



Automotive Power Semiconductors

Application Note

Behavioural Model for SABER Simulations (MAST)

Multichannel HITFET TLE6240GP

by D.Metzner, J. Schäfer, M. Glavanovics

Support:simulate@infineon.com

Contents:

A. Introduction	2
B Modular Modelling methology, Interface declarations	2
C The digital model	5
D. Model Parameters and Features:	10
E. Example Simulation circuits	11
F. Validation Results, Range of Validity	14
G. Installation hints, Simulator settings	21



A. Introduction

A mixed signal behavioural model of the multichannel HITFET Device TLE 6240GP was implemented into the simulation package SABER. The level of abstraction is suited to support the design of electronic systems.

Dynamic and static characteristics are implemented as well as selfheating effects, protection and feedback features.

The Serial Peripheral Interface (SPI) was implemented by a digital model which interacts with the analog part by the MAST modeling language.

With the aid of these models, expensive breadboard experiments can be reduced and critical operation modes can be identified.

For a detailed description of the Product please refer to

http://www.infineon.com/cmc_upload/0/000/013/552/tle6240gp.pdf

http://www.infineon.com/cmc_upload/0/000/010/276/ANPS034e_inf.pdf

http://www.infineon.com/cmc_upload/migrated_files/document_files/Application_Notes/anps043e.pdf

B Modular Modelling methology, Interface declarations

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

- modules where the 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

were implemented.

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

Output DMOS Transistor

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 been 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. To get a good tradeoff between accuracy and simulation time the circuit parts based on thresholds and logic functions were 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.

1) 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 diff. eq. used to determine the transient cooling conditions of the power device. Because 3D heatflow is important (particularly in the case of a monolithic multichannel device), a lumped structure has to be built. To achieve the tradeoff 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 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.

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.

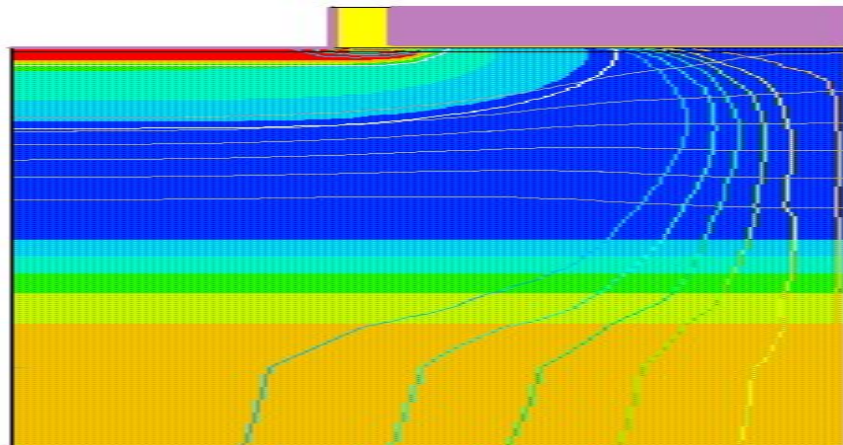


Figure 1: Potential and current flow lines in a DMOS-Transistor cell

According to Fig. 1, current paths and potential distributions in Silicon structures result in nonlinear voltage controlled current sources. At this point all Parameters are derived from the basic equations of semiconductor physics.

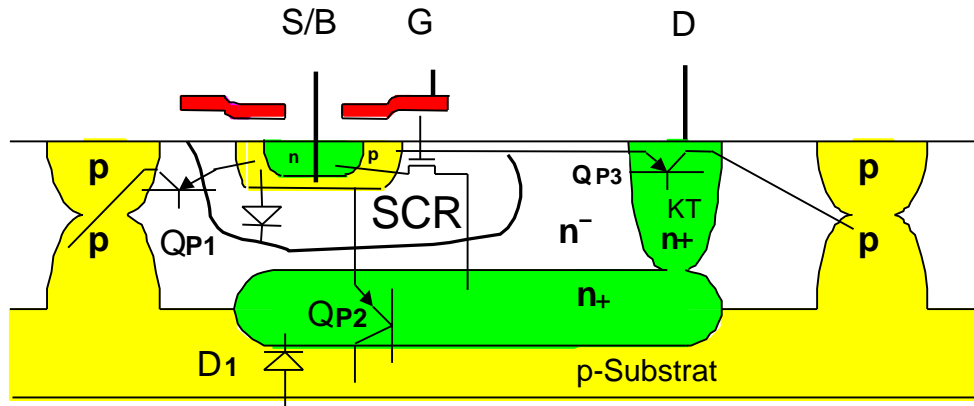


Figure 2: cross section of a DMOS cell in self isolating Smart Power technology

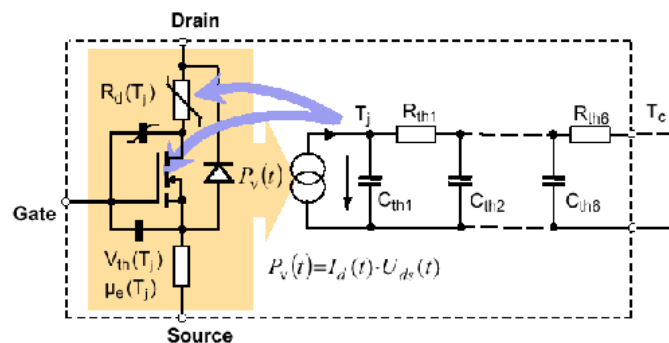


Figure 3 Compact model of the Power MOSTransistor (electrothermal), from [mmpn]

The next level of abstraction is the behavioural simplification of the gate drive, including current limitation, active zener clamping for switching off inductive loads and fast/soft shut down features. These are implemented by means of: state dependent switches with nonlinear characteristics and hysteresis.

An example for a behavioral implementation of an analog circuit is the *current limiting function*:

The gate drive block in general consists of an array of zener-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 Box1, the overcurrent flag $ov1q$ is shown.

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

```

-----
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_ocd,a_ocd)){ #voltage threshold at current shunt
    #temperature dependent (tflim)
    if (a_ocd==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_ocd,a_ocd)){# negative X-ing with hysteresis
    if (a_ocd==-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

C The digital model

The TLE6240 consists of two digital circuits (c.f. figure 4). 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 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.

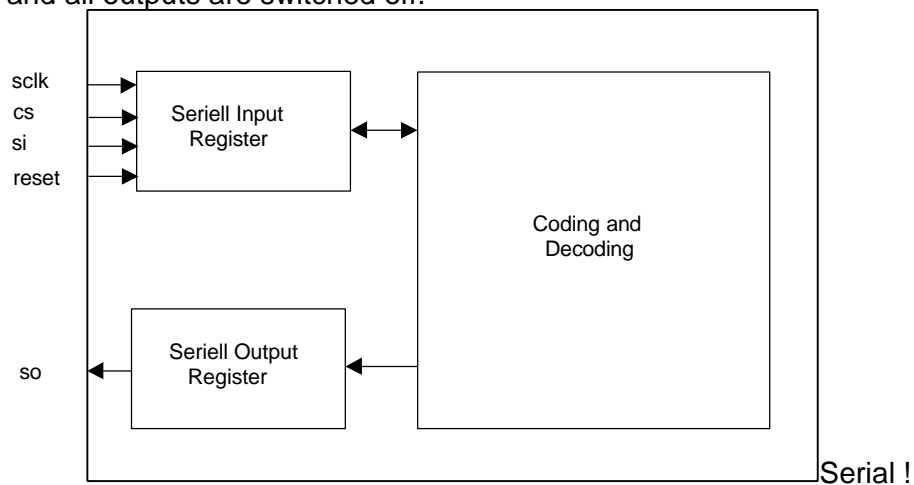


Figure 4: Functional Blocks of the SPI

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 these 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. 5), parasitic effects of the technology dependent devices must be included (e.g. MOS Capacitances).

```

when((event_on(csq,csq_old))|(event_on(clk,clk_old))|(event_on(resq))|(event_on(num))) {

    if((csq==14_0)&(clk==14_0)&(clk_old==14_1)&(resq==14_1)&(t_clokh_start>=tlead)&
        (t_clokl_start_clokh>=tsckh)&(t_clokh-t_clokl_old>=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.

