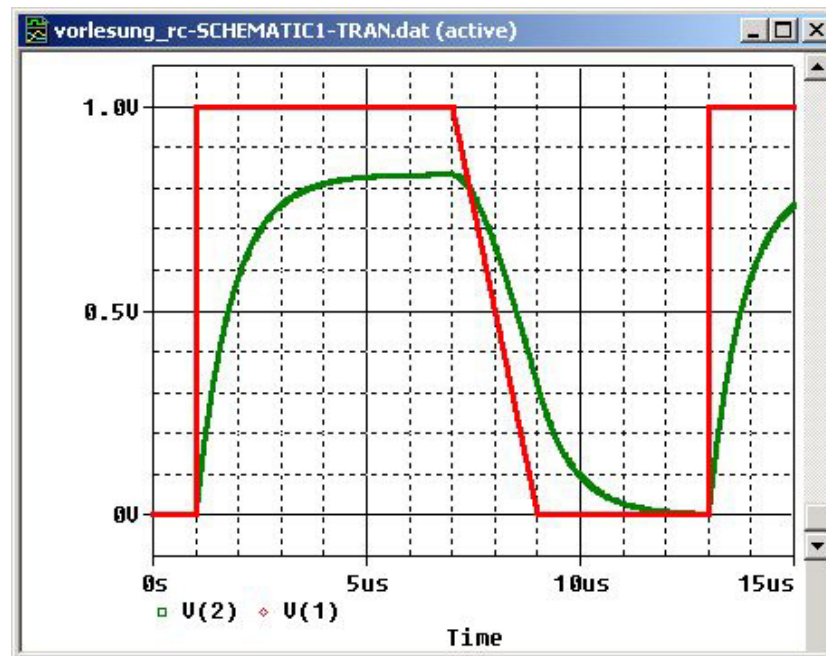

Analoge Simulation



Analoge Schaltungssimulation

Was ist das ?

- Ein Analoger Simulator arbeitet mit **kontinuierlichen Spannungen und Strömen**
- Er löst **numerisch** die Differential/Integralgleichungen, die die Bauteile beschreiben.
- Also: Analoge Simulation = **Kirchhoff'sche Regeln + DGL + Modelle**
- **WICHTIG:** Die Simulation ist nur so gut, wie die Modelle für die Bauelemente

Wozu ?

- Simulation analoger Schaltungen (Verstärker, Filter, Sense-Verstärker für RAMs, IO Pads, ...)
- **Genaues Verständnis** der Vorgänge in der Schaltung (z.B. Ladungsinjektion)
- Erzeugung der **Modelle für die digitale Simulation** (Durchlaufzeit, Anstiegszeit,...)
- Untersuchung der **Abhängigkeit** von Temperatur, Versorgungsspannung, Prozeßparametern, Device Matching, etc.
- Berücksichtigung der **parasitären Elemente** des Layouts (Kapazitäten, Leitungswiderstände)
- Abschätzung des Leistungsverbrauchs

Oft ist nur ein kleiner Teil eines Chips 'analog'

Aber: analoge Blöcke sind kritische Komponenten für die Gesamtperformance

Analoge Schaltungssimulation

Womit ?

- Unser Tool: **SPICE** (**S**imulation **P**rogram with **IC** **E**mphasis)
- Modelliert Widerstände, Kondensatoren, Dioden, bipolare Transistoren, MOSFETs, JFETs u.v.m.
- Viele verschiedene Analysemöglichkeiten (DC, AC, Transient, Rauschen,...)
- Eingabe ist eine **ASCII Datei**
- Heute erfolgen **Eingabe und Ausgabe oft grafisch** mit verschiedenen Tools

Schematic Editor ⇒ **Netzliste** ⇒ SPICE simuliert ⇒ Output File ⇒ Anzeige mit Viewer ⇒ Datenanalyse

