Technical Note NXP-R-TN-2006/00302

Issued: 4/2008

From BSIM3/4 to PSP

Conversion of flicker noise, junction, stress, and WPE parameters

A.J. Scholten, G.D.J. Smit, and D.B.M. Klaassen
(NXP-TSMC Research Center)

R. van Langevelde (Philips Research Europe)

G. Gildenblat, X. Li, and W. Wu
(Arizona State University)

Unclassified Report

PSP developers

- At NXP-TSMC Research Center, Eindhoven
  - Geert D.J. Smit
  - Andries J. Scholten
  - Dirk B.M. Klaassen
- At Philips Research Europe
  - Ronald van Langevelde
- At Arizona State University
  - Gennady Gildenblat
  - Hailing Wang (until 2005)
  - Xin Li
  - Weimin Wu

Authors' address

Authors' address	G.D.J. Smit	Gert-Jan.Smit@nxp.com
	A.J. Scholten	Andries.Scholten@nxp.com
	D.B.M. Klaassen	D.B.M.Klaassen@nxp.com
	G. Gildenblat	Gennady.Gildenblat@asu.edu

Title: From BSIM3/4 to PSP

Author(s): G.D.J. Smit, A.J. Scholten, and D.B.M. Klaassen (NXP Research)
Ronald van Langevelde (Philips Research Europe)
G. Gildenblat, X. Li, and W. Wu (Arizona State University)

Reviewer(s):

Technical Note: NXP-R-TN-2006/00302

Additional Numbers:

Subcategory:

Project: Compact modeling

Customer: NXP Semiconductors

Keywords: PSP Model, JUNCAP2 model, compact modelling, MOSFET, CMOS, circuit simulation, integrated circuits, stress, well proximity, noise, junctions

Abstract: In December 2005, the Compact Model Council (CMC) elected PSP to become the new industrial standard model for compact MOSFET modeling. This document provides BSIM3/4-to-PSP translation schemes for flicker noise, junction, stress, and WPE parameters, that may be useful for those who want to switch from BSIM3/4 to PSP.

Report history

April 2006

First release.

April 2008

- Added a chapter on stress and WPE parameters.
- Updated report to latest model version PSP 102.3.
- Updated typography and fixed several minor errors and typos.

Contents

Section 1

Introduction

This document is intended to provide a quick start with (i) the PSP flicker noise model and (ii) the JUNCAP2 junction model for those who are familiar with their BSIM3 or BSIM4 equivalents. Conversion schemes from BSIM3 and BSIM4 flicker noise and junction parameters to PSP and JUNCAP2 parameters are provided. The reader should be warned, however, that a full one-to-one translation is in general not possible. These translations are merely intended to provide BSIM users that are switching over to PSP with initial parameter sets.

Stress and WPE are also covered in this report. These models are virtually identical in BSIM in PSP. However, there are some important differences, which require close attention to avoid mistakes.

Throughout this report, BSIM parameters are typeset like `this`, while PSP parameters are typeset like **THIS**.

In Chapter 2 we will treat the translation of flicker noise parameters, and in Chapter 3 we will treat junction parameters. Finally, stress and well-proximity parameters are treated in Chapter 4. The translation schemes provided are verified using examples run in a circuit simulator.

Section 2

Flicker noise parameters

2.1 Introduction

The PSP flicker noise model has the same physical background [1] as that of the unified noise models in BSIM3v3 (`noimod` $=$ 2 or 3) and BSIM4 (`fnoimod` $=$ 1). Therefore, an approximate translation of the BSIM noise parameters into the corresponding PSP parameters is possible and will be outlined in this section. For other flicker noise models selected in BSIM3/4, no straightforward translation is possible.

2.2 BSIM3

2.2.1 Translation scheme

In BSIM3v3 the so-called unified flicker noise model is selected by setting the switch `noimod` to 2 or 3. In that case, the flicker noise is calculated using the parameters listed in Table 2.1.

The PSP flicker noise model has the following differences w.r.t the BSIM3v3 model:

- PSP-equivalents for `lintnoi` and `ef` are available from PSP 102.3 onwards. Note that PSP's counterpart of `lintnoi` is available only in the geometrical scaling model, not in the binning model.
- In PSP the calculations are done in a surface-potential based framework instead of a threshold-voltage-based framework. Therefore, the transition between weak and strong inversion is smoother and more physical in PSP.

The parameters `noia`, `noib`, and `noic` can be converted into the corresponding PSP parameters (see also [2]) using Table 2.2.

Table 2.1: Overview of BSIM3v3 unified flicker noise model parameters

BSIM3v3 parameter	**Description**
`noia`	noise parameter A
`noib`	noise parameter B
`noic`	noise parameter C
`ef`	flicker exponent
`lintnoi`	length reduction parameter offset

Table 2.2: Translation of BSIM3v3 to PSP flicker noise parameters

PSP parameter	**Value**
local model parameters	
NFA	$\frac{\mathtt{noia}}{10^8 \cdot W_\mathrm{E} \cdot L_\mathrm{E}}$
NFB	$\frac{\mathtt{noib}}{10^8 \cdot W_\mathrm{E} \cdot L_\mathrm{E}}$
NFC	$\frac{\mathtt{noic}}{10^8 \cdot W_\mathrm{E} \cdot L_\mathrm{E}}$
EF	`ef`
global model parameters	
NFALW	$\mathtt{noia} \cdot 10^4$
NFBLW	$\mathtt{noib} \cdot 10^4$
NFCLW	$\mathtt{noic} \cdot 10^4$
EFO	`ef`
LINTNOI	`lintnoi`
ALPNOI	2

PSP parameter	**Value**
binning model parameters	
PONFA	0
PONFB	0
PONFC	0
PLNFA	0
PLNFB	0
PLNFC	0
PWNFA	0
PWNFB	0
PWNFC	0
PLWNFA	$\mathtt{noia} \cdot 10^4$
PLWNFB	$\mathtt{noib} \cdot 10^4$
PLWNFC	$\mathtt{noic} \cdot 10^4$
POEF	`ef`

2.2.2 Verification

The BSIM3v3-to-PSP translation scheme for flicker noise has been tested using the Spectre simulator. With BSIM3v3, IV and CV simulations have been performed for a 10/10 μm n-channel device from a 90-nm CMOS technology. A PSP local parameter set has been generated which fits to the resulting IV and CV curves. In the BSIM3v3 the flicker noise parameters `noia`, `noib`, and `noic` have been set to their default values. The corresponding PSP parameters have been found using the translation scheme of Table 2.2. The noise of the BSIM3v3 and PSP models has been compared using a two-port noise analysis in Spectre. The flicker noise voltage spectral density on the 50 Ω output port is compared in Fig. 2.1(a) for BSIM3v3 and PSP. BSIM3v3 shows behavior which cannot be reproduced in PSP (which has a smoother and more realistic shape). The BSIM3v3 behavior is related to the parameter `noic`. When `noic`, and thus **NFC** in PSP, is set to zero, the agreement between BSIM3v3 and PSP is much better, as shown in Fig. 2.1(b).

In conclusion, the BSIM3v3-to-PSP translation scheme for flicker noise gives reasonable results, but a full one-to-one agreement is evidently not possible due to fundamental differences in the core model.

2.3 BSIM4

2.3.1 Translation scheme

In BSIM4 the unified flicker noise model is selected by setting the switch `fnoimod` to 1. In that case, the flicker noise is calculated using the parameters listed in Table 2.3.

The BSIM4 unified flicker noise model is very similar to its BSIM3v3 equivalent. Therefore, just as in the BSIM3v3 case, one can translate the parameters `noia`, `noib`, and `noic` into the corresponding PSP parame-