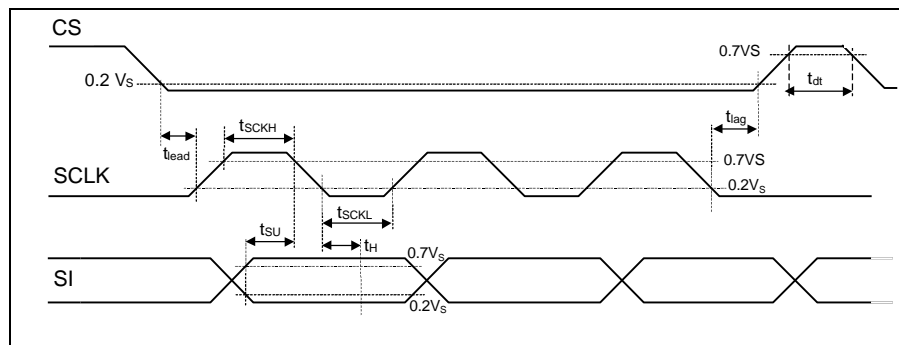


Figure. 5: Timing Specification of SPI

The detection of this time point (*point in time / event...*) 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_old)){
  if((csq==14_0) & (csq_old==14_1)){
    t_start=time
    num=num+1
  }
  else if((csq==14_1) & (csq_old==14_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 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.

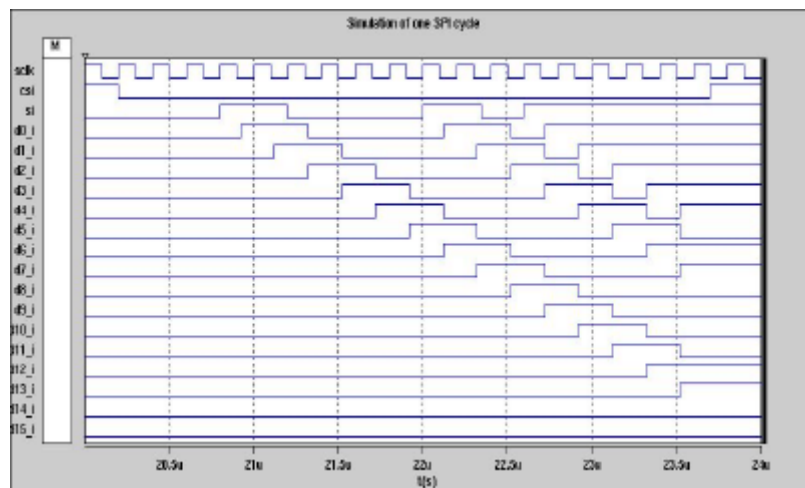


Figure 6: Read-in process from serial input

Interface declaration

The connection points are partly of physical nature (electrical and thermal pins), partly digital interfaces (serial out, serial in, clock, chip-select)

This separation is very desirable with respect to a CPU time- efficient execution of the model equations. With this strategy the analog model has to be evaluated only if the digital interface passes a switching command down to the output channel.

```

template tle6240 clk csi sdi sdo vdd gnd prg reset fault in1 out1 in2 out2 in3 out3 in4 out4 out5 out6
out7 out8 in9 out9 in10 out10 in11 out11 in12 out12 out13 out14 out15 out16 tcase

```

supply, ground, programming-input, reset, fault (open drain)

electrical vdd, gnd, prg, reset, fault

power output pins

electrical out1, out2, out3, out4, out5, out6, out7, out8, out9, out10, out11, out12,
out13,out14, out15, out16

parallel input pins 8 channels analog connection point electrical in1, in2, in3, in4, in9, in10, in11, in12

digital inputs for Serial Peripheral InterFace (SPI) state logic_4 csi, clk, sdi, sdo

leadframe thermal node; one common case temperature for connection with heatsink thermal_c tcase

- **General remarks:**
- Electrical Pins should never be left disconnected in order to avoid numerical instabilities
- The same applies to the thermal pin **tcase**, which is the only thermal interface from the internal thermal model to the heatsink.
- For thermal investigations, descent in the hierarchy into the TLE6240 model and display the thermal pins **tji** ($i=1\dots 16$) representing the junction temperatures of the output dmos transistors.
- Don't connect current sources directly to electrical pins, like in R_{dson}-„measurements“.

Functional structure of the TLE6240 with **digital** and **analog** blocks

Physical design

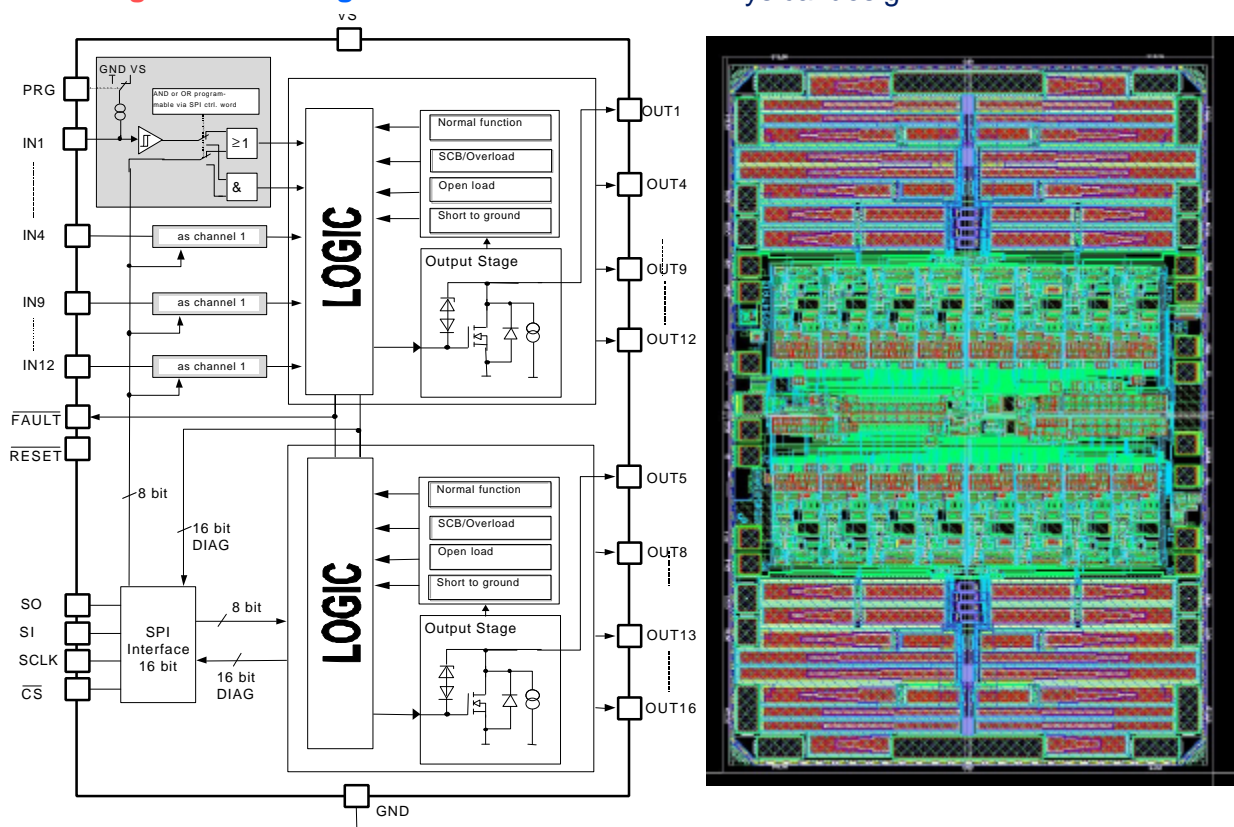


Fig 9
Figure 1 shows the functional concept of one channel which is adopted in the behavioural model as well: The top level model of each channel accordingly consists of a netlist of basic structural elements:



or-ed 16 faults, analog open drain output (=fault)
any fault can cause fault signal

or16.out gnd fault fault1 fault2 fault3 fault4 fault5 fault6 fault7 fault8 fault9 fault10 fault11 fault12 fault13
fault14 fault15 fault16

A/D conversion of parallel inputs with PRG-dependent interpretation of floating input
#

pib.1 resq prgi in1 vdd gnd p_in1
pib.2 resq prgi in2 vdd gnd p_in2
pib.3 resq prgi in3 vdd gnd p_in3
pib.4 resq prgi in4 vdd gnd p_in4
pib.9 resq prgi in9 vdd gnd p_in9
pib.10 resq prgi in10 vdd gnd p_in10
pib.11 resq prgi in11 vdd gnd p_in11
pib.12 resq prgi in12 vdd gnd p_in12