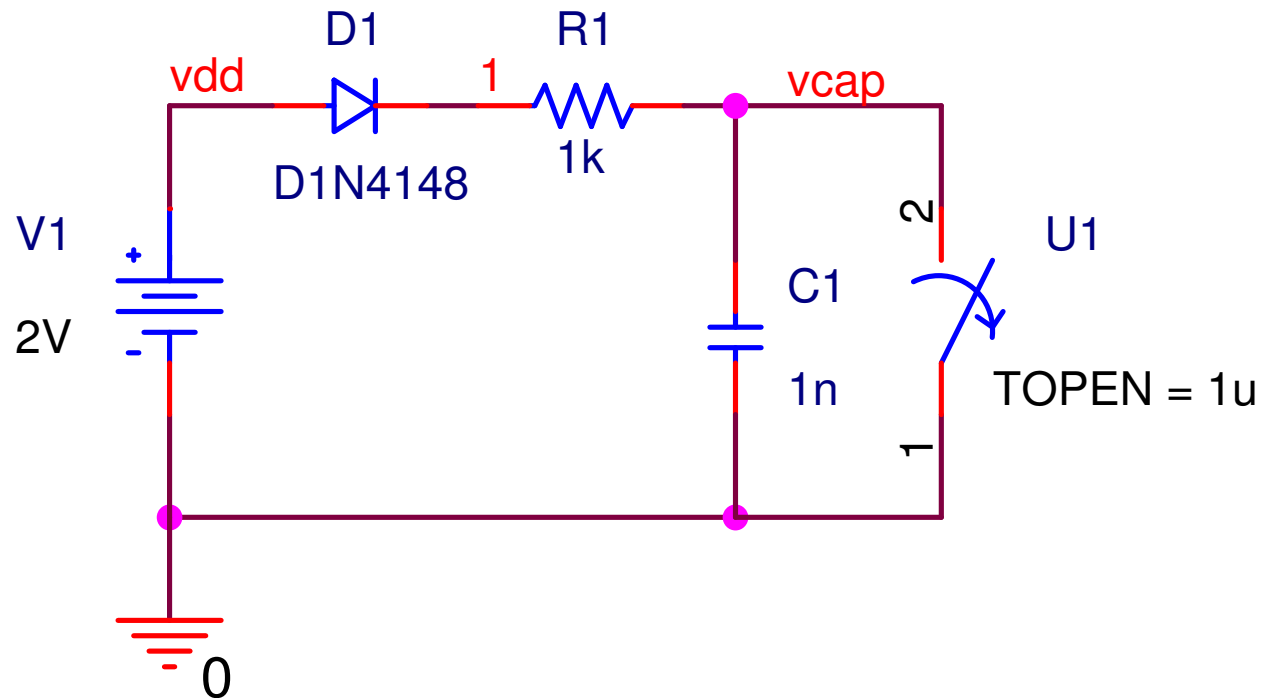
Warum müssen wir auf Netzlist-Level 'runter'?

- Wir müssen verstehen, was genau passiert, da alle graphischen Tools letztlich solch ein File generieren. Wenn etwas nicht klappt, muß man hier nachschauen!
- Mit Scripts lassen sich Simulationen automatisieren, z.B. zur Charakterisierung einer ganzen Bibliothek.
- Für kleine Designs ist die direkte Steuerung und Extraktion von Ergebnissen 'einfacher' und schneller ('Latex ⇔ Word'), daher manchmal

Editieren der Netzliste ⇒ SPICE simuliert ⇒ Messergebnisse
⇒ Output File ⇒ Anzeige mit Viewer