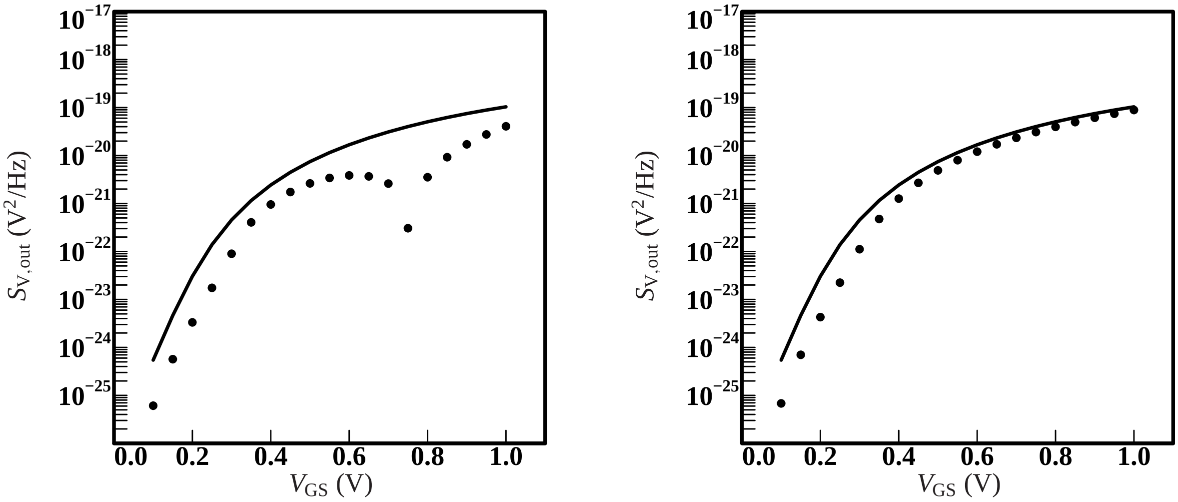

Figure 2.1: (*left*) Comparison of BSIM3v3 (markers) and PSP (line) flicker noise simulation in Spectre for a 10/10 μm n-channel device at a frequency of $f = 1$ kHz. The drain-source voltage is 1 V. BSIM3v3 parameters were set to their default values $\texttt{noia} = 1 \cdot 10^{20}$, $\texttt{noib} = 5 \cdot 10^{4}$, and $\texttt{noic} = -1.4 \cdot 10^{-12}$. The corresponding PSP parameters were $\mathbf{NFA} = 1 \cdot 10^{22}$, $\mathbf{NFB} = 5 \cdot 10^{6}$, and $\mathbf{NFC} = -1.4 \cdot 10^{-10}$. (*right*) Same, but now with $\texttt{noic} = 0$ and $\mathbf{NFC} = 0$.

Table 2.3: Overview of BSIM4 unified flicker noise model parameters

BSIM4 parameter	**Description**
`noia`	Flicker noise parameter A
`noib`	Flicker noise parameter B
`noic`	Flicker noise parameter C
`ef`	Flicker noise frequency exponent
`lintnoi`	length reduction parameter offset

ters. Note however that the BSIM4 parametrization is slightly different as compared to BSIM3v3:

- In BSIM3v3, the value of γ_{ox}, i.e., the attenuation coefficient of the electron wave function in the gate oxide, was fixed to 10^8 (which is the correct value when expressed in cm^{-1}). In BSIM4, however, this value is fixed to 10^{10} (which is the correct value when expressed in m^{-1}).
- In BSIM4, an additional factor q (i.e., elementary charge) has been included in the pre-factor of the flicker noise equation.

As a consequence, the values of the BSIM4 flicker noise parameters need to be $100/q$ times larger than their BSIM3v3 counterparts to get the same amount of flicker noise. Similarly, when we translate BSIM4 flicker noise parameters into PSP flicker noise parameters, we need to include an additional factor of $q/100$ as compared to the BSIM3v3 translation scheme of Table 2.2. This leads to the translation scheme as given in Table 2.4.

2.3.2 Verification

The BSIM4-to-PSP translation scheme for flicker noise has been tested using the Spectre simulator. With BSIM4, IV and CV simulations have been performed for a 10/10 μm n-channel device from a 90 nm CMOS

Table 2.4: Translation of BSIM4 to PSP flicker noise parameters. The symbol q denotes the elementary charge.

PSP parameter	Value
local model parameters	
NFA	$\frac{q \cdot \texttt{noia}}{10^{10} \cdot W_\mathrm{E} \cdot L_\mathrm{E}}$
NFB	$\frac{q \cdot \texttt{noib}}{10^{10} \cdot W_\mathrm{E} \cdot L_\mathrm{E}}$
NFC	$\frac{q \cdot \texttt{noic}}{10^{10} \cdot W_\mathrm{E} \cdot L_\mathrm{E}}$
EF	`ef`
global model parameters	
NFALW	$q \cdot \texttt{noia} \cdot 10^2$
NFBLW	$q \cdot \texttt{noib} \cdot 10^2$
NFCLW	$q \cdot \texttt{noic} \cdot 10^2$
EFO	`ef`
LINTNOI	`lintnoi`
ALPNOI	2

PSP parameter	Value
binning model parameters	
PONFA	0
PONFB	0
PONFC	0
PLNFA	0
PLNFB	0
PLNFC	0
PWNFA	0
PWNFB	0
PWNFC	0
PLWNFA	$q \cdot \texttt{noia} \cdot 10^2$
PLWNFB	$q \cdot \texttt{noib} \cdot 10^2$
PLWNFC	$q \cdot \texttt{noic} \cdot 10^2$
POEF	`ef`

technology. A PSP local parameter set has been generated which fits to these IV and CV curves. In the BSIM4 the flicker noise parameters `noia`, `noib`, and `noic` have been set to their default values. The corresponding PSP parameters have been found using the translation scheme of Table 2.4. The noise of the BSIM3v3 and PSP models has been compared using a two-port noise analysis in Spectre. The flicker noise voltage spectral density on the 50 Ω output port is compared in Fig. 2.2 for BSIM4 and PSP. Excellent agreement is observed, much better than in the BSIM3v3 case.

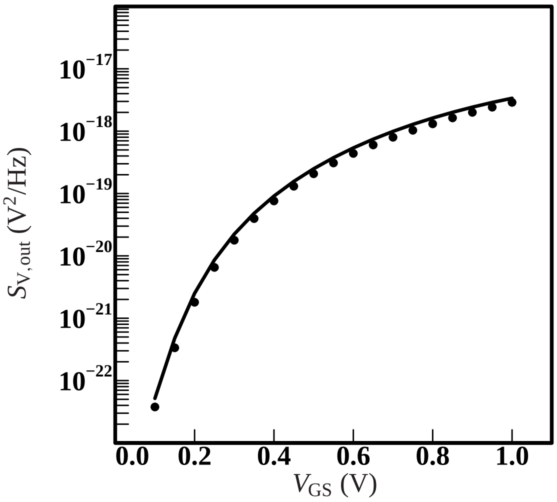

Figure 2.2: Comparison of BSIM4 (markers) and PSP (line) flicker noise simulation in Spectre for a 10/10 μm n-channel device at a frequency of $f = 1$ kHz. The drain-source voltage is 1 V. BSIM4 parameters were set to their default values `noia` $= 6.25 \cdot 10^{41}$, `noib` $= 3.125 \cdot 10^{26}$, and `noic` $= 8.75 \cdot 10^{9}$. The corresponding PSP parameters were **NFA** $= 1 \cdot 10^{23}$, **NFB** $= 5 \cdot 10^{7}$, and **NFC** $= 1.4 \cdot 10^{-9}$.

Section 3

Junction parameters

3.1 Introduction

This chapter is intended to provide a quick start with the JUNCAP2 junction model for those who are familiar with the BSIM3 or BSIM4 junction models. Conversion schemes from BSIM3 and BSIM4 junction parameters to JUNCAP2 parameters are provided.

3.2 BSIM3

3.2.1 Introduction

The information in this section is based on the documentation of BSIM3v3.3, as found on the BSIM website [3]. The junction parameters taken from this document are listed in the table below. It was found that the junction parameter list in the Spectre simulator was extended w.r.t. the Berkeley documentation (e.g., parameters `cbs`, `cbd`). Thus, readers should be careful in applying the information in this chapter to their own situation, because their BSIM3v3.3 version/implementation may also deviate from the description in the BSIM3v3.3 manual.

Name	**Description**	**Unit**
`as`	source junction area	m^2
`ps`	source junction perimeter	m
`ad`	drain junction area	m^2
`pd`	drain junction perimeter	m
`tnom`	temperature at which parameters are extracted	K
`cj`	bottom junction capacitance per unit area at zero bias	F/m^2
`mj`	bottom junction capacitance grading coefficient	-
`pb`	bottom junction built-in potential	V
`cjsw`	source/drain sidewall junction capacitance per unit length at zero bias	F/m
`mjsw`	source/drain sidewall junction capacitance grading coefficient	-

continued on next page…

. . . continued from previous page

Name	**Description**	**Unit**
`pbsw`	source/drain sidewall junction built-in potential	V
`cjswg`	source/drain gate side wall junction capacitance per unit length at zero bias	F/m
`mjswg`	source/drain gate side wall junction capacitance grading coefficient	-
`pbswg`	source/drain gate side wall junction built-in potential	V
`tpb`	temperature coefficient of `pb`	V/K
`tpbsw`	temperature coefficient of `pbsw`	V/K
`tpbswg`	temperature coefficient of `pbswg`	V/K
`tcj`	temperature coefficient of `cj`	1/K
`tcjsw`	temperature coefficient of `cjsw`	1/K
`tcjswg`	temperature coefficient of `cjswg`	1/K
`js`	saturation current density	A/m^2
`jssw`	side wall saturation current density	A/m
`nj`	emission coefficient	-
`xti`	junction current temperature exponent coefficient	-
`ijth`	limiting current	A

3.2.2 Instance parameters

The JUNCAP2 stand-alone model has inherited its instance parameters **AB**, **LS**, and **LG** from the JUNCAP (LEVEL=1) model. Here **AB** is the junction area, **LS** the junction isolation sidewall perimeter, and **LG** the junction gate edge sidewall perimeter. It is important to realize that the instance parameters in BSIM3 are defined somewhat differently. In BSIM3, the instance parameters `AS` and `AD` represent the junction area of source and drain, respectively[1]. The instance parameters `PS` and `PD` represent the *total* junction perimeter (isolation sidewall plus gate sidewall). In Table 3.2, the translation from BSIM3 to JUNCAP2 instance parameters is given. Here, W_E represents the BSIM3 effective width of the MOSFET.

