

1 Mextram 504 syntax

1.1 Introduction

The Modelbook version of the Mextram 504 model will no longer be supported. For those familiar with the Modelbook, the table below can help to find your way in the MextramDefinition document. The subjects (close to the paragraph titles) are placed next to the the paragraph numbers in both documents

Furthermore, in this document extra Mextram 504 documentation is found, such as the simulator specific information related to the SiMKit implementation of the model.

Table 1: Modelbook t504.pdf vs MextramDefinition_504.9_nxp.2_beta.pdf

Subject	Modelbook	MextramDef.
Model constants	1.3.1	4.5
Parameters and clipping	1.3.2	4.3
Temperature scaling	1.4.1	4.7
Mult scaling	1.4.2	4.6
Equivalent circuit	1.5	4.4
Model equations	1.5	2 and 4.8
Selfheating	1.6	4.14
DC Operating Point	1.7	4.18
Embedding	1.8.1	4.16
Implementation Issues	1.8.2	4.15
Parameter extraction	1.10	3

1.2 Simulator specific items

1.2.1 Pstar syntax

n channel model:	tn_n	(c, b, e) level=504, <parameters>
p channel model:	tp_n	(c, b, e) level=504, <parameters>
n channel self-heating model:	tnt_n	(c, b, e, dt) level=504, <parameters>
p channel self-heating model:	tpt_n	(c, b, e, dt) level=504, <parameters>
n channel substrate model:	tns_n	(c, b, e, s) level=504, <parameters>
p channel substrate model:	tps_n	(c, b, e, s) level=504, <parameters>
n channel substrate self-heating model:	tnst_n	(c, b, e, s, dt) level=504, <parameters>
p channel substrate self-heating model:	tpst_n	(c, b, e, s, dt) level=504, <parameters>
n	:	occurrence indicator
<parameters>	:	list of model parameters

c, b, e, s and dt are collector, base, emitter, substrate and self-heating terminals respectively.

1.2.2 Spectre syntax

n channel model:	model modelname bjtd504 type=npn ¹ <modpar> componentname c b e modelname <inpar>
p channel model:	model modelname bjtd504 type=ppn ² <modpar> componentname c b e modelname <inpar>
n channel self-heating model:	model modelname bjtd504t type=npn ¹ <modpar> componentname c b e dt modelname <inpar>
p channel self-heating model:	model modelname bjtd504t type=ppn ² <modpar> componentname c b e dt modelname <inpar>
n channel substrate model:	model modelname bjt504 type=npn ¹ <modpar> componentname c b e s modelname <inpar>
p channel substrate model:	model modelname bjt504 type=ppn ² <modpar> componentname c b e s modelname <inpar>
n channel substrate self-heating model:	

1.Either npn or npnv are interpreted as vertical npn (vertical only).

2.Either pnp or pnpv are interpreted as vertical pnp (vertical only).

model modelname bjt504t type=npn¹ <modpar>
 componentname c b e s dt modelname <inpar>

p channel substrate self-heating model:

model modelname bjt504t type=ppn² <modpar>
 componentname c b e s dt modelname <inpar>

modelname : name of model, user defined
 componentname : occurrence indicator
 <modpar> : list of model parameters³
 <inpar> : list of instance parameters³

c, b, e, s and dt are collector, base, emitter, substrate and self-heating terminals respectively.

1.2.3 ADS syntax

n channel model: model modelname bjtd504 gender=1 <modpar>
 componentname c b e modelname <instpar>

p channel model: model modelname bjtd504 gender=0<modpar>
 componentname c b e modelname <instpar>

n channel self-heating model: model modelname bjtd504t gender=1 <modpar>
 componentname c b e dt modelname <instpar>

p channel self-heating model: model modelname bjtd504t gender=0 <modpar>
 componentname c b e dt modelname <instpar>

n channel substrate model: model modelname bjt504 gender=1 <modpar>
 componentname c b e s modelname <instpar>

p channel substrate model: model modelname bjt504 gender=0 <modpar>
 componentname c b e s modelname <instpar>

n channel substrate self-heating model: model modelname bjt504t gender=1 <modpar>
 componentname c b e s dt modelname <instpar>

p channel substrate self-heating model: model modelname bjt504t gender=0 <modpar>
 componentname c b e s dt modelname <instpar>

modelname : name of model, user defined
 componentname : occurrence indicator

1.Either npn or npnv are interpreted as vertical npn (vertical only).