Motivation

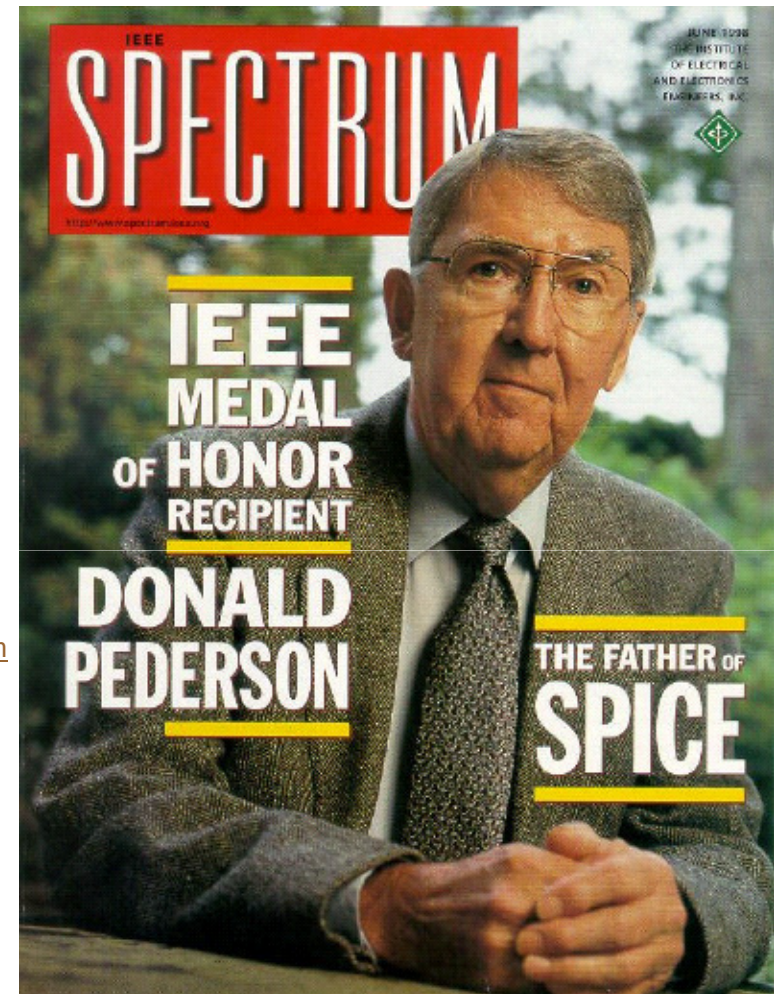
- Wie ändert sich v_{cap} in dieser Schaltung, wenn der Schalter geöffnet wird?
Genauer: Wie ist die Anstiegszeit, z.B. auf 90% des Endwertes? (Wie ist der Endwert?)



- Das ist schwierig zu berechnen, denn der Ladestrom ändert sich nicht-linear durch die Kennlinie der Diode $I_{diode} = I_0 [\exp(U_D/U_{th}) - 1]$.

Kurze Geschichte von SPICE

- 1968 Erste Entwicklungen an der UC Berkeley u. bei IBM
 - 1971 SPICE1 **Public Domain Tool** (Prof. D. Pederson)
 - 1975 SPICE2
 - 1993 SPICE3f Umstieg von FORTRAN auf C
 - 1997 SPICE3f5 Neueste Version aus Berkeley
-
- Wegen der freien Verfügbarkeit ist SPICE die Basis für viele kommerzielle Tools
 - Professionelles Tool: **HSPICE** (MetaSoftware, Avant)
 - Weiterhin: **ELDO** (Mentor, 1990)
 - 3-10x schneller
 - Verarbeitet auch sehr große Netzlisten (300.000 Transistoren)
 - Evaluation Version unter http://www.mentor.com/products/ic_nanometer_design/simulation/eldo/eval.cfm
 - Außerdem **SPECTRE** (von CADENCE)
 - **PSPICE** (Personal Computer SPICE) von MicroSim war/ist sehr verbreitet:
 - Kostenlose Demo-Version mit guter Funktionalität
 - Graphischer Post-Processor 'PROBE'
 - PSPICE wurde dann von ORCAD und dann (vor ca. 1 Jahr) von **CADENCE** übernommen. **Demo (auch CD)** verfügbar unter <http://www.orcad.com/download.orcaddemo.aspx>