For JUNCAP2 as embedded in the PSP MOSFET model, the junction dimensions can be specified in different ways, depending on the value of the switch **SWJUNCAP**. For all details on this, please refer to [4].

3.2.3 Junction capacitance parameters

Just like JUNCAP2, the BSIM3v3.3 junction capacitance model distinguishes bottom, isolation sidewall, and gate sidewall components of the capacitances.

Moreover, the BSIM3v3.3 junction capacitance model is parameterized in terms of a built-in voltage, grading coefficient, and zero-bias capacitance, just as JUNCAP2.

Nevertheless, the JUNCAP2 capacitance model is not fully compatible with the BSIM3v3.3 junction capacitance model. The reason for this is that the BSIM3v3.3 junction capacitance model, as compared to JUNCAP2,

---

[1] Note that BSIM3, in contrast to BSIM4, does not have the `permod` switch, discussed in section 3.3.2.

Table 3.2: Translation of BSIM3 junction instance parameters to JUNCAP2 instance parameters.

JUNCAP2-parameter	**Value**	
	source	drain
AB	`as`	`ad`
LS	`ps` $- W_\mathrm{E}$	`pd` $- W_\mathrm{E}$
LG	W_E	W_E

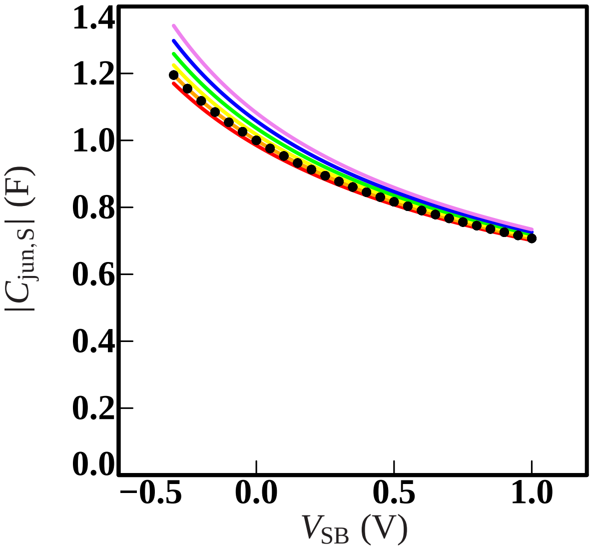

Figure 3.1: Comparison of BSIM3v3.3 simulation of junction capacitance with a JUNCAP2 simulation. The BSIM3v3.3 simulations showed no temperature dependence and are indicated with black markers. The JUNCAP2 simulations are indicated by colored lines, where the colors red, orange, yellow, green, blue, and violet indicate temperatures of -40, 0, 40, 80, 120, and 160 °C, respectively. The relevant BSIM3v3.3 parameters are `tnom` $= 40$, `cj` $= 1000$, `mj` $= 0.5$, `pb` $= 1$, `tcj` $= 0$, and `tpb` $= 0$. The corresponding JUNCAP2 parameters are **TRJ** $= 40$, **CJORBOT** $= 1000$, **PBOT** $= 0.5$, **VBIRBOT** $= 1$, **PHIGBOT** $= 1.16$.

has additional temperature scaling parameters for built-in voltages (`tpb`, `tpbsw`, and `tpbswg`), and zero-bias capacitances (`tcj`, `tcjsw`, `tcjswg`). For JUNCAP2, these effects are incorporated in the underlying physics and therefore these BSIM3v3 parameters have no JUNCAP2 equivalents. In addition, the BSIM3v3 handles the junction capacitance under forward bias differently than JUNCAP2.

Assuming that the above-mentioned additional temperature scaling parameters are zero or at a physically reasonable value, one can use the table at the end of this section to translate BSIM3v3.3 junction capacitance parameters into JUNCAP2 parameters. An example comparison of a BSIM3v3.3 simulation of the junction capacitance with a JUNCAP2 simulation is shown in Figure 3.1. Good agreement is observed. For practical situations the temperature dependence of junction capacitance is fairly small, and the different treatment of this in BSIM3v3 and JUNCAP2 only gives minor deviations in the junction capacitance.

3.2.4 Junction leakage current parameters

The BSIM3v3.3 junction leakage model is quite different from the JUNCAP2 model. Some of the major differences are:

- For junction currents, BSIM3v3.3 makes no distinction between isolation sidewall and gate sidewall. (N.B. for junction capacitances this distinction *is* made in BSIM3v3.3.)

- BSIM3v3.3 has no models for Shockley-Read-Hall, trap-assisted tunneling, band-to-band tunneling, and avalanche breakdown. In reverse bias, only ideal diode current is modeled by BSIM3v3.3.
- BSIM3v3.3 models non-ideality in the forward mode of operation using an emission coefficient (or "non-ideality factor"). In JUNCAP2 this is modeled by physical models for Shockley-Read-Hall and trap-assisted tunneling.
- In JUNCAP2 the temperature dependence of the ideal current can be tuned using the **PHIG** parameters, which represent the band gap voltage. In BSIM3v3 the band gap is fixed to 1.16 eV and a non-physical parameter `xti` is used to tune the temperature dependence. These two different descriptions are not fully compatible.

Let us first discuss the special case that the emission coefficient `nj` in BSIM3v3.3 is equal to 1. Now we can make an (approximate) translation to JUNCAP2, which is valid both in forward and reverse mode of operation. In this case, we can write the BSIM3v3.3 expression for the bottom component of the junction as

$$I_{\mathrm{bs}} = A_{\mathrm{s}} \cdot J_{\mathrm{s}} \cdot \left[\exp\left(\frac{q \cdot V_{\mathrm{bs}}}{k_{\mathrm{B}} \cdot T}\right) - 1\right], \tag{3.1}$$

with

$$J_{\mathrm{s}} = J_{\mathrm{s0}} \cdot \exp\left[\frac{E_{\mathrm{g0}}}{V_{\mathrm{tm0}}} - \frac{E_{\mathrm{g}}}{V_{\mathrm{tm}}} + \mathtt{xti} \cdot \ln\left(\frac{T}{T_{\mathrm{nom}}}\right)\right]. \tag{3.2}$$

Here E_{g} and E_{g0} represent the band gap in eV at the device temperature and at T_{nom}, respectively. The BSIM3v3.3 formulas for $E_{\mathrm{g}}(T)$ are the same as those for JUNCAP2, with the exception that the extrapolated zero-temperature band gap is fixed to 1.16 eV in BSIM3v3.3, while it is an adjustable parameter in JUNCAP2. Equating the BSIM3v3.3 and JUNCAP expressions leads to

$$3 \cdot \ln\left(\frac{T_{\mathrm{KD}}}{T_{\mathrm{KR}}}\right) + \frac{\mathbf{PHIGBOT} - 1.16}{\phi_{\mathrm{TR}}} \cdot \left(1 - \frac{\phi_{\mathrm{TR}}}{\phi_{\mathrm{TD}}}\right) = \mathtt{xti} \cdot \ln\left(\frac{T}{T_{\mathrm{nom}}}\right). \tag{3.3}$$

Now we approximate

$$1 - \frac{\phi_{\mathrm{TR}}}{\phi_{\mathrm{TD}}} \approx \ln\left(\frac{T_{\mathrm{KD}}}{T_{\mathrm{KR}}}\right). \tag{3.4}$$

Identifying T_{KD} with T and T_{KR} with T_{nom} we get

$$\mathtt{xti} \approx 3 + \frac{\mathbf{PHIGBOT} - 1.16}{\phi_{\mathrm{TR}}}. \tag{3.5}$$

Solving for **PHIGBOT** yields

$$\mathbf{PHIGBOT} \approx 1.16 + \frac{k_{\mathrm{B}} \cdot T_{\mathrm{nom}}}{q} \cdot (\mathtt{xti} - 3). \tag{3.6}$$

