Vorlesung 12 (11.12.2023) Simulation von Temperatureinflüssen

Befehl .TEMP 91 92 93 ... Beispiel Widerstand RW=RK(1+ $\alpha * \Delta 9$) α = TC1 ... Temperaturkoeffizient

1) <u>Übung (Temperatureinfluß bei Dioden : Verzeichnis Diode Temperatur)</u>



Notwendige Parametereinstellungen der Bauteile (s. Buch Seite 45)

b) D1N4002	.MODEL D1N400)2 D (IS=14.11E-	9 N=1.984	RS=33.89E-3	TRS1=7e-3
+ IKF=94.81	XTI=3 EG=1.110	CJO=51.17E-12	M=.2762	VJ=.3905 FC=	=.5
+ ISR = 100.0I	E-12 NR=2 BV=10	0.1 IBV=10 TT	=4.761E-6)		

TRS1	=	linearer Temperaturkoeffizient für den Reihenwiderstand (RS)
TBV1	=	linearer Temperaturkoeffizient für die Durchbruchspannung
RS	=	Reihenwiderstand

T_MEASURED=Die Temperatur für die die Daten gelten !

Der Wert der Durchlaßspannung kann durch den Parameter "N" = Emissions-Koeffizient verändert werden.

Um sowohl die Spannung zu variieren als auch mit 2 verschiedenen Temperaturen zu simulieren sind folgende Einstellungen vorzunehmen

a) primary sweep

b) secondary sweep

Si	mulation Settings - Temp1			×	Simulation S	ettings - Temp	1		×
lſ	General Analysis Include F	iles Libraries Stimulus Op	otions Data Collection	Probe Window	General	nalysis Include	Files Libraries Stimulus	Options Data Collection	Probe Window
-	Analysis type: DC Sweep Options: Secondary Sweep Monte Catlo/Worst Case Parametric Sweep Temperature (Sweep) Save Bias Point Load Bias Point	Sweep variable © Voltage source © Giobal parameter © Model parameter © Iemperature Sweep type © Linear © Logarithmic Decade © Value ligt	Name: V1 Model type: Model name: Paremeter name: Paremeter name: End value: End value: Increment	1 10 0.01	Analysis ty DC Sweet Options: ♥ Primary ♥ Secont ● Parame ● Tempet ● Save B ● Load B	pe: Sweep Jay Sweep Carlo/Worst Case tric Sweep ature (Sweep) ature (Sweep) as Point as Point	Sweep variable	Name: Model type: Model name: Earemeter name: Start value: ade Y End value: Increment: 50	
		OK Abb	Direchen Übernehma	Hilfe			ОК	Abbrechen Ü <u>b</u> ernehm	en Hilfe



Merke: Bei kleinem Strom sinkt die Durchlaßspannung mit ca. 1mV/K ; bei großem Strom steigt die Durchlaßspannung mit der Temperatur





→ ändert man TC1 von -600E-6 auf +600E-6, so erhält man ein PTC-Verhalten !

TC2 ergänzen z.B.: TC2=+20E-12 (sehr wenig Einfluß) bzw. TC2=20E-6 (sehr deutlicher Einfluss) TC1= linearer Temperaturkoefficient ; TC2 = quadratischer Temperaturkoefficient

mulation Settings - Rtes		
General Analysis Include	Files Libraries Stimulus Options Data Col	llection Probe Window
Analysis type: DC Sweep	Sweep variable Voltage source Name: Current source Gibbal parameter Model type: Model reme:	<u> </u>
Secondary Sweep Monte Carlo/Worst Case Parametric Sweep Temperature (Sweep) Save Bias Point Load Bias Point	<u>Temperature</u> <u>Earameter name</u> Sweep type <u>Start val</u> <u>End val</u>	e: -55
	C Loganthmic Decade right	nt: 1
	OK Abbrechen 05	ernehmen Hilfe