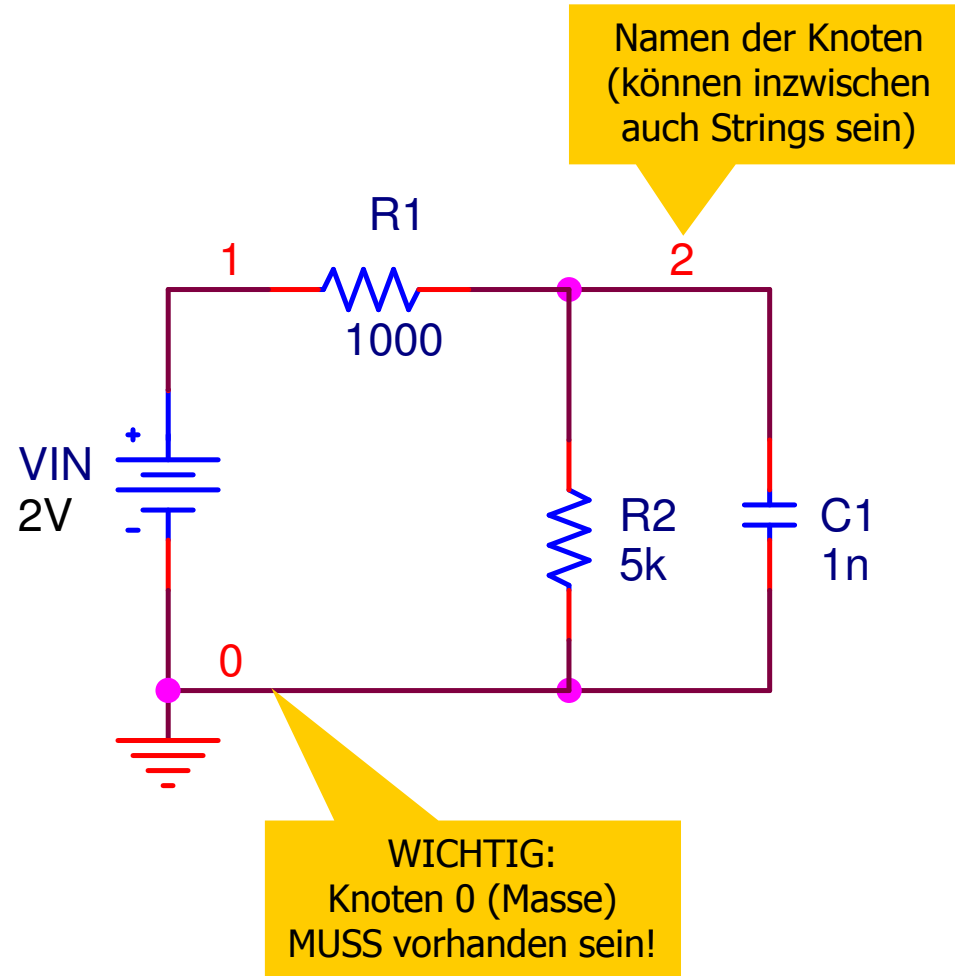


Ein erstes Beispiel

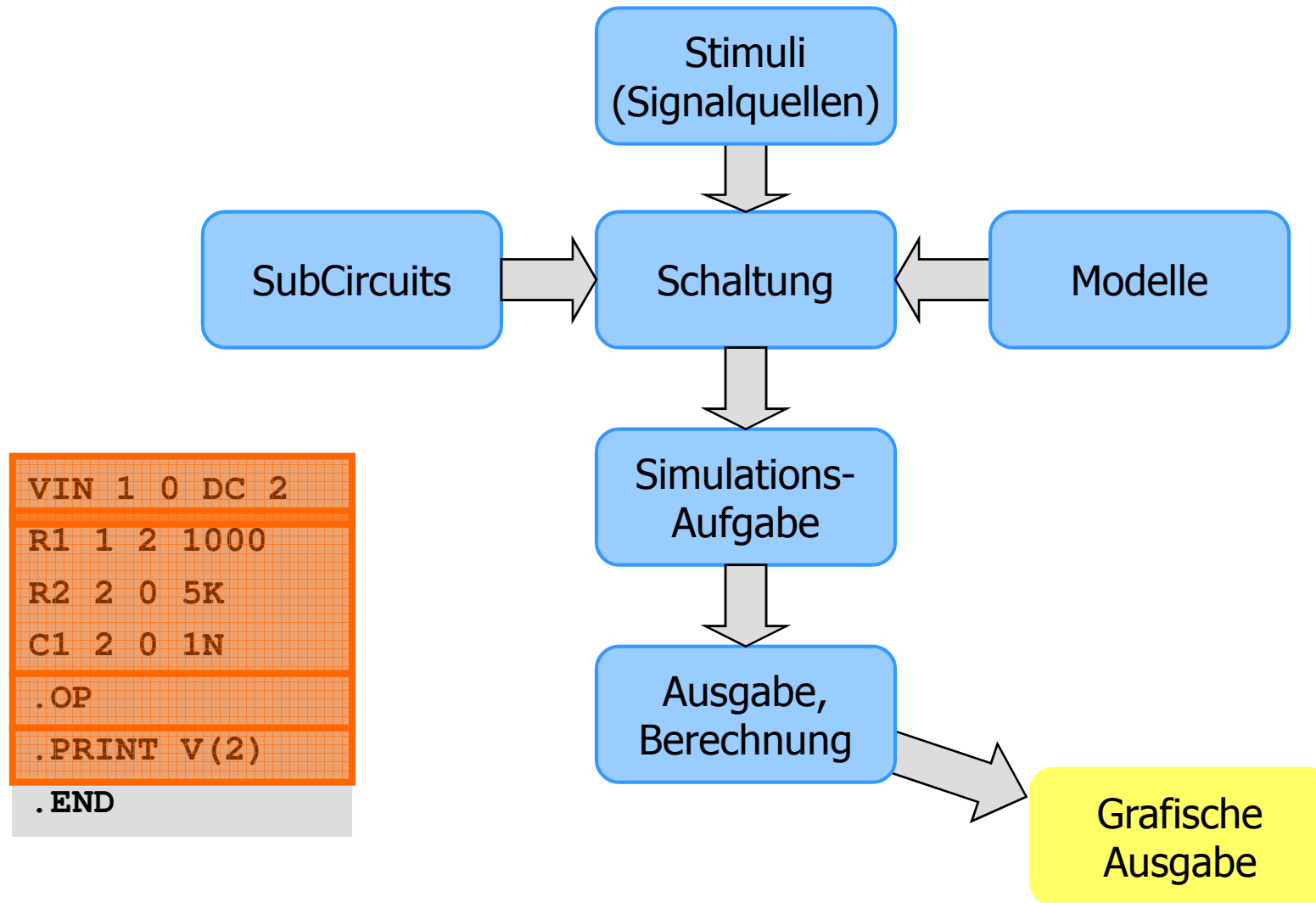
Demo: RC_glied.opj

SPICE Datei:

```
* Spannungsteiler  
VIN 1 0 DC 2  
R1 1 2 1000  
R2 2 0 5K  
C1 2 0 1N  
.OP  
.PRINT V(2)  
.END
```