For the more general case that $\mathtt{nj} \neq 1$, there is no straightforward translation from BSIM3v3 to JUNCAP2 in the forward mode of junction operation. Moreover, Eq. (3.6) has to be modified since in BSIM3v3.3 the emission coefficient NJ also influences the temperature dependence of the junction saturation current

$$J_{\mathrm{s}} = J_{\mathrm{s0}} \cdot \exp\left[\frac{\frac{E_{\mathrm{g0}}}{V_{\mathrm{tm0}}} - \frac{E_{\mathrm{g}}}{V_{\mathrm{tm}}} + \mathtt{xti} \cdot \ln\left(\frac{T}{T_{\mathrm{nom}}}\right)}{\mathtt{nj}}\right]. \tag{3.7}$$

If we neglect the T-dependence of E_{g} equating Eq. (3.7) to the JUNCAP2 ideal current equation leads to the following generalization of Eq. (3.6):

$$\mathbf{PHIGBOT} \approx \frac{1.16}{\mathtt{nj}} + \frac{k_{\mathrm{B}} \cdot T_{\mathrm{nom}}}{q} \cdot \left(\frac{\mathtt{xti}}{\mathtt{nj}} - 3\right). \tag{3.8}$$

In Figs. 3.2, and 3.3 (left), the IV results of the BSIM3v3.3 junction model with $\texttt{nj} = 1$ are compared with those of JUNCAP2. Here, we have used Eq. (3.8) to calculate the required value for **PHIGBOT** from the given value for `xti`. The full translation of junction IV-parameters is summarized in the table below. As expected, a close match between BSIM3v3.3 and JUNCAP2 is observed for the reverse behavior in the case that $\texttt{xti} = 3$, which corresponds to **PHIGBOT** $= 1.16$. But also for $\texttt{xti} = -3$, corresponding to **PHIGBOT** $= 1.0$, the match is very good (note that this is already a somewhat unrealistic case, since such a big deviation from 1.16 eV is not found in practice). Only in extremely unrealistic cases, such as $\texttt{xti} = -20$, corresponding to an unrealistic bandgap voltage **PHIGBOT** $= 0.54$, the approximate nature of Eq. (3.8) becomes apparent (Fig. 3.3, left). In the forward mode of operation, the agreement is also very good, except for some deviations at high currents due to the limiting behavior of the forward current in the Spectre simulator which was used for this test.

In Fig. 3.3 (right), a similar comparison is done for non-unity emission coefficient in BSIM3v3.3. Now the slope of the forward IV-curves differs in JUNCAP2, but the temperature scaling of the reverse current is still well fit due to the use of Eq. (3.8).

JUNCAP2 parameter	**Value**
General parameters	
TRJ	`tnom`
IMAX	`ijth`
Capacitance parameters	
CJORBOT	`cj`
CJORSTI	`cjsw`
CJORGAT	`cjswg`
VBIRBOT	`pb`
VBIRSTI	`pbsw`
VBIRGAT	`pbswg`
PBOT	`mj`
PSTI	`mjsw`
PGAT	`mjswg`
Ideal-current parameters	
PHIGBOT	$\frac{1.16}{\texttt{nj}} + \frac{k_{\mathrm{B}} \cdot T_{\mathrm{nom}}}{q} \cdot \left(\frac{\texttt{xti}}{\texttt{nj}} - 3\right)$
PHIGSTI	$\frac{1.16}{\texttt{nj}} + \frac{k_{\mathrm{B}} \cdot T_{\mathrm{nom}}}{q} \cdot \left(\frac{\texttt{xti}}{\texttt{nj}} - 3\right)$
PHIGGAT	$\frac{1.16}{\texttt{nj}} + \frac{k_{\mathrm{B}} \cdot T_{\mathrm{nom}}}{q} \cdot \left(\frac{\texttt{xti}}{\texttt{nj}} - 3\right)$
IDSATRBOT	`js`
IDSATRSTI	`jssw`
IDSATRGAT	`jssw`

continued on next page...

. . . continued from previous page

JUNCAP2 parameter	**Value**
Shockley-Read-Hall parameters	
CSRHBOT	0
CSRHSTI	0
CSRHGAT	0
XJUNSTI	10^{-7}
XJUNGAT	10^{-7}
Trap-assisted tunneling parameters	
CTATBOT	0
CTATSTI	0
CTATGAT	0
MEFFTATBOT	0.25
MEFFTATSTI	0.25
MEFFTATGAT	0.25
Band-to-band tunneling parameters	
CBBTBOT	0
CBBTSTI	0
CBBTGAT	0
FBBTRBOT	$1 \cdot 10^9$
FBBTRSTI	$1 \cdot 10^9$
FBBTRGAT	$1 \cdot 10^9$
STFBBTBOT	0
STFBBTSTI	0
STFBBTGAT	0
Avalanche breakdown parameters	
VBRBOT	1001
VBRSTI	1001
VBRGAT	1001
PBRBOT	4
PBRSTI	4

continued on next page. . .

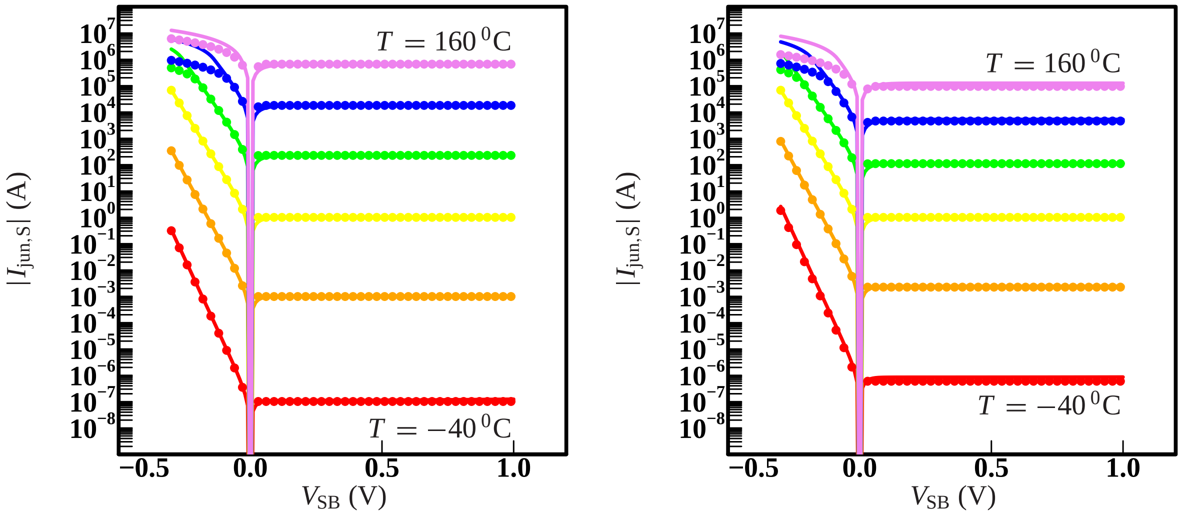

Figure 3.2: (*left*) Comparison of BSIM3v3.3 simulation with `nj` $= 1$ and `xti` $= 3$ (markers) with JUNCAP2 simulation with **PHIGBOT** $= 1.16$ (lines). (*right*) Comparison of BSIM3v3.3 simulation with `nj` $= 1$ and `xti` $= -3$ (markers) with JUNCAP2 simulation with **PHIGBOT** $= 1.0$ (lines). Temperatures are -40, 0, 40, 80, 120, and 160 °C, and are indicated by colors red, orange, yellow, green, blue, and violet, respectively.

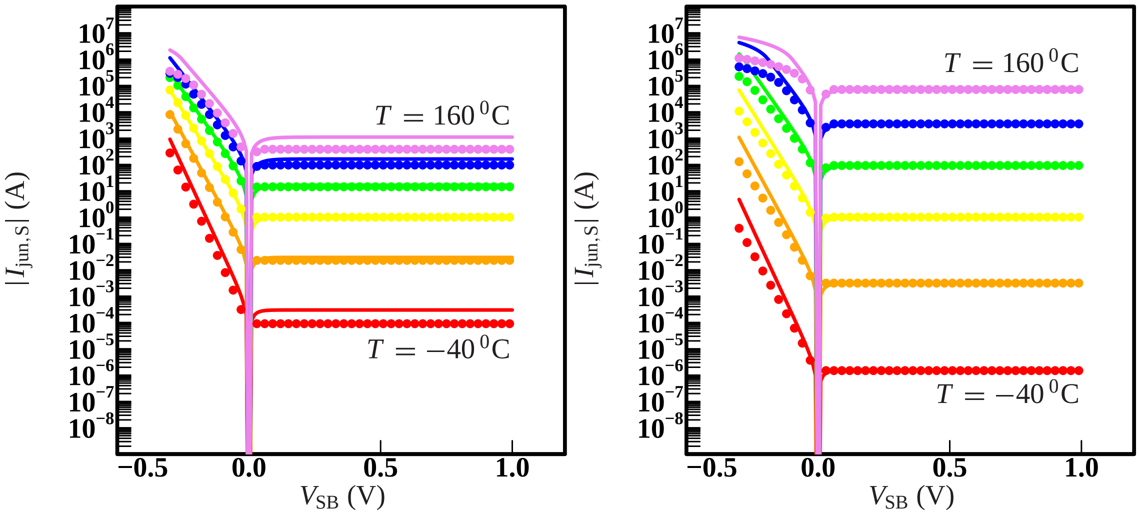

Figure 3.3: (*left*) Comparison of BSIM3v3.3 simulation with `nj` $= 1$ and `xti` $= -20$ (markers) with JUNCAP2 simulation with **PHIGBOT** $= 0.54$ (lines). (*right*) Comparison of BSIM3v3.3 simulation with `nj` $= 1.2$ and `xti` $= 3$ (markers) with JUNCAP2 simulation with **PHIGBOT** $= 0.953$ (lines). Temperatures are -40, 0, 40, 80, 120, and 160 °C, and are indicated by colors red, orange, yellow, green, blue, and violet, respectively.

