC networks, involving vertical and lateral components, still leaving the parameter values for each element undefined.

With symbolic algebra software a closed form solution for the step responses can be calculated with arbitrary network parameters. At last, an optimizer with adequate constraints is used to find a reasonable combination of these parameters (R_i and C_i) which has the most similar transient response.

Fig. 4 is a simple example to illustrate the method. For a detailed description please refer to [noe].

sources. At this point all Parameters are derived from the basic equations of semiconductor physics.

A good example is the derivation of the Miller Capacitance, which is located at the depleted junction body-drift region (SCR in figure 5b):

The relevant equations are Poissons equation and the 2nd Maxwell equation:

$$\text{div}(\text{grad}(V)) = \frac{qel * Nd}{\epsilon}$$

$$j_{DIS} = \frac{d}{dt} \epsilon * \text{grad}(V) := C_j(V) * dV/dt$$

where Nd represents the doping concentration in the drift region, qel the elementary charge and ϵ the dielectric permmissivity of Silicon.

After simplification (one dimensional quasistatic approach) we obtain the well known expression for junction capacitance:

$$C_j(V) = \sqrt{\frac{qel * Nd}{2 * \epsilon}} (V - V_j)$$

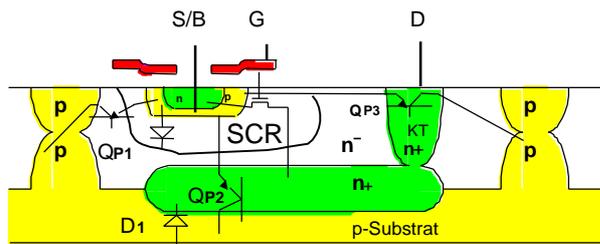


Figure 5b: cross section of a DMOS cell in self isolating Smart Power technology

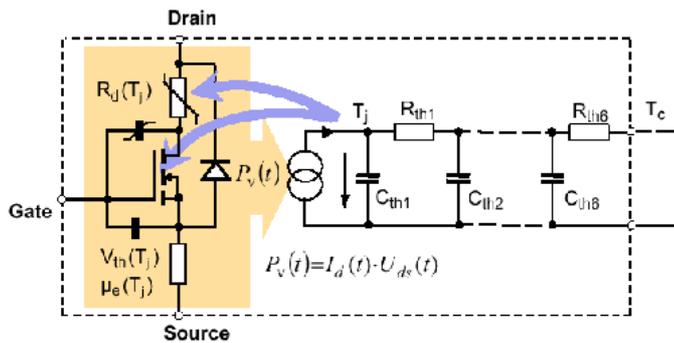


Figure 5c Compact model of the Power MOSTransistor (electrothermal), from [mmpn]

iii) the next level of abstraction describes the gate drive, including current limitation, active zehner clamping for switching off inductive

```

values{
    tflim=1+5e-4*(25-tc(tj))           # temperature drift of detection threshold
    iclim=(v(source,gndp)-vdslim*tflim) # transconductance to control Ugs of DMOS
    if (iclim>idlim) iclim=idlim        # iclim=discharge current of MOS gate
    if (iclim<0) iclim=0                # limiting of discharge current
}

when(threshold(v(source,gndp),vdslim*tflim,b_ocrd,a_ocrd)){ #voltage threshold at current shunt
    #temperature dependent (tflim)
    if (a_ocrd==1) schedule_event(time,ovlq,l4_1)           #set overcurrent flag at pos. X-ing
    schedule_next_time(time)                               #force analog timestep
}

when(threshold(v(source,gndp),vdslim*tflim-2m,b_ocrd,a_ocrd)){# negative X-ing with hysteresis
    if (a_ocrd==-1) schedule_event(time,ovlq,l4_0)# reset overcurrent flag
    schedule_next_time(time)                       # force analog timestep
}