SPI subcircuit call

spi_6240.1 clk csi resq sdi uol uoh dl1 dl2 dl3 dl4 dl5 dl6 dl7 dl8 dh9 dh10 dh11 dh12 dh13 dh14 dh15
dh16 db1l db1h db2l db2h db3l db3h db4l db4h db5l db5h db6l db6h db7l db7h db8l db8h db9l db9h
db10l db10h db11l db11h db12l db12h db13l db13h db14l db14h db15l db15h db16l db16h fault1 fault2
fault3 fault4 fault5 fault6 fault7 fault8 fault9 fault10 fault11 fault12 fault13 fault14 fault15 fault16 p_in1
p_in2 p_in3 p_in4 p_in9 p_in10 p_in11 p_in12 sdo

A/D conversion of prgi - input, inverted
a_d.prg vdd prg gnd prgi

reset buffer with reset/resq A/D Conversion Outp: resh resq resqq
resbuf.1 vdd gnd reset plb off resh resqq resq # o.k: reset(eing), resq(outp), resh(outp), off(outp)

Supply-UV detection with latch
supply_spt.1 vdd gnd resq plb nlb nhb off resh

subcircuits for channels 1 to 16
no., RDSon, type, DMOS block

K1, 1 Ohm, par, LU/3
Iss01.1 p_in1 dl1 prgi uol resq rfsdq off reshn1 fault1 db1l db1h out1 gnd vdd tj1 ts1 tcu
K2, 1 Ohm, par, LU/4
Iss01.2 p_in2 dl2 prgi uol resq rfsdq off reshn2 fault2 db2l db2h out2 gnd vdd tj2 ts2 tcu
K3, 1 Ohm, par, RU/4
Iss01.3 p_in3 dl3 prgi uol resq rfsdq off reshn3 fault3 db3l db3h out3 gnd vdd tj3 ts3 tcu
K4, 1 Ohm, par, RU/3
Iss01.4 p_in4 dl4 prgi uol resq rfsdq off reshn4 fault4 db4l db4h out4 gnd vdd tj4 ts4 tcu
K5, 1 Ohm, ser, RO/3
Iss01.5 imh dl5 imh imh resq rfsdq off dumres fault5 db5l db5h out5 gnd vdd tj5 ts5 tcu
K6, 1 Ohm, ser, RO/4
Iss01.6 imh dl6 imh imh resq rfsdq off dumres fault6 db6l db6h out6 gnd vdd tj6 ts6 tcu
K7, 1 Ohm, ser, LO/4
Iss01.7 imh dl7 imh imh resq rfsdq off dumres fault7 db7l db7h out7 gnd vdd tj7 ts7 tcu
K8, 1 Ohm, ser, LO/3
Iss01.8 imh dl8 imh imh resq rfsdq off dumres fault8 db8l db8h out8 gnd vdd tj8 ts8 tcu
#-----

K9, 0.3 Ohm, par, LU/1
Iss030.9 p_in9 dh9 prgi uol resq rfsdq off reshn9 fault9 db9l db9h out9 gnd vdd tj9 ts9 tcu



```
### K12, 0.3 Ohm, par, RU/1  
Iss030.12 p_in12 dh12 prgi uol resq rfshq off reshn12 fault12 db12l db12h out12 gnd vdd tj12 ts12 tcu  
### K13, 0.3 Ohm, ser, RO/1  
Iss030.13 imh dh13 imh imh resq rfshq off dumres fault13 db13l db13h out13 gnd vdd tj13 ts13 tcu  
### K16, 0.3 Ohm, ser, LO/1  
Iss030.16 imh dh16 imh imh resq rfshq off dumres fault16 db16l db16h out16 gnd vdd tj16 ts16 tcu  
#-----
```

```
### K10, 0.35 Ohm, par, LO/1  
Iss035.10 p_in10 dh10 prgi uol resq rfshq off reshn10 fault10 db10l db10h out10 gnd vdd tj10 ts10 tcu  
### K11, 0.35 Ohm, par, LU/2  
Iss035.11 p_in11 dh11 prgi uol resq rfshq off reshn11 fault11 db11l db11h out11 gnd vdd tj11 ts11 tcu  
### K14, 0.35 Ohm, ser, RO/2  
Iss035.14 imh dh14 imh imh resq rfshq off dumres fault14 db14l db14h out14 gnd vdd tj14 ts14 tcu  
### K15, 0.35 Ohm, ser, LO/2  
Iss035.15 imh dh15 imh imh resq rfshq off dumres fault15 db15l db15h out15 gnd vdd tj15 ts15 tcu  
#-----
```

```
## thermal model: 4 geometric units (blocks) each with 3 lateral paths  
## tj (junction), ts(sensor)
```

```
therm4.lo tj16 ts16 tj15 ts15 tj8 ts8 tj7 ts7 tcu ## thermal DMOS-Block LO (links oben)  
therm4.ro tj13 ts13 tj14 ts14 tj5 ts5 tj6 ts6 tcu ## thermal DMOS-Block RO (rechts oben)  
therm4.ru tj12 ts12 tj11 ts11 tj4 ts4 tj3 ts3 tcu ## thermal DMOS-Block RU (rechts unten)  
therm4.lu tj9 ts9 tj10 ts10 tj1 ts1 tj2 ts2 tcu ## thermal DMOS-Block LU (links unten)
```

```
## simple cu-block (leadframe)  
scth.cu tcu 0 = 0.4  
srth.cu tcu tcase =20m  
}
```

D. Model Parameters and Features:

In general, the setup of parameters is done according to typical data sheet values. In some cases, where only maximum (conservative) values are given in the data sheet, the model represents a realistic, typical value.

Thermal Protection:

Additional model-components for the temperature sensor of each channel, have been integrated into the active DMOS area. Those are needed to account for the dynamic characteristics of the sensor with a time constant of approximately 200 μ s.

For thermal protection the sensor temperature is monitored and overtemperature turn-off is initiated. Retriggering via hysteresis is implemented.

Overvoltage Protection & Shutdown

The protection threshold voltage is decreasing with temperature

The shutdown threshold voltage (with hysteresis) is increasing with temperature.

Undervoltage Protection & Shutdown

The protection threshold voltage is modeled with hysteresis decreasing with temperature

Short Circuit Protection / Overcurrent protection

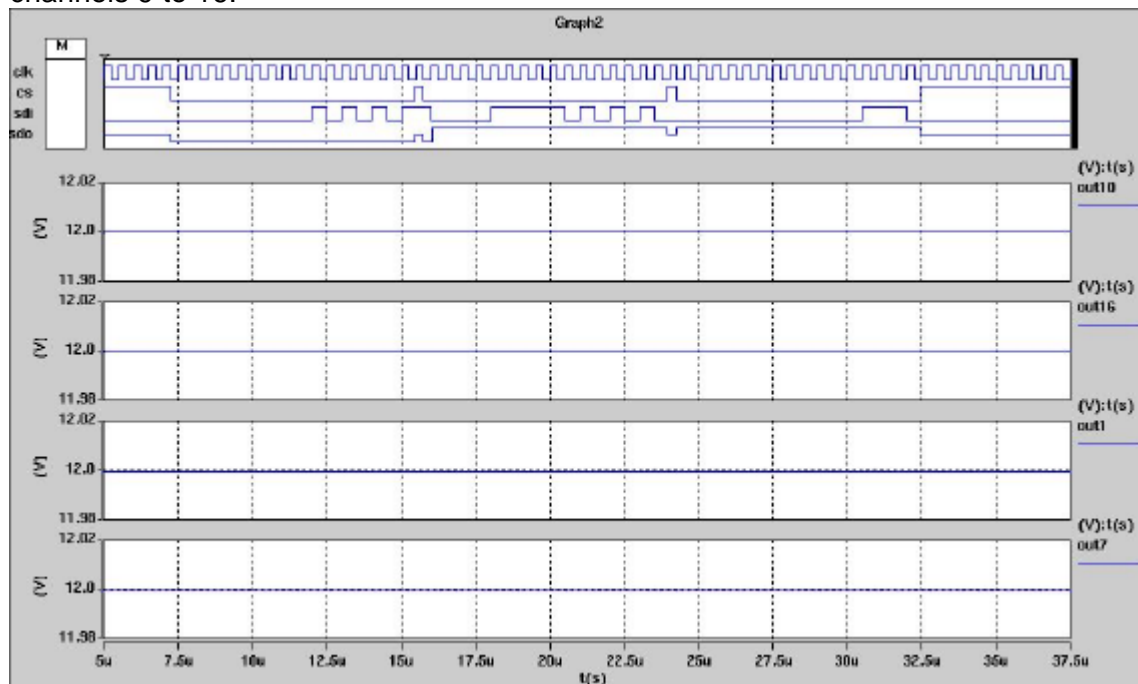
Dependencies on Operating Voltage and temperature effects are dominated by DMOS Transistor, gate charge and discharge circuit and dynamic characteristics of temperature sensor.