2.Either pnp or pnpv are interpreted as vertical pnp (vertical only).

3.For more details of these Spectre parameters see also Cadence Spectre Circuit Simulator Reference, version 5.0.

<modpar> : list of model parameters
 <instpar> : list of instance parameters

c, b, e, s and dt are collector, base, emitter, substrate and self-heating terminals respectively.

1.2.4 The ON/OFF condition for Pstar

The solution of a circuit involves a process of successive calculations. The calculations are started from a set of 'initial guesses' for the electrical quantities of the nonlinear elements. A simplified DCAPPROX mechanism for devices using ON/OFF keywords is mentioned in the Pstar User Manual. By default the devices start in the default state.

TN-level 504			
	Default	ON	OFF
V_{BC1}	-1.0	0.0	-1.0
V_{B1C1}	-1.0	0.0	-1.0
V_{B2C1}	-1.0	0.0	-1.0
V_{B2C2}	-1.0	0.0	-1.0
V_{B1E1}	0.65	0.75	-0.3
V_{B2E1}	0.65	0.75	-0.3
V_{B1B2}	1.0×10^{-6}	1.0×10^{-6}	0.0

TNT-level 504			
	Default	ON	OFF
DT	0.0	0.0	0.0

TNS-level 504			
	Default	ON	OFF
V_{BC1}	-1.0	0.0	-1.0
V_{B1C1}	-1.0	0.0	-1.0
V_{B2C1}	-1.0	0.0	-1.0
V_{B2C2}	-1.0	0.0	-1.0
V_{B1E1}	0.65	0.75	-0.3
V_{B2E1}	0.65	0.75	-0.3
V_{B1B2}	1.0×10^{-6}	1.0×10^{-6}	0.0
V_{SC1}	-5.0	-5.0	-5.0

TNST- level 504			
	Default	ON	OFF
DT	0.0	0.0	0.0

For the TP, TPS, TPT and TPST-level 504 the numbers remain the same but have a negative value, i.e. 1.0 becomes -1.0.

1.2.5 The ON/OFF condition for Spectre

n-channel					
	Default	OFF	Saturation	Reverse	Forward
V_{BC1}	-1.0	-1.0	0.75	0.75	0.0
V_{B1C1}	-1.0	-1.0	0.75	0.75	0.0
V_{B2C1}	-1.0	-1.0	0.75	0.75	0.0
V_{B2C2}	-1.0	-1.0	0.75	0.75	0.0
V_{B1E1}	0.65	0.65	0.75	0.0	0.75
V_{B2E1}	0.65	0.65	0.75	0.0	0.75
V_{B1B2}	10.e-6	10.e-6	0.0	10.e-6	10.e-6

n-channel with self-heating					
	Default	OFF	Saturation	Reverse	Forward
DT	0.0	0.0	0.0	0.0	0.0

n-channel with substrate					
	Default	OFF	Saturation	Reverse	Forward
V_{BC1}	-1.0	-1.0	0.75	0.75	0.0
V_{B1C1}	-1.0	-1.0	0.75	0.75	0.0
V_{B2C1}	-1.0	-1.0	0.75	0.75	0.0
V_{B2C2}	-1.0	-1.0	0.75	0.75	0.0
V_{B1E1}	0.65	0.65	0.75	0.0	0.75
V_{B2E1}	0.65	0.65	0.75	0.0	0.75
V_{B1E2}	10.e-6	10.e-6	0.0	10.e-6	10.e-6
V_{SC1}	-5.0	-5.0	-5.0	-5.0	-5.0
V_{B1B}	0.0	0.0	0.0	0.0	0.0
V_{C1C}	0.0	0.0	0.0	0.0	0.0
V_{E1E}	0.0	0.0	0.0	0.0	0.0

n-channel with substrate and self-heating					
	Default	OFF	Saturation	Reverse	Forward
<i>DT</i>	0.0	0.0	0.0	0.0	0.0

For the p-channel devices the numbers remain the same but have a negative value, i.e. 1.0 becomes -1.0.