... continued from previous page

JUNCAP2 parameter	**Value**
PBRGAT	4

3.3 BSIM4

3.3.1 Introduction

The information in this section is based on the documentation of BSIM4.5.0, as found on the BSIM website [3]. The junction parameters taken from this documentation are listed the following table.

Name		**Description**	**Unit**
source	**drain**		
`tnom`		temperature at which parameters are extracted	°C
`cjs`	`cjd`	bottom junction capacitance per unit area at zero bias	F/m^2
`mjs`	`mjd`	bottom junction capacitance grading coefficient	-
`pbs`	`pbd`	bottom junction built-in potential	V
`cjsws`	`cjswd`	isolation-edge sidewall junction capacitance per unit length	F/m
`mjsws`	`mjswd`	isolation-edge sidewall junction capacitance grading coefficient	-
`pbsws`	`pbswd`	isolation-edge sidewall junction built-in potential	V
`cjswgs`	`cjswgd`	gate-edge sidewall junction capacitance per unit length	F/m
`mjswgs`	`mjswgd`	gate-edge sidewall junction capacitance grading coefficient	-
`pbswgs`	`pbswgd`	gate-edge sidewall junction built-in potential	V
`tpb`		temperature coefficient of `pbs`, `pbd`	V/K
`tpbsw`		temperature coefficient of `pbsws`, `pbswd`	V/K
`tpbswg`		temperature coefficient of `pbswgs`, `pbswgd`	V/K
`tcj`		temperature coefficient of `cjs`, `cjd`	1/K
`tcjsw`		temperature coefficient of `cjsws`, `cjswd`	1/K
`tcjswg`		temperature coefficient of `cjswgs`, `cjswgd`	1/K
`jss`	`jsd`	bottom junction reverse saturation current density	A/m^2
`jsws`	`jswd`	isolation-edge sidewall reverse saturation current density	A/m
`jswgs`	`jswgd`	gate-edge sidewall reverse saturation current density	A/m
`njs`	`njd`	emission coefficient of junction	-
`xtis`	`xtid`	junction current temperature exponent	-
`ijthsrev`	`ijthdrev`	limiting current in reverse bias region	A
`ijthsfwd`	`ijthdfwd`	limiting current in forward bias region	A
`xjvbs`	`xjvbd`	fitting parameter for diode breakdown	-
`bvs`	`bvd`	breakdown voltage	V

continued on next page...

. . . continued from previous page

Name		**Description**	**Unit**
source	**drain**		
`jtss`	`jtsd`	bottom trap-assisted saturation current density	A/m^2
`jtssws`	`jtsswd`	STI sidewall trap-assisted saturation current density	A/m
`jtsswgs`	`jtsswgd`	gate-edge sidewall trap-assisted saturation current density	A/m
`vtss`	`vtsd`	bottom trap-assisted voltage dependent parameter	V
`vtssws`	`vtsswd`	STI sidewall trap-assisted voltage dependent parameter	V
`vtsswgs`	`vtsswgd`	gate-edge sidewall trap-assisted voltage dependent parameter	V
`njts`		non-ideality factor for `jtss`, `jtsd`	-
`njtssw`		non-ideality factor for `jtssws`, `jtsswd`	-
`njtsswg`		non-ideality factor for `jtsswgs`, `jtsswgd`	-
`xtss`	`xtsd`	power dependence of `jtss`, `jtsd` on temperature	-
`xtssws`	`xtsswd`	power dependence of `jtssws`, `jtsswd` on temperature	-
`xtsswgs`	`xtsswgd`	power dependence of `jtsswgs`, `jtsswgd` on temperature	-
`tnjts`		temperature coefficient for `njts`	-
`tnjtssw`		temperature coefficient for `njtssw`	-
`tnjtsswg`		temperature coefficient for `njtsswg`	-

3.3.2 Instance parameters

The meaning of the BSIM4 junction instance parameters depends on the value of the switch `permod`. When `permod` = 1 (default), the instance parameters have the same meaning as in BSIM3, and the translation scheme is similar to Table 3.2 for BSIM3. When `permod` = 0 however the parameters `ps` and `pd` only refer to the isolation-edge part of the junction perimeter. In Table 3.5, the translation from BSIM4 to JUNCAP2 instance parameters is given. Here, W_{effcj} represents the BSIM4 effective junction width.

For JUNCAP2 as embedded in the PSP MOSFET model, the junction dimensions can be specified in different ways, depending on the value of the switch **SWJUNCAP**. For all details on this, please refer to [4].

3.3.3 Junction capacitance and leakage current parameters

The translation of BSIM4 to JUNCAP2 junction parameters is very similar to the BSIM3-to-JUNCAP2 translation, discussed in Section 3.2.3. The discussion here will be limited to the items that are specific for BSIM4:

- A large part (not all) of the BSIM4 junction parameters can be set separately for source and drain. A similar construction is available from PSP 102.3 onwards. See below for more details.
- Not only the junction capacitance, but also the junction currents now distinguish a STI sidewall and a gate-edge sidewall contribution. This feature can be translated directly to JUNCAP2.
- A simple junction breakdown model has been added. This feature can be translated directly to JUNCAP2.

Table 3.5: Translation of BSIM4 junction instance parameters to JUNCAP2 instance parameters.

JUNCAP2 parameter	**Value**			
	`permod` = 1 (default)		`permod` = 0	
	source	**drain**	**source**	**drain**
AB	`as`	`ad`	`as`	`ad`
LS	`ps` $- W_{\mathrm{effcj}}$	`pd` $- W_{\mathrm{effcj}}$	`ps`	`pd`
LG	W_{effcj}	W_{effcj}	W_{effcj}	W_{effcj}

- A fit-function based trap-assisted tunneling model has been added. This model is incompatible with the physics-based model for trap-assisted tunneling in JUNCAP2. A straightforward translation is not possible.
- An addition to limiting of forward current, limiting of reverse current is introduced. A similar feature does not exist in JUNCAP2.

Asymmetric junctions

Most of the BSIM4 junction parameters can be set separately for source and drain.

For PSP 102.2 and older, this construction does not exist for JUNCAP2 as embedded in the PSP model. However, in case this feature is needed, one can easily generate a compound model (a.k.a. subcircuit model) consisting of a junction-less PSP model (**SWJUNCAP** $= 0$) and two instances of the stand-alone JUNCAP2 model between source/bulk and drain/bulk terminals, see Fig. 3.4. (N.B. This compound model has *no* internal nodes and thus adds no computational complexity as compared to PSP with built-in junctions.)

From PSP 102.3 onwards, separate sets of junction model parameters are available for the source and drain side junction in PSP. The source-side parameters have the same name as in the symmetric case. The drain-side parameters have an additional ‘D’ in the name (e.g. **PGAT** vs. **PGATD**). By default (**SWJUNASYM** $= 0$), the junctions are considered to be symmetric in PSP and the drain-side parameters are ignored. If **SWJUNASYM** $=$ 1, the drain-side parameters will be used for the drain junction.

Note that there is an important difference between BSIM and PSP in how the asymmetry is invoked and how the model selects which parameter to use (source-side or drain-side parameter).

- In BSIM, the selection is done for each parameter individually. For example, if `cjd` is not given, `cjs` is used for both the source and drain junction. If `cjd` *is* given, `cjd` is used for the drain-side junction, and `cjs` is used for the source-side junction only. A similar mechanism applies to all junction parameters.
- In PSP, the selection is done *collectively*. If **SWJUNASYM** $= 0$ (default), all drain-side parameters are ignored (whether they are given or not) and the source-side parameters are used for both the source and drain junction. If **SWJUNASYM** $= 1$, source-side parameters are used for the source junction and drain-side parameters are used for the drain junction.