E. Example Simulation circuits

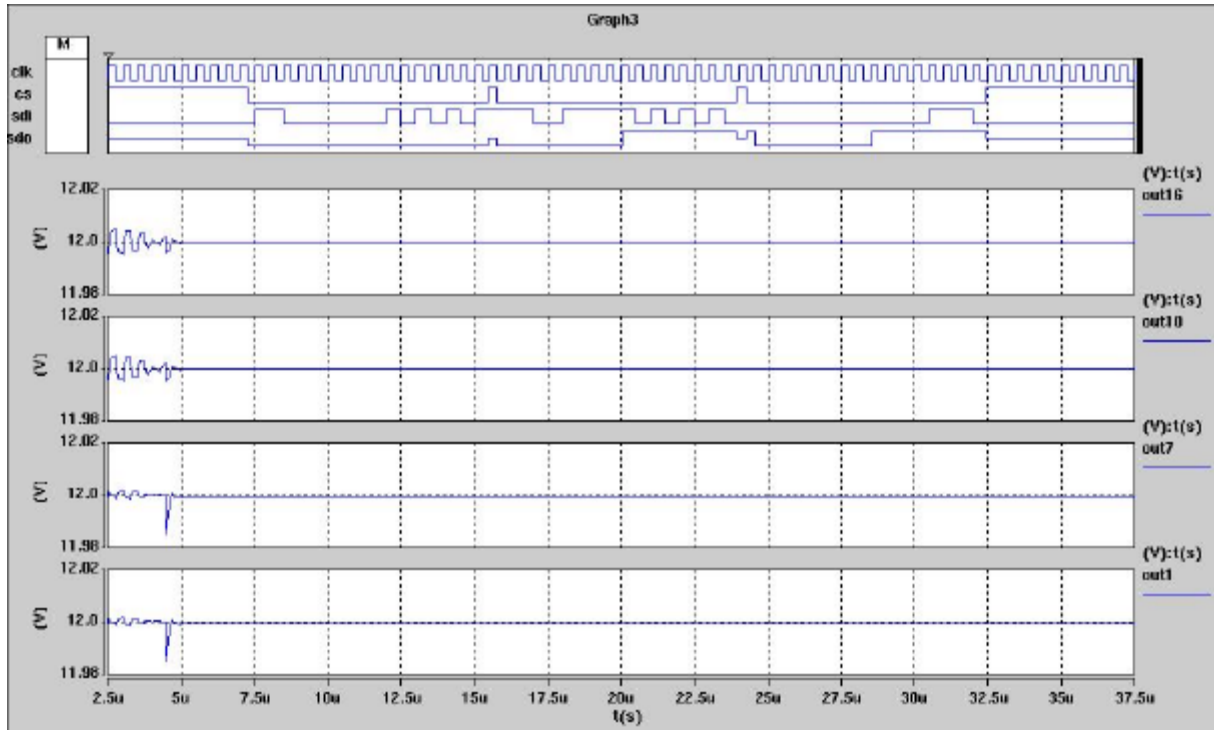
In order to get a quick insight into the functionalities of the models, simulation test patterns have been included into test bench files “<filename>.sin”: These examples of input patterns relate to control words described in the datasheet of the TLE6240:

Ten specific control words are recognised, having the following functions:			<filename>
No.	SI Control and Data Byte	Function	
1	LLLL LLLL XXXX XXXX	'Full Diagnosis' (two bits per channel) performed for channels 1 to 8. No change to output states.	<i>full_diagnosis</i>
2	HHLL LLLL XXXX XXXX	State of the eight parallel inputs and '1-bit Diagnosis' for channel 1 to 8 is provided	<i>1bit_diagnosis</i>
3	HLHL LLLL XXXX XXXX	Echo-function of SPI; SI direct connected to SO	<i>echo_on</i>
4	LLHH LLLL DDDDDDDD	IN1...4 and serial data bits 'OR'ed. 'Full Diagnosis' performed for channels 1 to 8.	<i>ored</i>
5	HHHH LLLL DDDDDDDD	IN1...4 and serial data bits 'AND'ed. 'Full Diagnosis' performed for channels 1 to 8.	<i>anded</i>
6	LLLL HHHH XXXX XXXX	'Full Diagnosis' (two bits per channel) performed for channels 9 to 16. No change to output states.	
7	HHLL HHHH XXXX XXXX	State of the eight parallel inputs and '1-bit Diagnosis' for channel 9 to 16 is provided.	
8	HLHL HHHH XXXX XXXX	Echo-function of SPI; SI direct connected to SO	
9	LLHH HHHH DDDDDDDD	IN9...12 and serial data bits 'OR'ed. 'Full Diagnosis' performed for channels 9 to 16.	
10	HHHH HHHH DDDDDDDD	IN9...12 and serial data bits 'AND'ed. 'Full Diagnosis' performed for channels 9 to 16.	

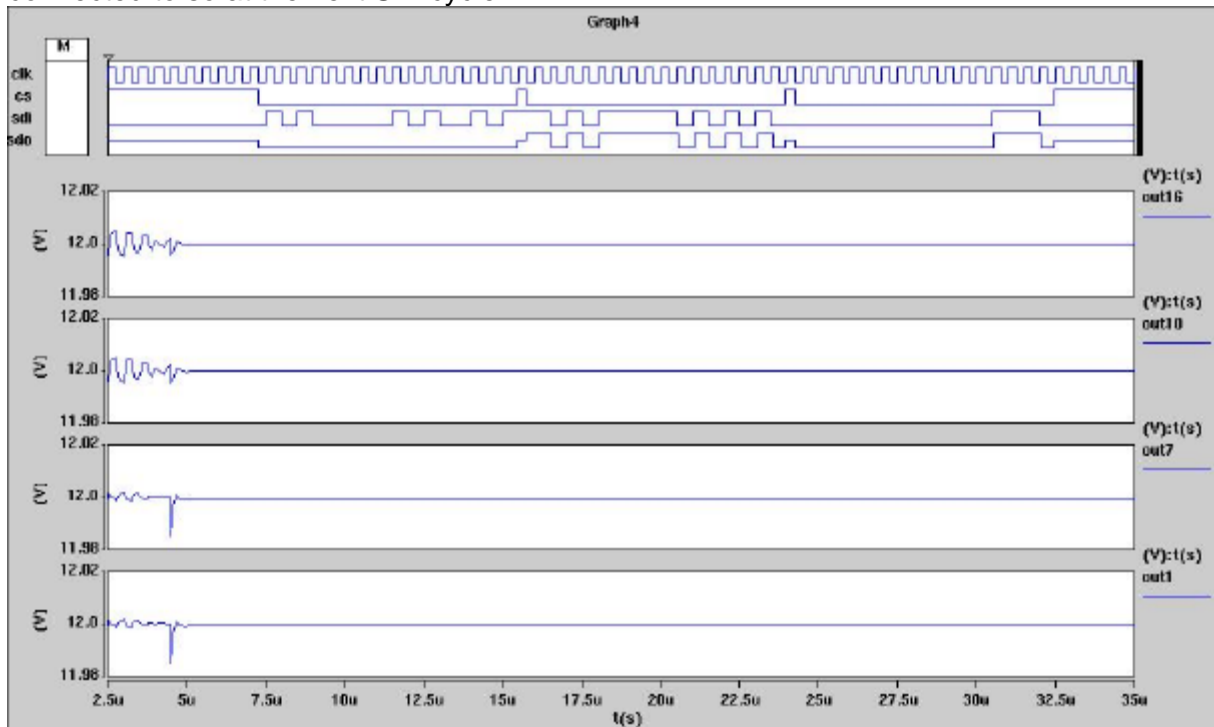
Example1: full diagnosis (two status bits per channel). The channels are not changed. The 1st SPI cycle is performed for channels 1 to 8 and the 2nd SPI cycle is performed for channels 9 to 16.