```

Box 1: Example for behavioral MAST Implementation of a current limiting functionality

iv) The digital model

The TLE6240 consists of two digital circuits (c.f. figure 2). The first part of the digital circuit is the logic. This circuit encodes the error modes of each channel in form of two bits. The main part of the digital circuits is the SPI which is divided in 3 blocks: A serial input register, a serial output register and a coding and decoding circuit (figure 6). With the **cs** signal the system microcontroller selects the TLE6240. When this pin is in a logic low state data are transferred from the microcontroller to the TLE6240 and back. The signal **sclk** is the system clock of the TLE6240. On the falling

loads and fast/soft shut down features. These are implemented by means of: state dependent switches with nonlinear characteristics and hysteresis.

A classical example for a behavioral implementation of an analog circuit is given in Box 1 using a *current limiting function*:

The gate drive block in general consists of an array of zehner-diodes, several types of BJTs and/or MOSFETs which can be divided into sensing elements, comparators, clamping structures and charge/discharge paths for the gate of the output Power DMOS.

In the current limiting concept an additional gate discharge path is turned on, if the measured current exceeds a specified value.

The gate voltage is then limited by an analog feedback loop, thus shifting the output transistor to a lower curve $I_d=f(V_{ds})$.

The implementation is done with a simple transconductance to discharge the MOS-gate.

In addition, an interface to the digital world is needed for error processing: In the example of Fig. 6, the overcurrent flag **ovlq** is shown.

In MAST, this functionality is implemented by the **threshold** function:

edge of **sclk** the serial input register (shift register) accepts the signals on the pin **si** while the serial output shifts diagnosis information out of the shift register at the rising edge of the system clock. The serial input information will be shifted on the pin **si**, the most significant bit first. Diagnosis data bits are shifted out in serie on the pin **so**.

The signal reset is a logic low-active signal. If this signal is logic low then the SPI shift registers are cleared and all outputs are switched off.

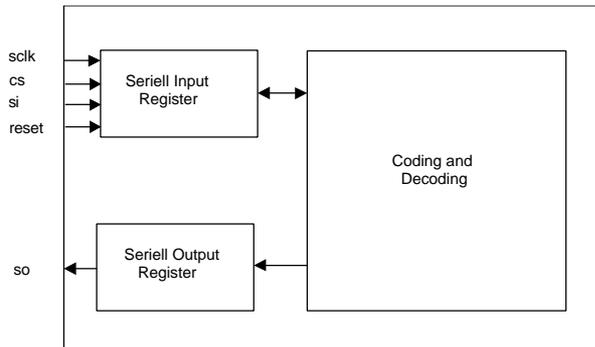


Figure 6: Functional Blocks of the SPI

The circuit of the Serial Peripheral Interface includes more than two thousand transistor functions. The behavioral model of the SPI is subdivided in the same functions blocks as the SPI transistor circuit.

For the implementation of the SPI the HDL language MAST model has been chosen. It consists only of digital language elements (Box 2). Therefore the digital solver (event controlled) of the network simulator will be used. The model is to be very fast; and to show a very good numerical stability.