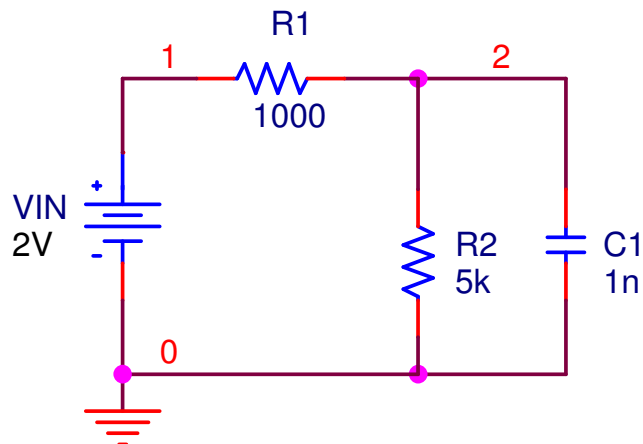


Grundelemente von SPICE

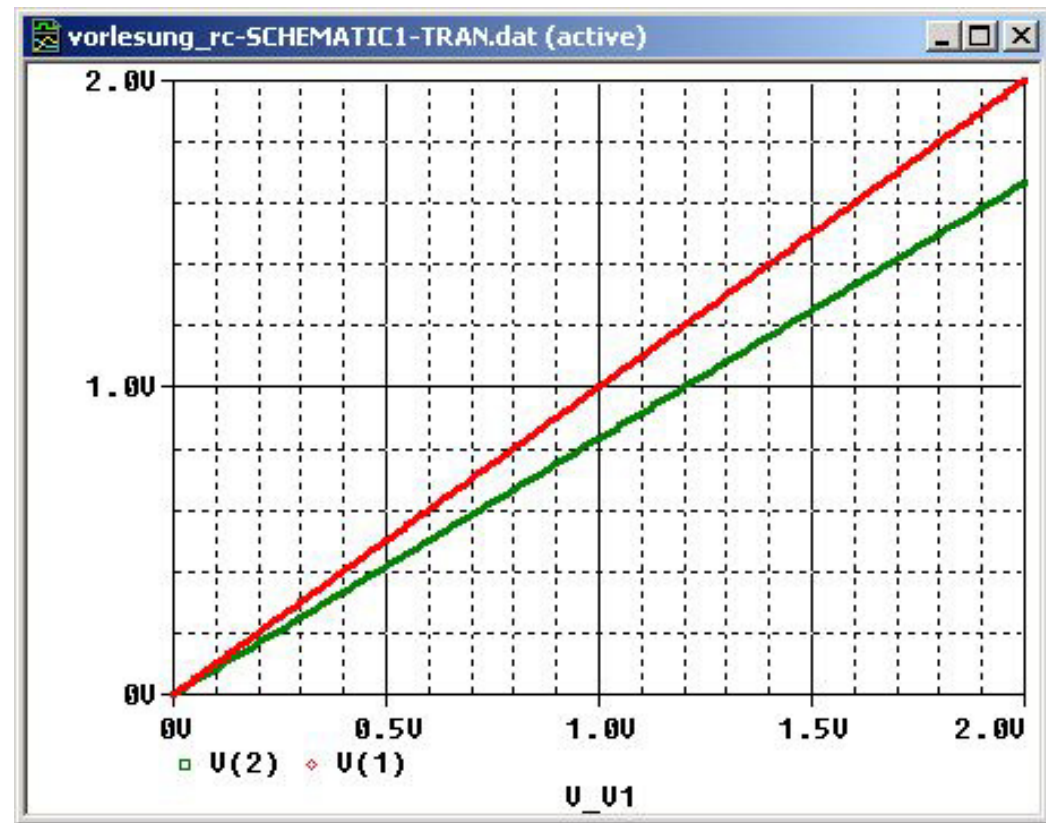


DC Analyse

- Statische Großsignalanalyse
- Alle **Kapazitäten werden** aus der Schaltung **entfernt**, Spulen auf den Ohm'schen Widerstand reduziert.
- Der Arbeitspunkt wird durch ein iteratives Verfahren berechnet (.OP Analyse)
- Die Spannung (der Strom) einer statischen Quelle wird variiert
- Nützlich zur Berechnung von **Arbeitspunkten**, Transferverhalten etc.