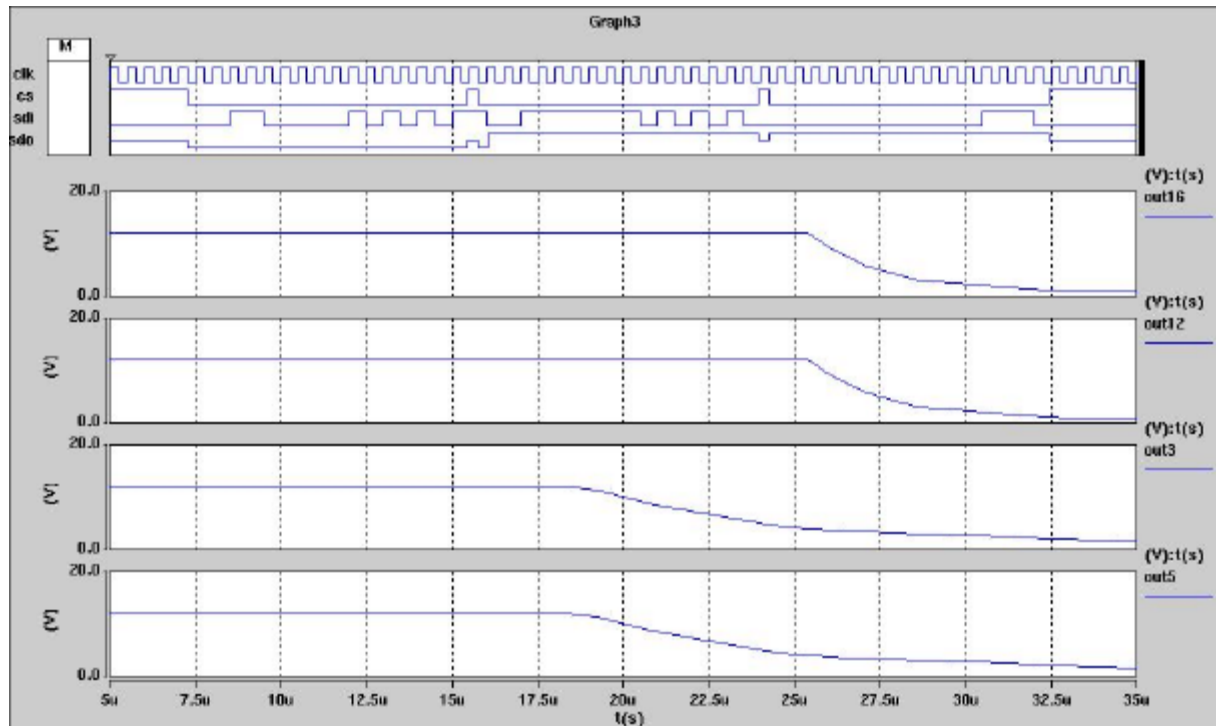
Example2 : This is the 1-bit diagnosis mode. This mode provides the state of the eight parallel inputs and the 1-bit diagnosis for the channel 1 to 8 (SPI cycle 1) acv. For the channel 9 to 16 (SPI cycle 2)



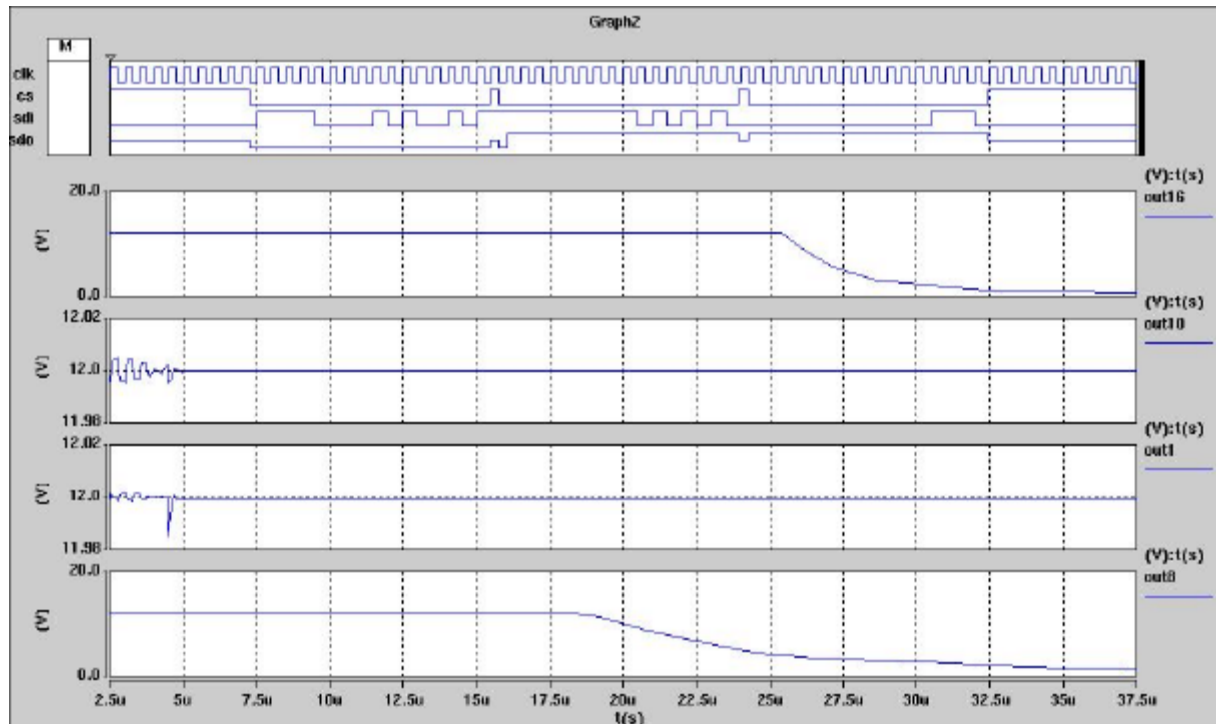
Example 3: this example shows the echo-function of the SPI. At this mode the si signal is connected to so at the next SPI cycle.



Example 4: OR operation two bit diagnosis. Each of the parallel input signals in1 to in4 (SPI cycle1) acv. In9 to in12 (SPI cycle 2) are ORed with the corresponding si data bits.



Example 5: AND operation and two bit diagnosis. In this mode the parallel inputs in1 to in4 and the serial inputs (SPI inputs) are connected with the logic AND operation.



F. Validation Results, Range of Validity

In general, the concept of an electronic datasheet was introduced into the simulation model. Although this does not imply that simulation results guarantee device properties, most of the data sheet values are represented in the model. Validation should therefore be related to typical data sheet values rather than to application measurements of a particular device.

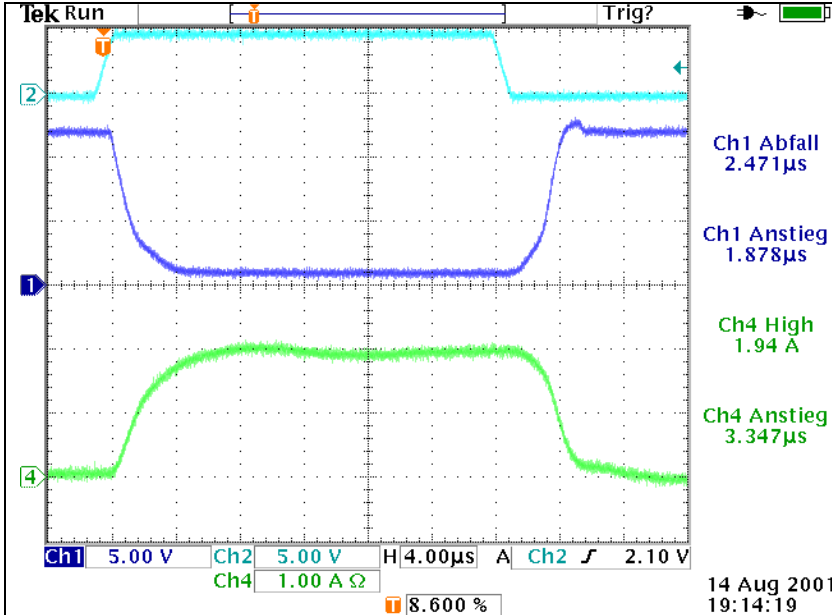
However, especially in the thermal characteristics, conservative values are given in the datasheet which would lead to too pessimistic simulation results. Therefore, electrothermal characteristics have been modeled with more realistic assumptions and must therefore be validated separately.

A good method is to apply short circuit conditions because the load current level and the toggling frequency is an excellent criterion for electrothermal validation.

Similar considerations apply to the dV/dt characteristics if EMI investigations have to be performed.