1.2.6 The ON/OFF condition for ADS

n-channel	
	Default
V_{BC1}	0.0
V_{B1C1}	0.0
V_{B2C1}	0.0
V_{B2C2}	0.0
V_{B1E1}	0.0
V_{B2E1}	0.0
V_{B1B2}	0.0

n-channel with self-heating	
	Default
<i>DT</i>	0.0

n-channel with substrate	
	Default
V_{BC1}	0.0

V_{B1C1}	0.0
V_{B2C1}	0.0
V_{B2C2}	0.0
V_{B1E1}	0.0
V_{B2E1}	0.0
V_{B1E2}	0.0
V_{SC1}	0.0
V_{B1B}	0.0
V_{C1C}	0.0
V_{E1E}	0.0

n-channel with substrate and self-heating	
	Default
DT	0.0

For the p-channel devices the numbers remain the same but have a negative value, i.e. 1.0 becomes -1.0.

1.3 Example of Selfheating Usage

Below, an example (*Pstar*) is given to illustrate the working of self-heating.

□ Example

```
Title: example self-heating 504;

circuit;
  tnt_1(c, b, 0, dt) level=504,Rth=100,cth=1e-9;
  e_c (c, 0) 5;
  e_b (b, 0) 1;
end;

dc;
  print: vn(dt), op(pdiss.tnt_1);
end;
run;

result:

DC   Analysis.
VN(DT)                =      9.284E+00
Pdiss.TNT_1           =      92.840E-03
```

The voltage on node dT is $9.284E+00$ V, which means that the local temperature is increased by 9.284 °C.

A Hyp functions

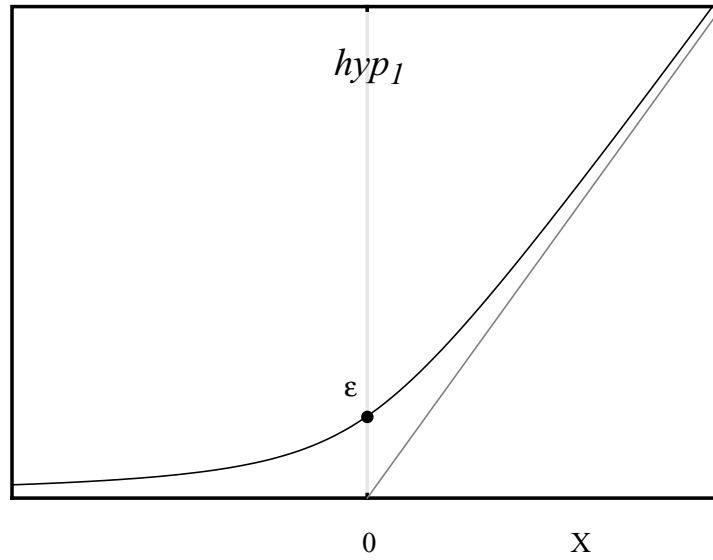


Figure 1: $hyp_1(x;\epsilon) = \frac{1}{2} \cdot (x + \sqrt{x^2 + 4 \cdot \epsilon^2})$