Serial input are entered into the input shift register when a logic low at the **cs** signal (chip select), a falling edge at the system clock and a logic high at the reset signal (Box 2) occur. The when section becomes active if one of this logic signals is changed (event-triggered). So far only the principal function of the input shift register has been considered. In order to take into account the accurate timing behavior of the SPI interface (Fig. 7), parasitic effects of the technology dependent devices must be included (e.g. MOS Capacitances).

```

-----
when((event_on(csq,csq_oldd))|(event_on(clk,clk_oldd))|
(event_on(resq))|(event_on(num))){

if((csq==l4_0)&(clk==l4_0)&(clk_oldd==l4_1)&(resq==l4_1)
&(t_clokh_t_start>=tlead)&(t_clokh_t_clokh>=tsckh)
&(t_clokh-t_clokl_oldd>=tsckl)&(num==1)){
handle0=schedule_event(time+th,d0_i,sdii)
handle1=schedule_event(time+th,d1_i,d0_i)
handle2=schedule_event(time+th,d2_i,d1_i)
handle3=schedule_event(time+th,d3_i,d2_i)
handle4=schedule_event(time+th,d4_i,d3_i)
handle5=schedule_event(time+th,d5_i,d4_i)
handle6=schedule_event(time+th,d6_i,d5_i)
handle7=schedule_event(time+th,d7_i,d6_i)
handle8=schedule_event(time+th,d8_i,d7_i)
handle9=schedule_event(time+th,d9_i,d8_i)
handle10=schedule_event(time+th,d10_i,d9_i)
handle11=schedule_event(time+th,d11_i,d10_i)
handle12=schedule_event(time+th,d12_i,d11_i)
handle13=schedule_event(time+th,d13_i,d12_i)
handle14=schedule_event(time+th,d14_i,d13_i)
handle15=schedule_event(time+th,d15_i,d14_i)
handle16=schedule_event(time+th,d16_i,d15_i)
}
}
-----

```

Box 2: Digital Modeling with MAST

In the if-assignment (Box 2) the timing behavior (enable lead time of falling edge **cs** to rising edge **sclk** is 200ns) of the SPI model is noticed by the expression as $t_clokh_t_start \geq tlead$. These are discontinuous Mast® variables which are set equal to the global simulator variable "time" at specified events. E.g. the variable t_start is the time at which the **cs** signal has changed from high to low.

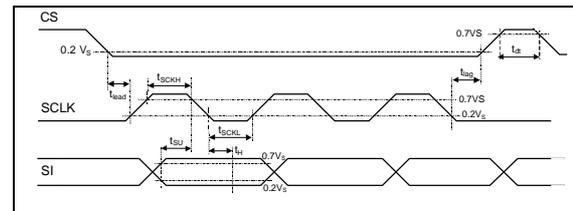


Figure 7: Timing Specification of SPI

The detection of this time point is implemented in a "when"-section (Box 3). Whenever an event on the **cs** signal occurs, the "when" section will be activated.

```

-----
when(event_on(csq,csq_oldd)){
if((csq==l4_0) & (csq_oldd==l4_1)){
t_start=time
num=num+1
}
else if((csq==l4_1) & (csq_oldd==l4_0)){
t_end=time
}
}
-----

```

Box 3: Implementation of timing behavior

Figure 8 shows a simulation of one SPI cycle. The top signal is the system clock, followed by chip-select and serial input.

The following signals **d0_i - d15_i** are the internal signals of the input shift register. As shown in the example the input shift register does not accept data before the second system clock pulse.

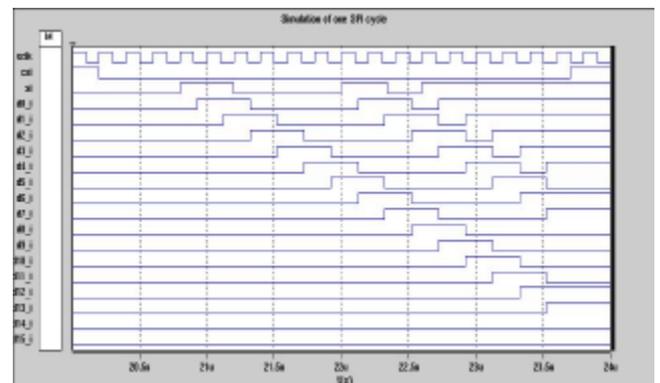


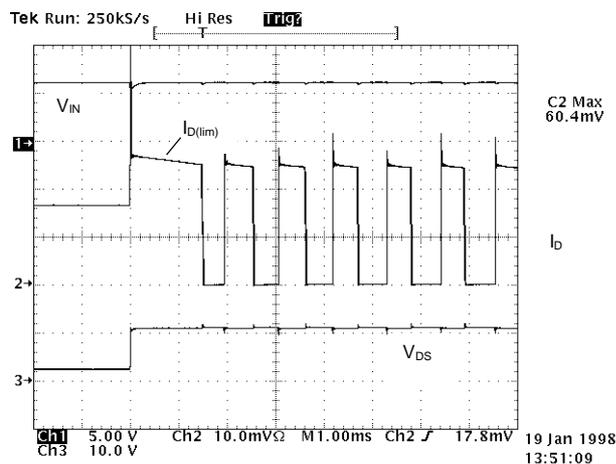
Figure 8: Read-in process from serial input