TLE6240 Model Validation Results

Test Equipment: hp81104A Pulse/Pattern Generator
TDS 3054 Oscilloscope w. active current probe (Ch.4)
EM Test VDS 200 Power Generator/Amplifier
Supply: Vcc = 5.0V, 10nF Bypass

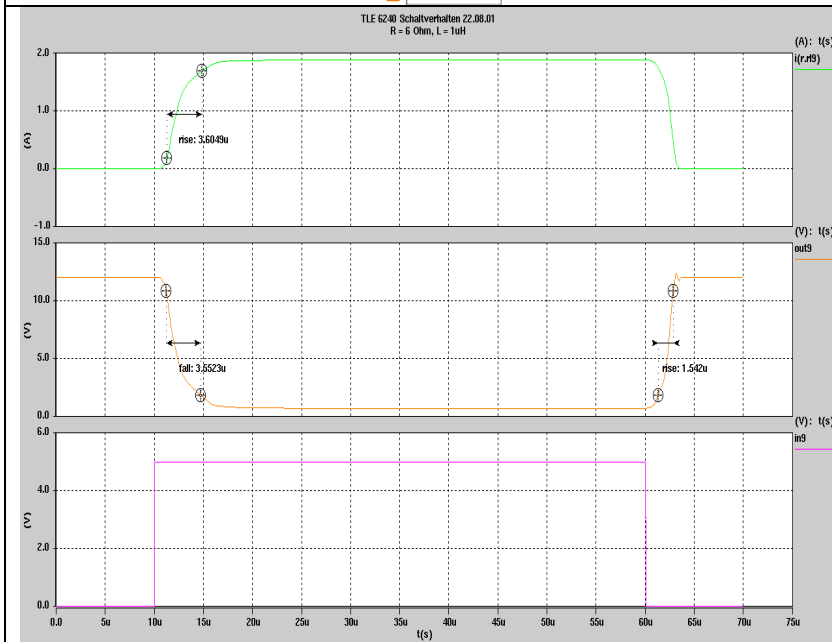


Measured switching cycle at Out9

Tcase = 25°C
Vbat = 12V
R = 6 Ohm
L = 0 (purely ohmic load)

Ch. 2: In9
Ch.1: Out9
Ch.4: Current (Out9)

14 Aug 2001
19:14:19

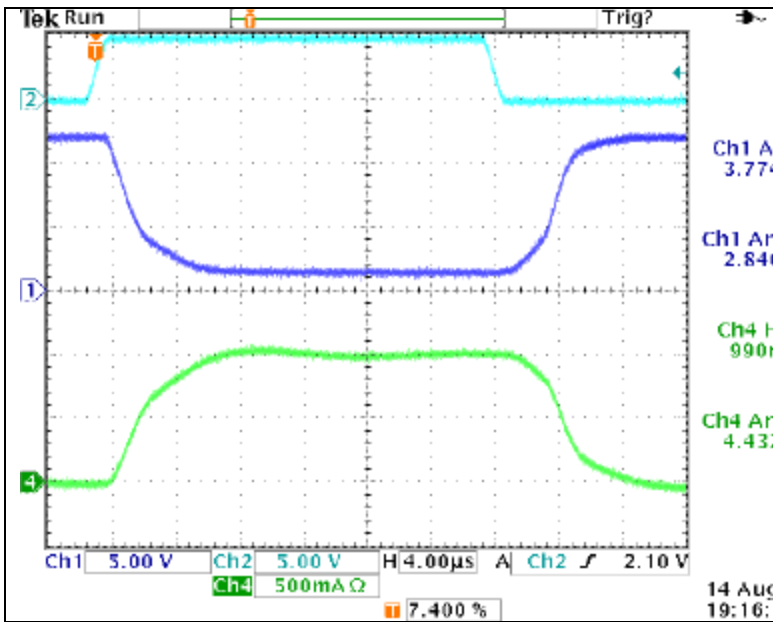


Simulation of switching cycle at Out9

R = 6 Ohm
L = 1µH
Vbat = 12V
Tcase = 25°C

Switching times:

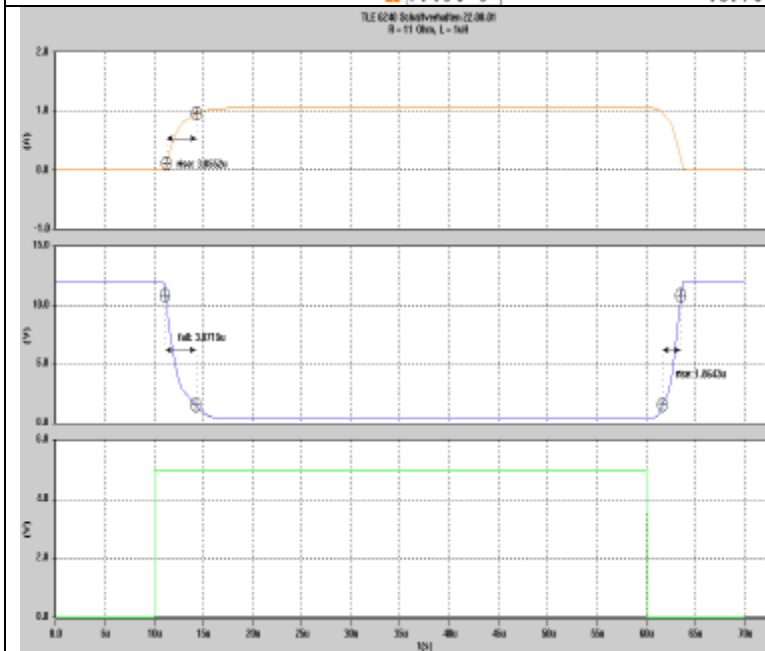
Risetime Out9 = 1.542µs (measured 1.88µs)
Falltime Out9 = 3.5523µs (measured 2.47µs)
Risetime I9 = 3.6049µs (measured 3.35µs)



Measured switching cycle at Out1

T_{case} = 25°C
V_{bat} = 12V
R = 11 Ohm
L = 0 (purely ohmic load)

Ch. 2: In1
Ch.1: Out1
Ch.4: current (Out1)



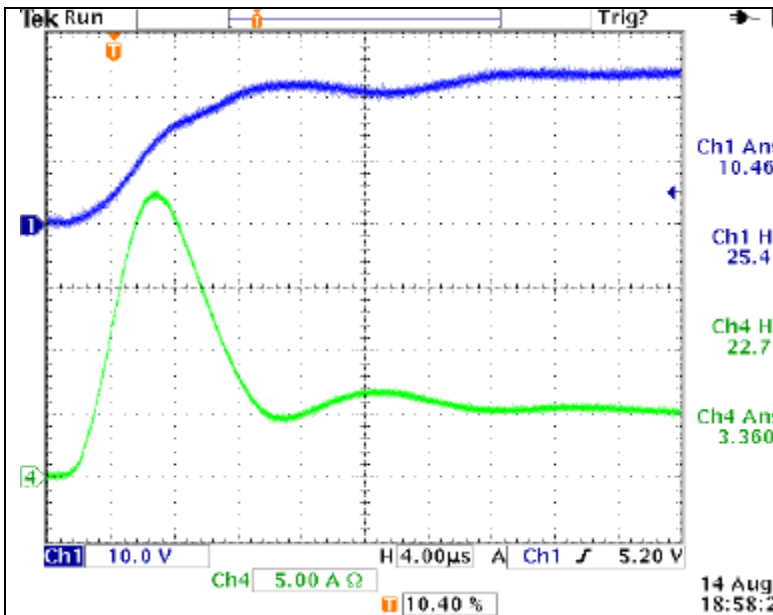
Simulation switching cycle Out1

R = 11 Ohm
L = 1µH
V_{bat} = 12V
T_{case} = 25°C

Risetime Out9 = 1.8643µs (measured 2.85µs)
Falltime Out9 = 3.0715µs (measured 3.77µs)
Risetime I9 = 3.0552µs (measured 4.43µs)