3. Übung NTC – Nachbildung Seite 76ff

Verzeichnis: ../Beispiele/NTCTest <u>Schaltbild</u>







File Edit View Model Plot Iools Window Help Image: State of the state
Model Lift X Image: Construction of the construct
Model List x Model Name Type nkc* SUBCKT SUBCKT subckt ncc 1 2 For x SUBCKT ref 4 0 10K gout 0 5 poly(2) (5,0) (4,0) 0 0 0 0 1.0 ref 4 0 10K gout 0 5 poly(1) 6 0 3.266 + -0.16633619 + 0.0046450693 + 8.6656955e-5 + 1.017213e-6 + -3.866803e-9 + -8.8615615e-11 + 1.678045e-12 + -1.3013017e-14
+ 4.851/U31e-17 + -6.8866237e-20 r0 5 0 1.0 itemp 0 6 dc 1.0 rt 6 0 rtemp 0.001 .model rtemp res (r=1 tc1=1000)

PTC aus dem UP erzeugen :



Übung : Erstellen eines PTC als neues Modell.

a) Modelldefinition des NTC einfach kopieren und den Namen ändern \rightarrow (NTC nach PTC bis .subckt und .end...)

Schritt 1 : Verändern des Wertes für tc1 von +1000 auf -1000

.subckt ptc 1 2 ; Namen ändern : statt ntc ptc verwenden ! EOUT 1 3 poly(2) (5,0) (4,0) 0 0 0 0 1.0 vsense 3 2 dc 0.0 fout 0 4 vsense 1.0 rref 4 0 10K gout 0 5 poly(1) 6 0 3.266 +-0.16633619 +0.0046450693+-8.6856965e-5+ 1.017213e-6 +-3.8668603e-9 + -8.8615615e-11 +1.678045e-12+-1.3013017e-14 + 4.8617031e-17 +-6.8866237e-20 r0 5 0 1.0 itemp 0 6 dc 1.0 rt 6 0 rtemp 0.001 .model rtemp res (r=1 tc1=-1000) bei dem NTC ist dieser Wert tc1=+1000 !!! .ends ptc (Achtung auch hier ist statt ntc ptc einzutragen !)

Ergebnis nach der Veränderung von tc1 :







```
.options tnom=0
.subckt ptc2 1 2
EOUT 1 3 poly(2) (5,0) (4,0) 0 0 0 0 1.0
vsense 2 3 dc 0.0
fout 0 4 vsense 1.0
rref 4 0 10K
gout 0 5 poly(1) 6 0 -100 (bei dem NTC war statt der 100 hier 3.266 eingetragen)
+-0.16633619
+0.0046450693
+ -8.6856965e-5
+ 1.017213e-6
+-3.8668603e-9
+-8.8615615e-11
+1.678045e-12
+-1.3013017e-14
+ 4.8617031e-17
+ -6.8866237e-20
r0 5 0 1.0
itemp 0 6 dc 1.0
rt 6 0 rtemp 0.001
.model rtemp res (r=1 tc1=-1000)
.ends ptc2
```





Worst Case Analyse

Am Beispiel eines einfachen ohmschen Spannungsteilers R1/R2 wird die Worst-Case-Analyse mit dem Simulationsprogramm "Worst" gezeigt. In der Modellanweisung für die Widerstände ist eine Toleranz von +-10% vorgegeben. Berechnet wird die Teilspannung über dem Widerstand R2 am Knoten 2.



Definition der Widerstände:

.model rmod RES (r=1 dev=10%) ; r ist der Widerstandsfaktor Der Widerstand "RMOD" stammt aus der Laboreigenen Bibliothek "LEK.LIB"



Die grüne Kurve zeigt die Nenndaten; die andere Kurve zeigt das Verhalten. Welche Abweichung dargestellt werden soll ist in den oben dargestellten Simulationsoptionen einzustellen.

z.B.: Worst case direction: Hi oder Lo