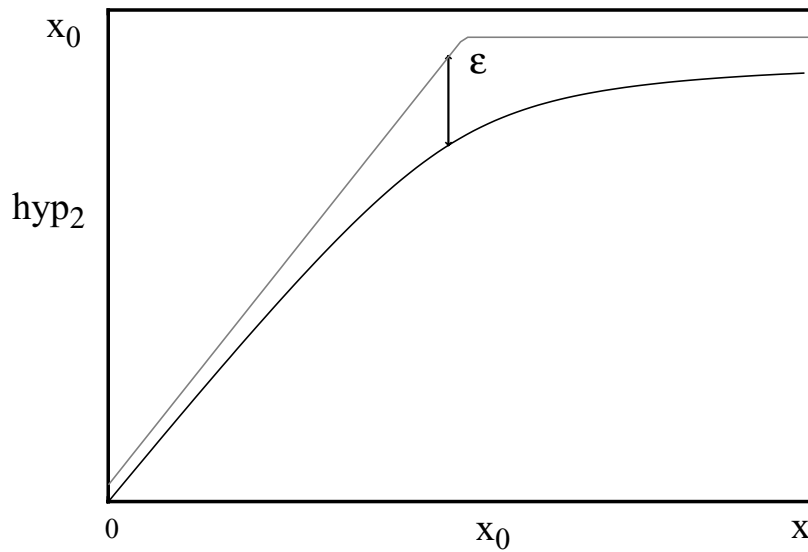


Figure 2: $hyp_2(x;x_0;\epsilon) = x - hyp_1(x - x_0;\epsilon)$

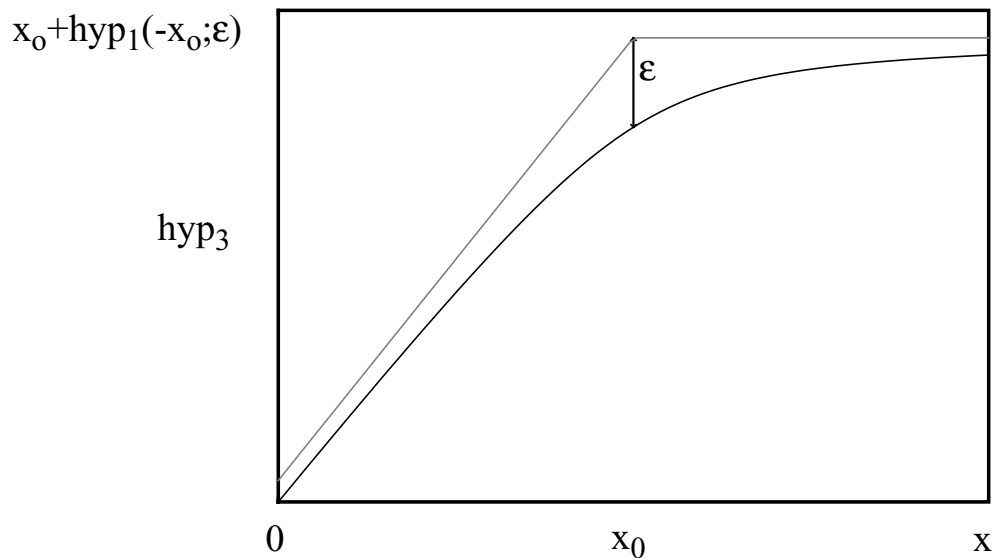


Figure 3: $hyp_3(x;x_0;\epsilon) = hyp_2(x;x_0;\epsilon) - hyp_2(0;x_0;\epsilon)$ for $\epsilon = \epsilon(x_0)$