Application Note Multichannel HITFET[®] Behavioural SABER – Model TLE6240GP



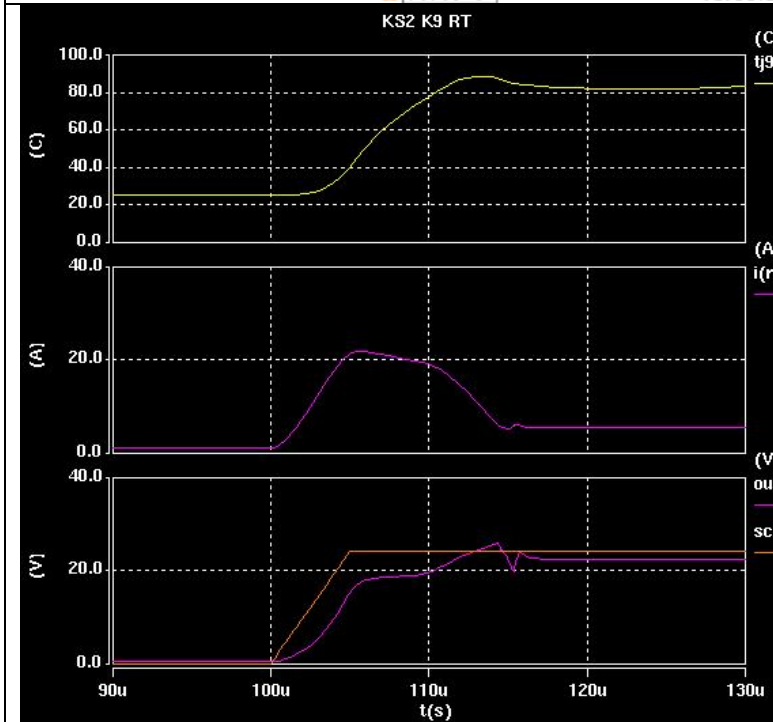
Measurement Short circuit 2 at Out9

Current Overshoot @ Vbat= 24V

Tcase = 25°C
Vbat = 24V
In9 = 5V (static on)
R ≈ 0.1 Ohm, L ≈ 1µH (wiring)

Ch.1: Out9
Ch.4: Current (Out9)

(the voltage slope is limited by the external pulse generator)



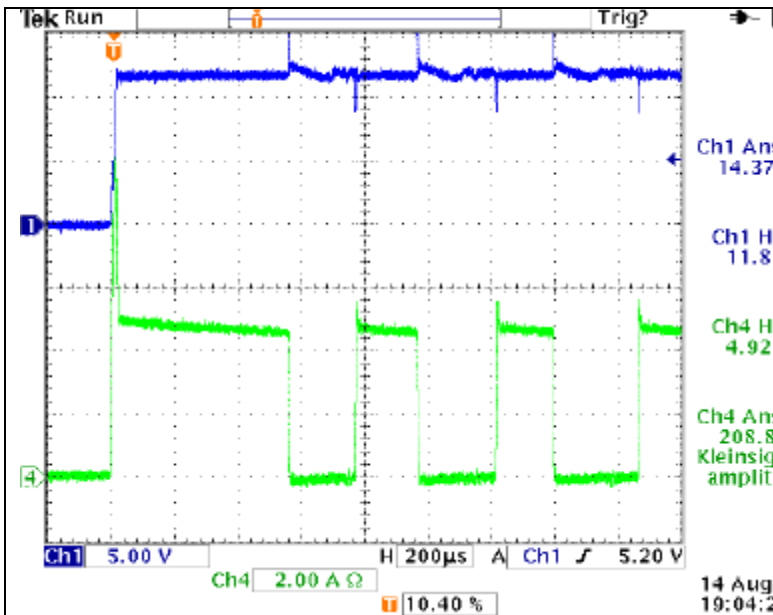
Simulation Shortcircuit 2 atOut9

R = 0.1 Ohm
L = 1µH
Vbat = 24V
Tcase = 25°C

Current Maximum 22A (measured: 22.5A)
Current limitation: 5A (measured 5A)
Junction temperature rise ΔT=65K



Application Note Multichannel HITFET[®] Behavioural SABER – Model TLE6240GP



Measurement:

Continuous Shortcircuit (2) at Out9 Thermal Toggling @ 12V, 125°C

T_{case} = 125°C (by self heating)

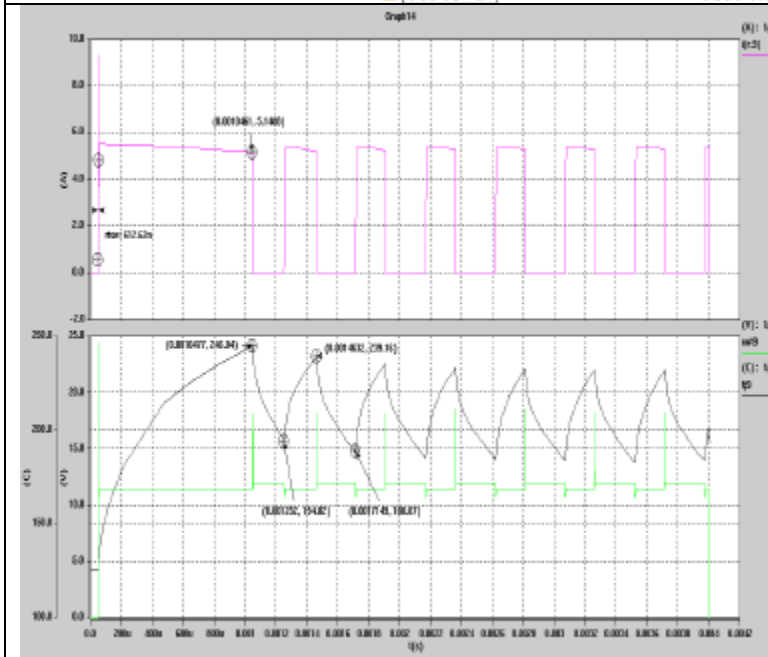
V_{bat} = 12V

I_{n9} = 5V (static on)

R ca. 0.1 Ohm, L ca. 1µH (wiring)

Ch.1: Out9

Ch.4: Current (Out9)



Simulation thermal Toggling at Out9

R = 0.1 Ohm

L = 1µH

V_{bat} = 12V

T_{case} = 125°C

I_{n9} = 5V

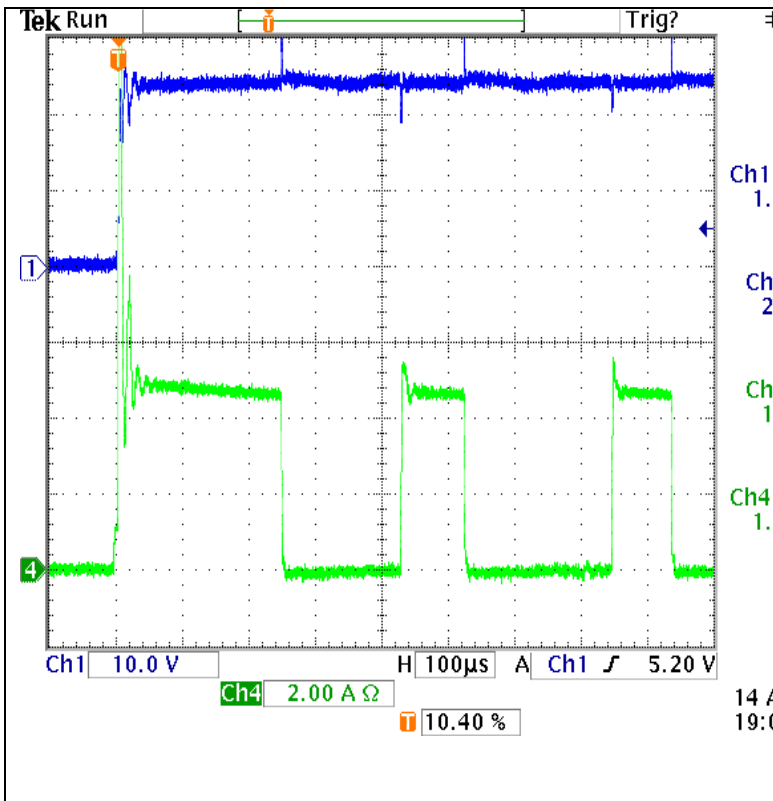
Duration first pulse: 950µs (measured: 560µs)

Toggling period: 450µs (measured: 420µs)

Maximum T_j: 245°C (no measurement)



Application Note Multichannel HITFET[®] Behavioural SABER – Model TLE6240GP



Tek00012.pcx

Continuous short Circuit at Out9
Thermal toggling @ 24V, 125°C

Tcase = 125°C

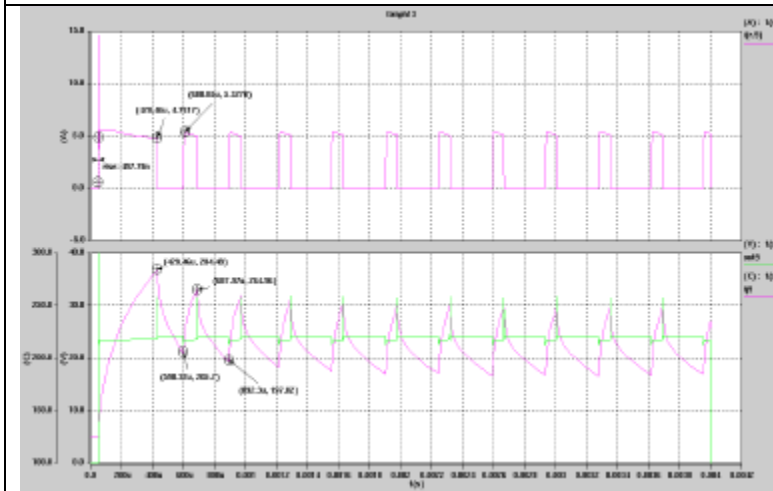
Vbat = 24V

In1 = 5V (static On)