  The main difference with BSIM is that (when **SWJUNASYM** $= 1$) if some drain-side parameter is not given, its default value will be used in PSP (that is, *not* the source-side value). As a result, even if only a few junction parameters are different for source and drain, in PSP one must supply the *full* list of drain-side parameters.

Conversion scheme

The table below shows the BSIM4-to-JUNCAP2 junction parameter conversion scheme.

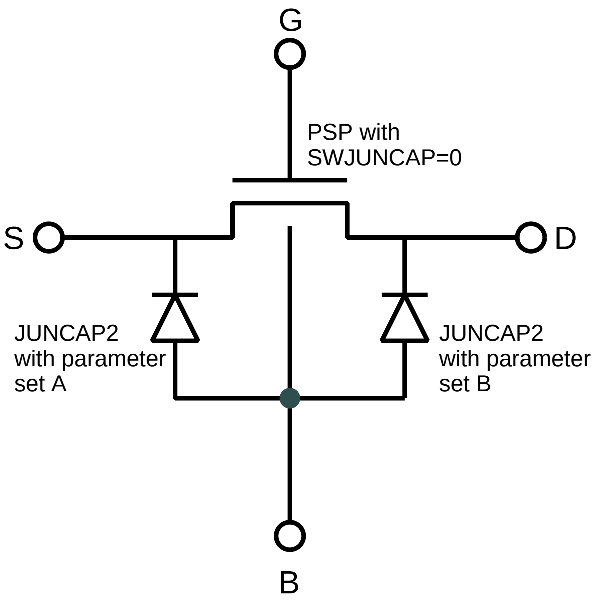

Figure 3.4: Compound model of junction-less PSP and two instances of the stand-alone JUNCAP2 model.

Name	**Value**	
	source	**drain**
General parameters		
TRJ	`tnom`	
IMAX	`ijthsfwd`	`ijthdfwd`
Capacitance parameters		
CJORBOT	`cjs`	`cjd`
CJORSTI	`cjsws`	`cjswd`
CJORGAT	`cjswgs`	`cjswgd`
VBIRBOT	`pbs`	`pbd`
VBIRSTI	`pbsws`	`pbswd`
VBIRGAT	`pbswgs`	`pbswgd`
PBOT	`mjs`	`mjd`
PSTI	`mjsws`	`mjswd`
PGAT	`mjswgs`	`mjswgd`
Ideal-current parameters		
PHIGBOT	$\frac{1.16}{\texttt{njs}} + \frac{k_{\mathrm{B}} \cdot T_{\mathrm{nom}}}{q} \cdot \left(\frac{\texttt{xtis}}{\texttt{njs}} - 3\right)$	$\frac{1.16}{\texttt{njd}} + \frac{k_{\mathrm{B}} \cdot T_{\mathrm{nom}}}{q} \cdot \left(\frac{\texttt{xtid}}{\texttt{njd}} - 3\right)$

continued on next page...

... continued from previous page

Name	**Value**	
	source	**drain**
PHIGSTI	$\frac{1.16}{\texttt{njs}} + \frac{k_\mathrm{B} \cdot T_\mathrm{nom}}{q} \cdot \left(\frac{\texttt{xtis}}{\texttt{njs}} - 3\right)$	$\frac{1.16}{\texttt{njd}} + \frac{k_\mathrm{B} \cdot T_\mathrm{nom}}{q} \cdot \left(\frac{\texttt{xtid}}{\texttt{njd}} - 3\right)$
PHIGGAT	$\frac{1.16}{\texttt{njs}} + \frac{k_\mathrm{B} \cdot T_\mathrm{nom}}{q} \cdot \left(\frac{\texttt{xtis}}{\texttt{njs}} - 3\right)$	$\frac{1.16}{\texttt{njd}} + \frac{k_\mathrm{B} \cdot T_\mathrm{nom}}{q} \cdot \left(\frac{\texttt{xtid}}{\texttt{njd}} - 3\right)$
IDSATRBOT	`jss`	`jsd`
IDSATRSTI	`jsws`	`jswd`
IDSATRGAT	`jswgs`	`jswgd`
Shockley-Read-Hall parameters		
CSRHBOT	0	0
CSRHSTI	0	0
CSRHGAT	0	0
XJUNSTI	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
XJUNGAT	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$
Trap-assisted tunneling parameters		
CTATBOT	0	0
CTATSTI	0	0
CTATGAT	0	0
MEFFTATBOT	0.25	0.25
MEFFTATSTI	0.25	0.25
MEFFTATGAT	0.25	0.25
Band-to-band tunneling parameters		
CBBTBOT	0	0
CBBTSTI	0	0
CBBTGAT	0	0
FBBTRBOT	$1 \cdot 10^{9}$	$1 \cdot 10^{9}$
FBBTRSTI	$1 \cdot 10^{9}$	$1 \cdot 10^{9}$
FBBTRGAT	$1 \cdot 10^{9}$	$1 \cdot 10^{9}$
STFBBTBOT	0	0
STFBBTSTI	0	0

continued on next page...

. . . continued from previous page

Name	**Value**	
	source	**drain**
STFBBTGAT	0	0
	Avalanche breakdown parameters	
VBRBOT	`bvs`	`bvd`
VBRSTI	`bvs`	`bvd`
VBRGAT	`bvs`	`bvd`
PBRBOT	4	4
PBRSTI	4	4
PBRGAT	4	4

3.3.4 Verification

In this section we verify the BSIM4-to-JUNCAP2 translation scheme for junction parameters. Because of the similarity between BSIM3 and BSIM4 junction models the same kind of agreement is expected as for the BSIM3 case discussed before.

In Fig. 3.5, we compare the junction capacitances of BSIM4 with their JUNCAP2 counterparts. As expected, the plot is very similar to Fig. 3.1, and the same comments apply.

In Fig. 3.6, we show some comparisons between BSIM4 junction currents and the JUNCAP2 currents derived from that using the translation scheme. Again, the plots are very similar to their BSIM3 counterparts Fig. 3.2 (left) and 3.3 (right), discussed previously.

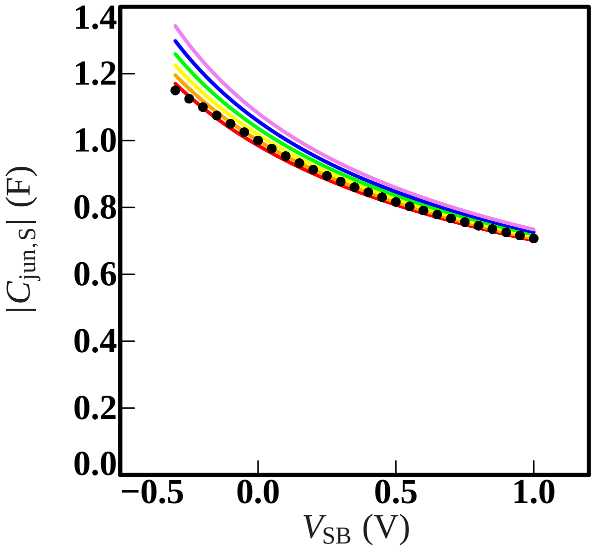

Figure 3.5: Comparison of BSIM4 simulation of junction capacitance with a JUNCAP2 simulation. The BSIM4 simulations showed no temperature dependence and are indicated with black markers. The JUNCAP2 simulations are indicated by colored lines, where the colors red, orange, yellow, green, blue, and violet indicate temperatures of -40, 0, 40, 80, 120, and 160 °C, respectively. The relevant BSIM4 parameters are `tnom` $= 40$, `cj` $= 1000$, `mj` $= 0.5$, `pb` $= 1$, `tcj` $= 0$, and `tpb` $= 0$. The corresponding JUNCAP2 parameters are **TRJ** $= 40$, **CJORBOT** $= 1000$, **PBOT** $= 0.5$, **VBIRBOT** $= 1$, **PHIGBOT** $= 1.16$.