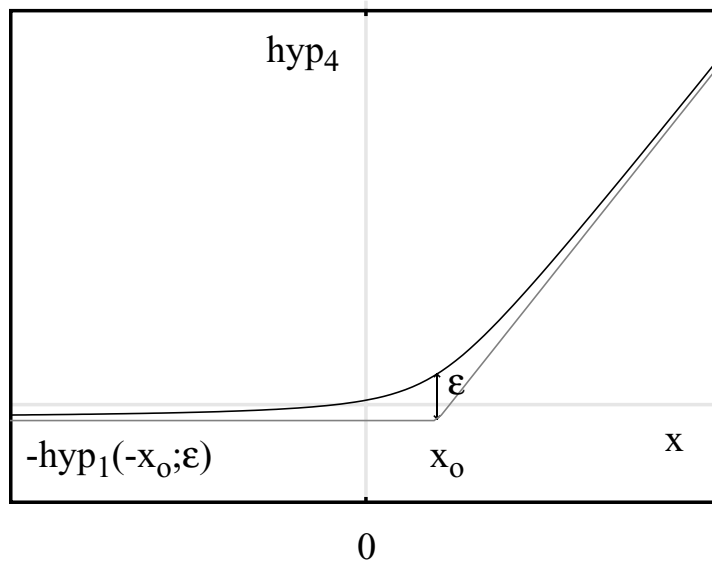


Figure 4: $hyp_4(x;x_0;\epsilon) = hyp_1(x - x_0;\epsilon) - hyp_1(-x_0;\epsilon)$

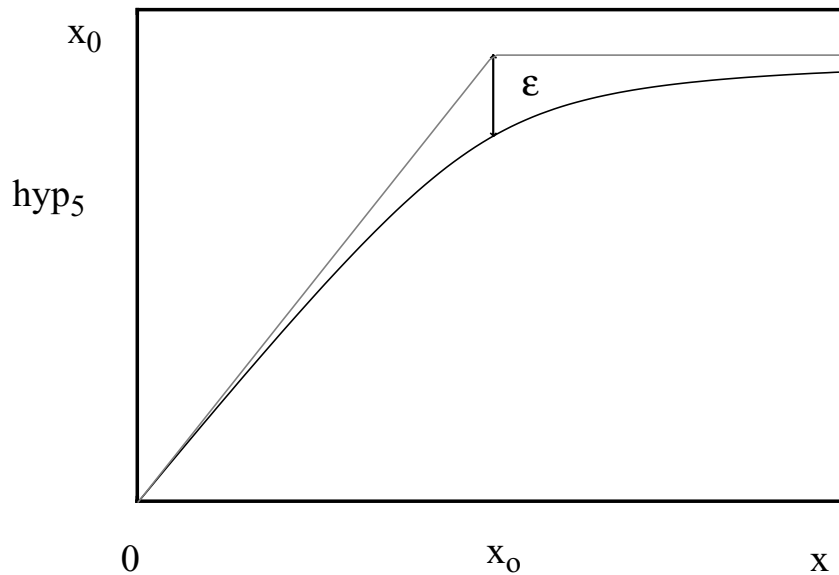


Figure 5: $hyp_5(x; x_0; \epsilon) = x_0 - hyp_1\left(x_0 - x - \frac{\epsilon^2}{x_0}, \epsilon\right)$ for $\epsilon = \epsilon(x_0)$

The hypm-function:

$$hypm[x, y; m] = \frac{x \cdot y}{(x^{2 \cdot m} + y^{2 \cdot m})^{1/(2 \cdot m)}} \quad (1.1)$$

B Spectre Specific Information

Imax, Imelt, Jmelt parameters

Introduction

Imax, Imelt and Jmelt are Spectre-specific parameters used to help convergence and to prevent numerical problems. We refer in this text only to the use of Imax model parameter in Spectre with SiMKit devices since the other two parameters, Imelt and Jmelt, are not part of the SiMKit code. For information on Imelt and Jmelt refer to Cadence documentation.

Imax model parameter

Imax is a model parameter present in the following SiMKit models:

- juncap and juncap2
- psp and pspnqs (since they contain juncap models)

In Mextram 504 (bjt504) and Modella (bjt500) SiMKit models, Imax is an internal parameter and its value is set through the adapter via the Spectre-specific parameter Imax.