R ca. 0.1 Ohm, L ca. 1μH (Wiring)

Ch.1: Out9

Ch.4: Strom (Out9)



Simulation thermalToggling at Out9

R = 0.1 Ohm

L = 1μH

Vbat = 24V

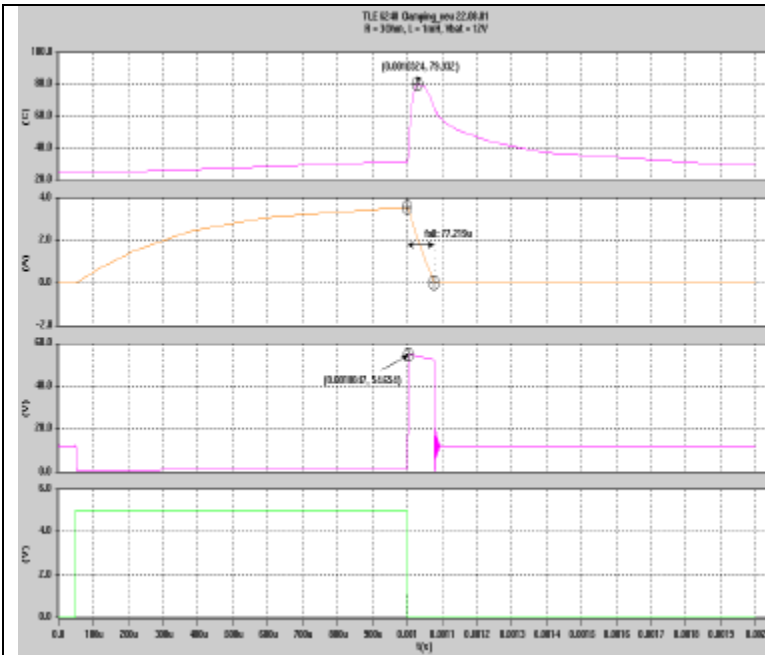
Tcase = 125°C

In9 = 5V

Duration 1st pulse: 350μs (measured: 260μs)

Toggling period: 350μs (measured: 320μs)

Maximum Tj: 284°C (no measurement)

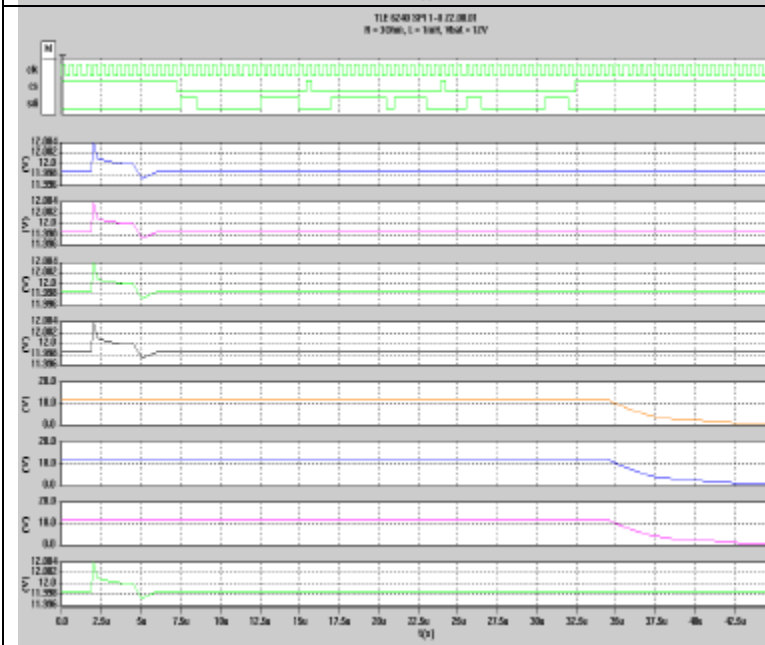


Simulation Clamping Out9

R = 3 Ohm
L = 1mH
Vbat = 12V
Tcase = 25°C

Falltime I9 = 77.219µs

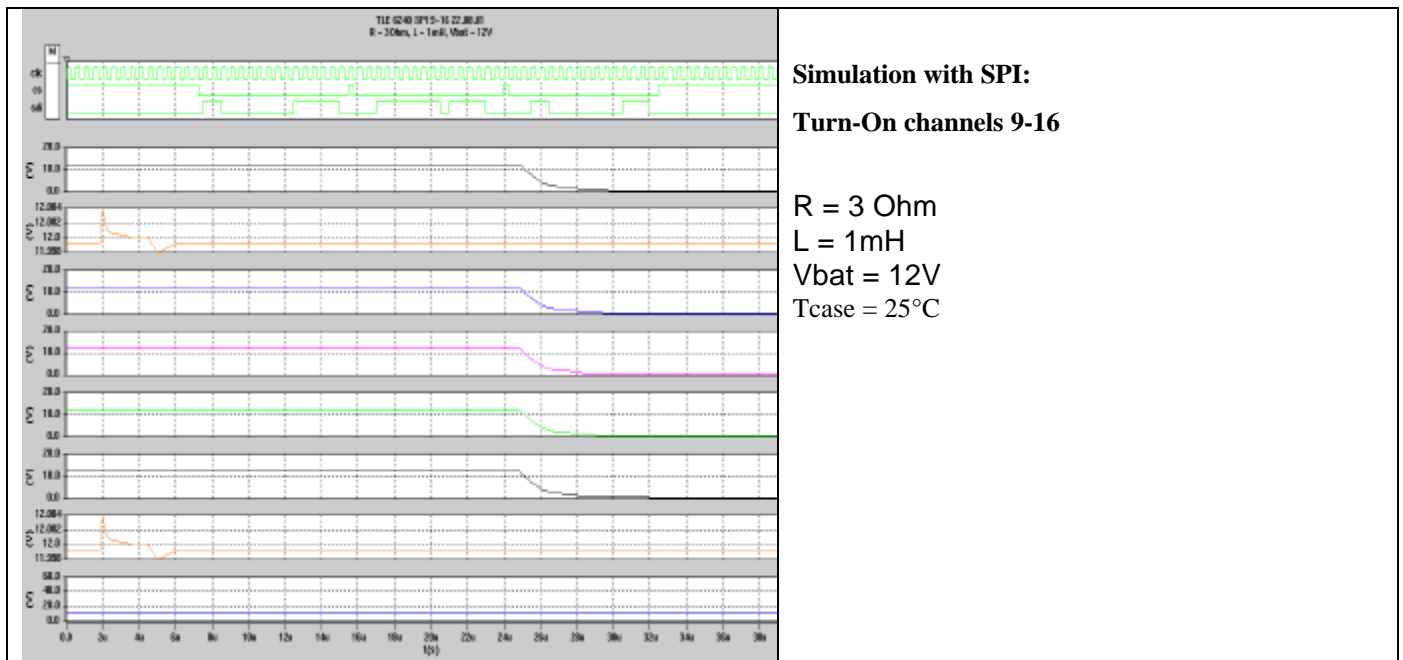
Clamping voltage = 54.654V
(Datasheet: min 45V, typ 50V, max 60V)



Simulation with SPI

Turn on channels 1-8

R = 3 Ohm
L = 1mH
Vbat = 12V
Tcase = 25°C



G. Installation hints, Simulator settings.

The selfextracting ZIP file „TLE6240.exe“ contains the following files:

- This application note: “AppNoteTLE6240_SABER.pdf”
- The include-file „components_S0947.sin“:(encrypted) functional components of integrated functions
- The model file (template) „tle6240.sin“ which uses the previously mentioned components
- One test circuit „test_tle6240.sin“.
- SPI example files from Chapter E: “1bit_diagnosis.sin”, “anded.sin”, “echo_on.sin”, “ored.sin”, “full_diagnosis.sin”.

For installation, copy all files into the simulation directory or into the saber data path

The **recommended simulator settings** are different from default only for transient analysis:

Saber> tr (tn 10..... „target newton iterations“

This is necessary because of the highly nonlinear nature of the model. Ignoring this advice will lead to extremely small timesteps, long simulation time and convergence problems.

For technical support please contact: simulate@infineon.com