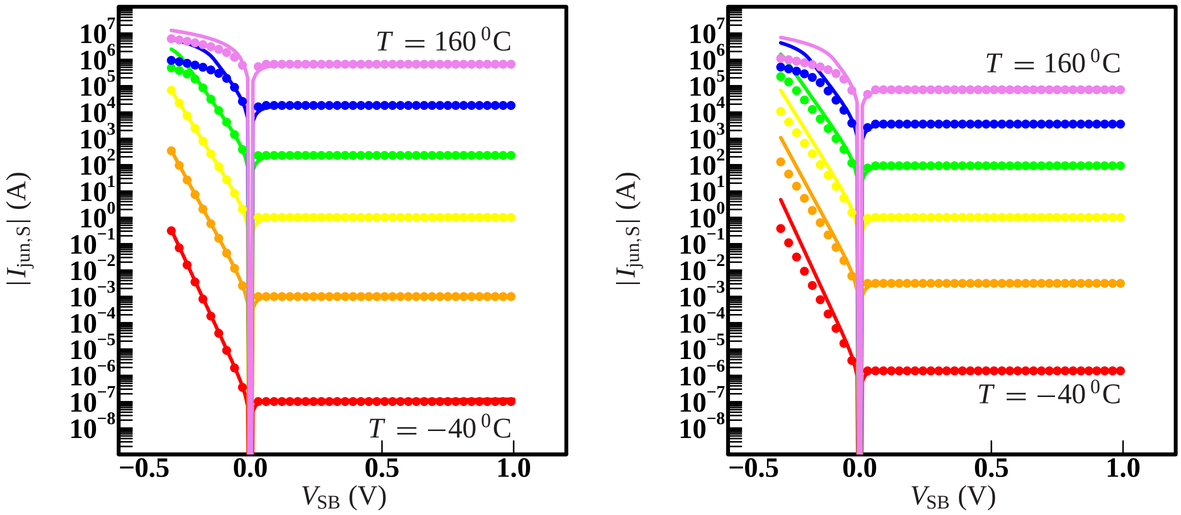

Figure 3.6: (a) Comparison of BSIM4 simulation with `nj` $= 1$ and `xti` $= 3$ (markers) with JUNCAP2 simulation with **PHIGBOT** $= 1.16$ (lines). (b) Comparison of BSIM4 simulation with `nj` $= 1.2$ and `xti` $= 3$ (markers) with JUNCAP2 simulation with **PHIGBOT** $= 0.953$ (lines). Temperatures are -40, 0, 40, 80, 120, and 160 °C, and are indicated by colors red, orange, yellow, green, blue, and violet, respectively.

Section 4

Stress and WPE

4.1 Introduction

In order to model the effect of stress and WPE in PSP, the CMC-standardized models which are incorporated in BSIM are also integrated in PSP.

- These stress and WPE models give layout dependent Vth-shifts and mobility changes. This is sufficiently generic to be used in compact models other than BSIM.
- Extracted model parameters for BSIM can be reused for PSP, without any additional characterization effort.

Although in essence, the stress and WPE models in PSP are carbon copies of those in BSIM, some minor modifications were inevitable. As a result, some simple conversion scheme has to be applied to the model parameters as well. The main reasons for this are:

- Some parameter names have changed to meet PSP naming conventions. In particular, zeros (‘0’) in BSIM-names are changed to O’s (‘O’) in PSP-names.
- The threshold voltage of a p-type device is negative in BSIM, while in PSP the sign of all parameters is chosen as if it were for an n-type device. Therefore, the sign of a V_{th}-shift in in a p-type device in BSIM has to be reversed for PSP.
- The parameter `eta0` in BSIM, which is affected by the stress-model, does not directly reflect DIBL; in stead, this parameter is scaled by a strongly L-dependent prefactor. Therefore, the same prefactor has to be applied when converting the effect of stress on DIBL from BSIM to PSP.
- The L- and W-dependence of the WPE-model in BSIM makes use of BSIM’s binning equations. As a result, the unit of some of these parameters depends on the value of `binunit`. In PSP, the unit is fixed, so in some cases a conversion factor needs to be applied.
- There is no equivalent in PSP of the BSIM parameter `k2` (second order back-bias sensitivity of the threshold voltage). Therefore, the effect of stress on `k2` (through the parameters `stk2` and `lodk2`) cannot be translated to PSP.

4.2 Stress model parameters

4.2.1 Conversion scheme

In Table 4.1, the PSP parameter values are expressed in terms of BSIM parameter values.

Table 4.1: PSP parameter values in terms of BSIM parameter values. The algorithm to compute the value of θ is given in Sec. 4.2.2. For the BSIM parameters `stk2` and `lodk2`, no PSP-equivalent exists.

PSP parameter	NMOS	PMOS
SAREF	`saref`	`saref`
SBREF	`sbref`	`sbref`
WLOD	`wlod`	`wlod`
KUO	`ku0`	`ku0`
KVSAT	`kvsat`	`kvsat`
TKUO	`tku0`	`tku0`
LKUO	`lku0`	`lku0`
WKUO	`wku0`	`wku0`
PKUO	`pku0`	`pku0`
LLODKUO	`llodku0`	`llodku0`
WLODKUO	`wlodku0`	`wlodku0`
KVTHO	`kvth0`	$-$`kvth0`
LKVTHO	`lkvth0`	`lkvth0`
WKVTHO	`wkvth0`	`wkvth0`
PKVTHO	`pkvth0`	`pkvth0`
LLODVTH	`llodvth`	`llodvth`
WLODVTH	`wlodvth`	`wlodvth`
STETAO	$\theta \cdot$ `steta0`	$\theta \cdot$ `steta0`
LODETAO	`lodeta0`	`lodeta0`

4.2.2 Calculation of θ

In BSIM, an L-dependent prefactor is applied in the DIBL model *after* the stress-induced change to `eta0` is applied. Because the L-dependence of DIBL in PSP is different from that in BSIM, it is not possible to exactly copy the effect of stress on $V_{\rm thsat}$ from BSIM to PSP for all channel lengths at the same time.

Because DIBL is most important for the shortest channels, the best choice is to match the PSP value to BSIM for the shortest channel. In practice, this leads to a good match between the models for all channel lengths. Only when the DIBL-component of the stress effect is relatively large, one can expect some differences between PSP and BSIM for intermediate channel lengths.

The algorithm to compute the value of θ is rather involved, due to the complicated L-dependence of DIBL in BSIM. The scheme below is based on BSIM 4.6.0 documentation and source code as issued by UC at Berkeley [3].

Table 4.2: Default values of BSIM parameters needed to compute θ.

BSIM parameter	Default value
`ndep`	$1.7 \cdot 10^{17}$
`phin`	0
`dsub`	`drout` (default of `drout` is 0.56)
`toxe`	$3 \cdot 10^{-9}$
`epsrox`	3.9
`xl`	0
`lint`	0
`ll`	0
`lln`	1
`lw`	0
`lwn`	1
`lwl`	0
`tnom`	27

Required BSIM parameters

Values of several BSIM parameters are needed for this conversion. For completeness, the default values (which need to be used if a parameter is not given in the BSIM modelcard) are also given in Table 4.2. As explained above, one needs to select one specific device geometry for which PSP and BSIM stress-induced DIBL shifts are to be matched. In the equations below, L and W refer to the designed length and width of this particular device (in practice, dependence on width is weak or nonexistent). Note that some of the parameters in the table below may be binned! In that case, one should take care to use the values from the correct bin.

Algorithm

$$dL = \mathtt{lint} + \frac{\mathtt{ll}}{L^{\mathtt{lln}}} + \frac{\mathtt{lw}}{W^{\mathtt{lwn}}} + \frac{\mathtt{lwl}}{L^{\mathtt{lln}} \cdot W^{\mathtt{lwn}}} \tag{4.1}$$

$$L_{\text{eff}} = L + \mathtt{xl} - 2 \cdot dL \tag{4.2}$$

$$\varepsilon_{\text{Si}} = 1.03594 \cdot 10^{-10} \tag{4.3}$$

$$\varepsilon_0 = 8.85418 \cdot 10^{-12} \tag{4.4}$$

$$T_{\text{nom}} = \mathtt{tnom} + 273.15 \tag{4.5}$$

$$K_Q = 8.617087 \cdot 10^{-5} \tag{4.6}$$

$$V_{\text{tm0}} = K_Q \cdot T_{\text{nom}} \tag{4.7}$$

$$E_{\text{g0}} = 1.16 - \frac{7.02 \cdot 10^{-4} \cdot T_{\text{nom}} \cdot T_{\text{nom}}}{T_{\text{nom}} + 1108.0} \tag{4.8}$$