In models that contain junctions, the junction current can be expressed as:

$$I = I_s \exp\left(\frac{V}{N \cdot \phi_{TD}} - 1\right) \quad (1.2)$$

The exponential formula is used until the junction current reaches a maximum (explosion) current Imax.

$$I_{max} = I_s \exp\left(\frac{V_{expl}}{N \cdot \phi_{TD}} - 1\right) \quad (1.3)$$

The corresponding voltage for which this happens is called Vexpl (explosion voltage). The voltage explosion expression can be derived from (1):

$$V_{expl} = N \cdot \phi_{TD} \log\left(\frac{I_{max}}{I_s}\right) + 1 \quad (1.4)$$

For $V > V_{expl}$ the following linear expression is used for the junction current:

$$I = I_{max} + (V - V_{expl}) \frac{I_s}{N \cdot \phi_{TD}} \exp\left(\frac{V_{expl}}{N \cdot \phi_{TD}}\right) \quad (1.5)$$

The default value of the I_{max} model parameter for SiMKit is 1000A. The default value of I_{max} for Mextram 504 and Modella is 1A. I_{max} should be set to a value which is large enough so it does not affect the extraction procedure.

Region parameter

Region is an Spectre-specific model parameter used as a convergence aid and gives an estimated DC operating region. The possible values of region depend on the model:

- For Bipolar models:
 - subth: Cut-off or sub-threshold mode
 - fwd: Forward
 - rev: Reverse
 - sat: Saturation.
 - off¹
 -
- For MOS models:
 - subth: Cut-off or sub-threshold mode;
 - triode: Triode or linear region;
 - sat: Saturation
 - off¹

For PSP and PSPNQS all regions are allowed, as the PSP(NQS) models both have a MOS part and a juncap (diode). Not all regions are valid for each part, but when e.g. region=forward is set, the initial guesses for the MOS will be set to zero. The same holds for setting a region that is not valid for the JUNCAP.

- For diode models:
 - fwd: Forward
 - rev: Reverse
 - brk: Breakdown
 - off¹

¹.Off is not an electrical region, it just states that the user does not know in what state the device is operating

Model parameters for device reference temperature in Spectre

This text describes the use of the `tnom`, `tref` and `tr` model parameters in Spectre with SiMKit devices to set the device reference temperature.

A Simkit device in Spectre has three model parameter aliases for the model reference temperature, `tnom`, `tref` and `tr`. These three parameters can only be used in a model definition, not as instance parameters.

There is no difference in setting `tnom`, `tref` or `tr`. All three parameters have exactly the same effect. The following three lines are therefore completely equivalent:

```
model nmos11020 mos11020 type=n tnom=30
model nmos11020 mos11020 type=n tref=30
model nmos11020 mos11020 type=n tr=30
```

All three lines set the reference temperature for the `mos11020` device to 30 C.

Specifying combinations of `tnom`, `tref` and `tr` in the model definition has no use, only the value of the last parameter in the model definition will be used. E.g.:

```
model nmos11020 mos11020 type=n tnom=30 tref=34
```

will result in the reference temperature for the `mos11020` device being set to 34 C, `tnom=30` will be overridden by `tref=34` which comes after it.

When there is no reference temperature set in the model definition (so no `tnom`, `tref` or `tr` is set), the reference temperature of the model will be set to the value of `tnom` in the options statement in the Spectre input file. So setting:

```
options1 options tnom=23 gmin=1e-15 reltol=1e-12 \
  vabstol=1e-12 iabstol=1e-16
model nmos11020 mos11020 type=n
```

will set the reference temperature of the `mos11020` device to 23 C.

When no `tnom` is specified in the options statement and no reference temperature is set in the model definition, the default reference temperature is set to 27 C.

So the lines:

```
options1 options gmin=1e-15 reltol=1e-12 vabstol=1e-12 \
  iabstol=1e-16
model nmos11020 mos11020 type=n
```

will set the reference temperature of the mos11020 device to 27 C.

The default reference temperature set in the SiMKit device itself is in the Spectre simulator never used. It will always be overwritten by either the default "options tnom", an explicitly set option tnom or by a tnom, tref or tr parameter in the model definition.