F. Model Validation:

In general, a model specification should reflect typical datasheet values. Accordingly, all relevant datasheet features have to be validated. Fine-tuning of parameters is possible for adaptation of the model to a particular device or in order to give a tighter correlation to characterization results.

In Fig 9, the validation of the current limiting function and the thermal behavior is demonstrated by a "short circuit2" operation mode:

At $t=2\text{msec}$ one channel is turned on with a load current of 0.8A which is slightly below the



current limitation threshold (datasheet: $1.0\text{A} \dots 2.0\text{A}$). At $t=4\text{msec}$ a short circuit occurs across the load terminals. After a short overshoot, the switch current is limited to 1.3A and the voltage at the output rises to the battery voltage level. The instantaneous power density at the output DMOS transistor is approximately $400\text{W}/\text{mm}^2$ which heats the junction beyond the overtemperature protection level. So called thermal toggling is initiated, the protection function turns the DMOS off and on repetitively depending on the detected temperature level.

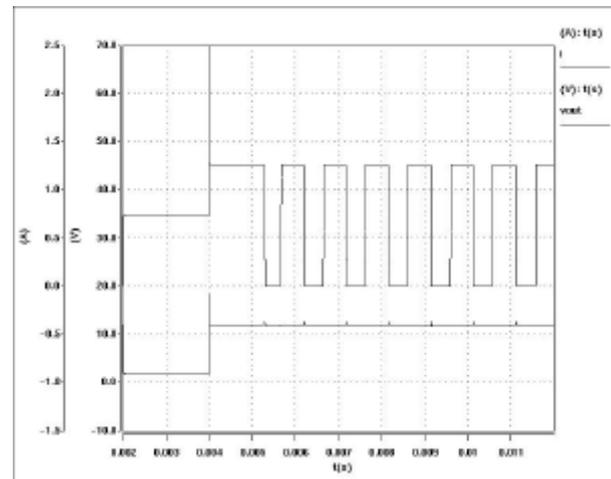


Figure 9: Shortcircuit at load during on-state, left: measurement, right: simulation

G. Conclusion:

The described model approach makes an inclusion into a System level simulation possible. Compared to a transistor level simulation with a primitive environment (like it is used in IC design) the simulation time reduces by a factor of 500 while preserving all features relevant for the application.

The mid-term modeling strategy is to provide models for all new automotive&industrial products before engineering samples are available. Thus, the circuit designer can take advantage of a significant reduction in development time. Actual trends indicate that in the future VHDL-AMS will be the preferred language in the Automotive sector.

Potential advantages are the interchangeability between different simulators and more efficient model implementation (in comparison to e.g. box 2).

The models are available for a free download on the companies simulation homepage:

<http://www.infineon.com/simulate>

Advantages of the model exchange are evident on side of the component supplier as well as for customers:

- the component supplier gets better insight into the requirements and processes in complex applications with regard to the electrical system design
- the system designer realizes and understands the dynamic behavior, limits, potentials and possible failures of the component in the system
- a powerful language in the dialogue for specification and product definition is created.

References:

[noe]:

Noebauer, G., "Creating Compact Models Using Standard Spreadsheet Software", *Proc. of 17th IEEE Semiconductor Thermal Measurement & Management Symp.*, 2001

[mmpn]:

http://www.infineon.com/cmc_upload/migrated_files/document_files/Application_Notes/mmpn_eng.pdf