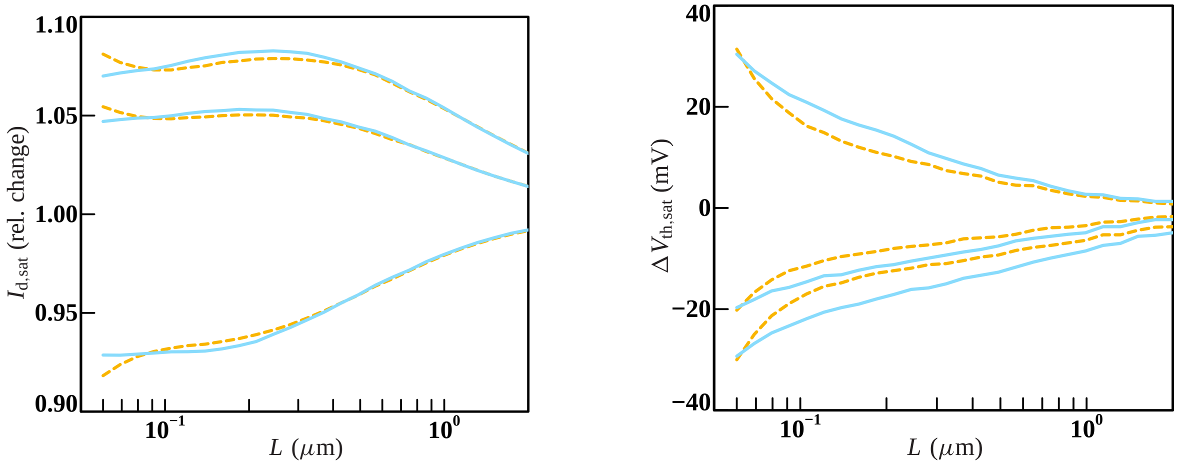

Figure 4.1: Relative change in $I_{\rm d,sat}$ (left) and shift in $V_{\rm th,sat}$ (right) as a function of L for **SA** $= 0.2, 1$ and 3 μm. Dashed lines are BSIM results, solid lines are PSP results. More details are given in the text. Noise in the $V_{\rm th,sat}$-curve is due to limited numerical accuracy.

$$n_{\rm i} = 1.45 \cdot 10^{10} \cdot \left(\frac{T_{\rm nom}}{300.15}\right)^{3/2} \cdot \exp\left(21.5565981 - \frac{E_{\rm g0}}{2.0 \cdot V_{\rm tm0}}\right) \tag{4.9}$$

$$\Psi_{\rm s} = 0.4 + V_{\rm tm0} \cdot \ln \frac{\mathtt{ndep}}{n_{\rm i}} + \mathtt{phin} \tag{4.10}$$

$$X_{\rm dep0} = \sqrt{\frac{2 \cdot \varepsilon_{\rm Si} \cdot \Psi_{\rm s}}{1.60219 \cdot 10^{-19} \cdot \mathtt{ndep} \cdot 10^{6}}} \tag{4.11}$$

$$l_{\rm t0} = \sqrt{\frac{\varepsilon_{\rm Si} \cdot \mathtt{toxe} \cdot X_{\rm dep0}}{\mathtt{epsrox} \cdot \varepsilon_0}} \tag{4.12}$$

$$\theta = \frac{1}{2 \cdot \left[\cosh\left(\frac{\mathtt{dsub} \cdot L_{\rm eff}}{l_{\rm t0}}\right) - 1\right]} \tag{4.13}$$

4.2.3 Example

In Fig. 4.1, one can see an example of a typical result of the above algorithm. Starting point were a BSIM modelcard and a PSP modelcard for some 65 nm technology, which were fully aligned for the reference device, i.e. for **SA** $=$ **SAREF** $= 0.4$ μm and **SB** $=$ **SBREF** $= 0.4$ μm. In particular, $I_{\rm d,sat}$ and $V_{\rm th,sat}$ were exactly matched between the two models. Starting from the BSIM stress parameters, the procedure described above was applied to yield the corresponding PSP stress parameters. Then, simulations were performed for both BSIM and PSP for a number of values of **SA** (and taking **SB** $=$ **SA** in all cases) and as a function of channel length L. The results are shown in the figure.

4.3 WPE model parameters

Conversion of WPE parameters from BSIM to PSP is fairly straightforward. In Table 4.3, the PSP parameter values are expressed in terms of BSIM parameter values.

Table 4.3: PSP parameter values for WPE model in terms of BSIM parameter values. For the BSIM parameters `k2we`, `lk2we`, `wk2we` and `pk2we` no PSP-equivalent exists. The value of scaling factor γ depends on the value of the BSIM parameter `binunit`, as given in Table 4.4.

PSP global	PSP binning	NMOS	PMOS
KVTHOWEO	**POKVTHOWE**	`kvth0we`	$-$`kvth0we`
KVTHOWEL	**PLKVTHOWE**	$\gamma \cdot$ `lkvth0we`	$-\gamma \cdot$ `lkvth0we`
KVTHOWEW	**PWKVTHOWE**	$\gamma \cdot$ `wkvth0we`	$-\gamma \cdot$ `wkvth0we`
KVTHOWELW	**PLWKVTHOWE**	$\gamma^2 \cdot$ `pkvth0we`	$-\gamma^2 \cdot$ `pkvth0we`
KUOWEO	**POKUOWE**	`ku0we`	`ku0we`
KUOWEL	**PLKUOWE**	$\gamma \cdot$ `lku0we`	$\gamma \cdot$ `lku0we`
KUOWEW	**PWKUOWE**	$\gamma \cdot$ `wku0we`	$\gamma \cdot$ `wku0we`
KUOWELW	**PLWKUOWE**	$\gamma^2 \cdot$ `pku0we`	$\gamma^2 \cdot$ `pku0we`
WEB	**WEB**	`web`	`web`
WEC	**WEC**	`wec`	`wec`
SCREF	**SCREF**	`scref`	`scref`

Table 4.4: The value of the scaling factor γ depends on the value of the BSIM parameter `binunit`.

`binunit`	γ
1 (default)	1
$\neq 1$	10^6

4.3.1 Example

In Fig. 4.2, one can see an example of a typical result of the above conversion scheme for WPE. Similar to the stress example earlier in this section, starting point were a BSIM modelcard and a PSP modelcard for some 65 nm technology, which were fully aligned for the model without WPE, i.e. for **SC** $= 0$. In particular, $I_{\rm d,sat}$ and $V_{\rm th,sat}$ were exactly matched between the two models. Starting from the BSIM stress parameters, the procedure described above was applied to yield the corresponding PSP WPE parameters. Then, simulations were performed for both BSIM and PSP for a number of values of **SC** and as a function of channel width W. The results are shown in the figure.

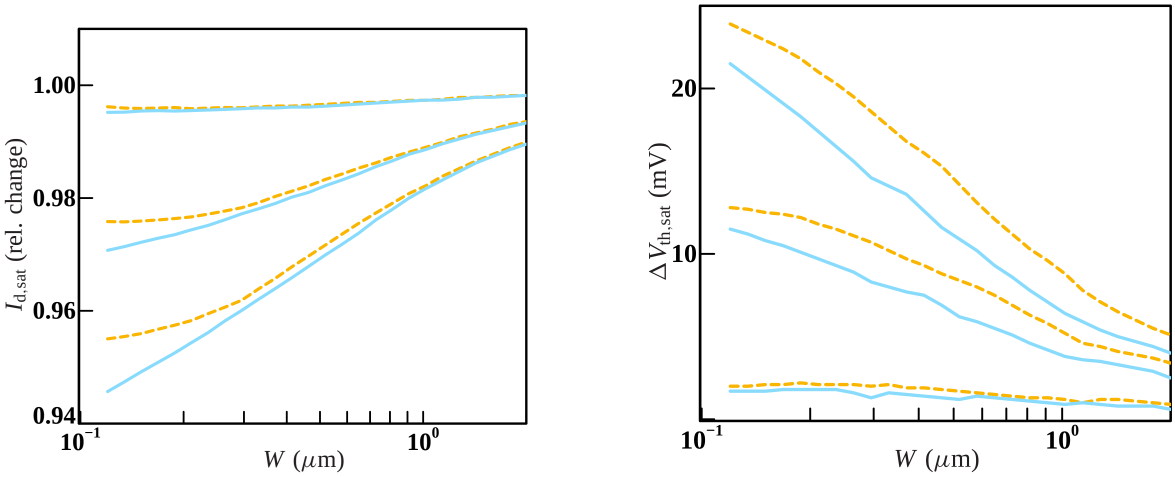

Figure 4.2: Relative change in $I_{\mathrm{d,sat}}$ (left) and shift in $V_{\mathrm{th,sat}}$ (right) as a function of W for **SC** $= 0.15, 0.3$ and 1 μm. Dashed lines are BSIM results, solid lines are PSP results. More details are given in the text. Noise in the $V_{\mathrm{th,sat}}$-curve is due to limited numerical accuracy.

References

[1] K.K. Hung *et al.*, IEEE Trans. El. Dev., Vol. 37, pp. 654–665 (1990); ibid., pp. 1323-1332 (1990).

[2] A.J. Scholten and D.B.M. Klaassen, *New 1/f noise model in MOS Model 9, level 903*, Nat.Lab. NL-UR 816/98, available on `http://www.nxp.com/models/mos_models/model9/`.

[3] `http://www-device.eecs.berkeley.edu/~bsim3/`

[4] G.D.J. Smit, R. van Langevelde, A.J. Scholten, D.B.M. Klaassen, G. Gildenblat, X. Li, H. Wang, and W. Wu, *PSP 102.3*, available on `http://pspmodel.asu.edu/`.