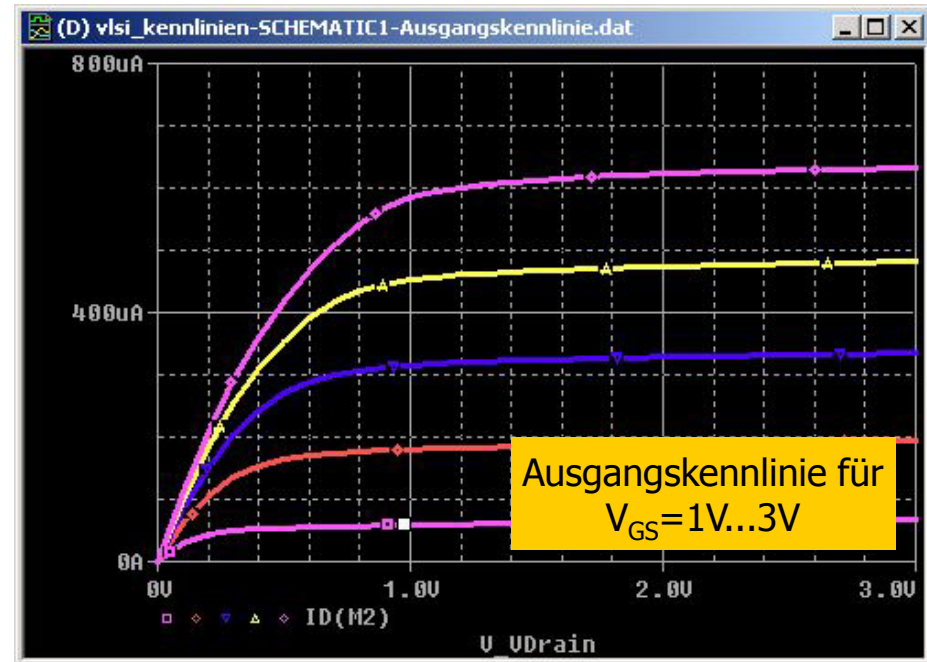
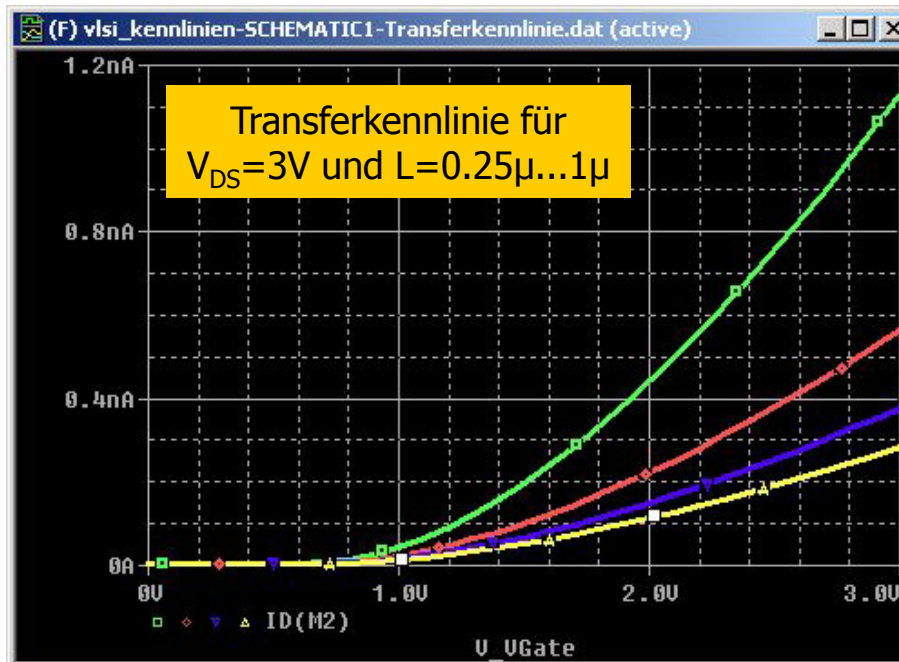
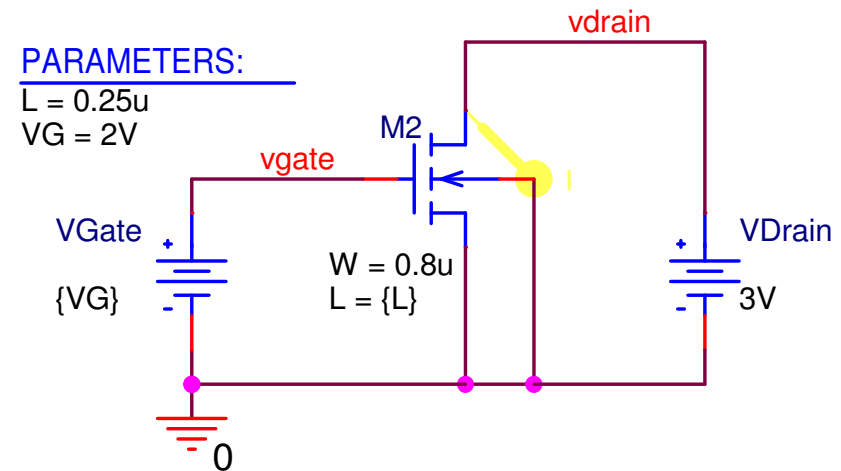
```
VIN 1 0 DC 2
R1 1 2 1000
R2 2 0 5K
C1 2 0 1N
.DC LIN VIN 0 2 0.01
.END
```



DC-Analyse: Kennlinien

Demo: Kennlinien.opj

- Mit einer DC-Analyse kann man die statischen Kennlinien der Transistoren verifizieren

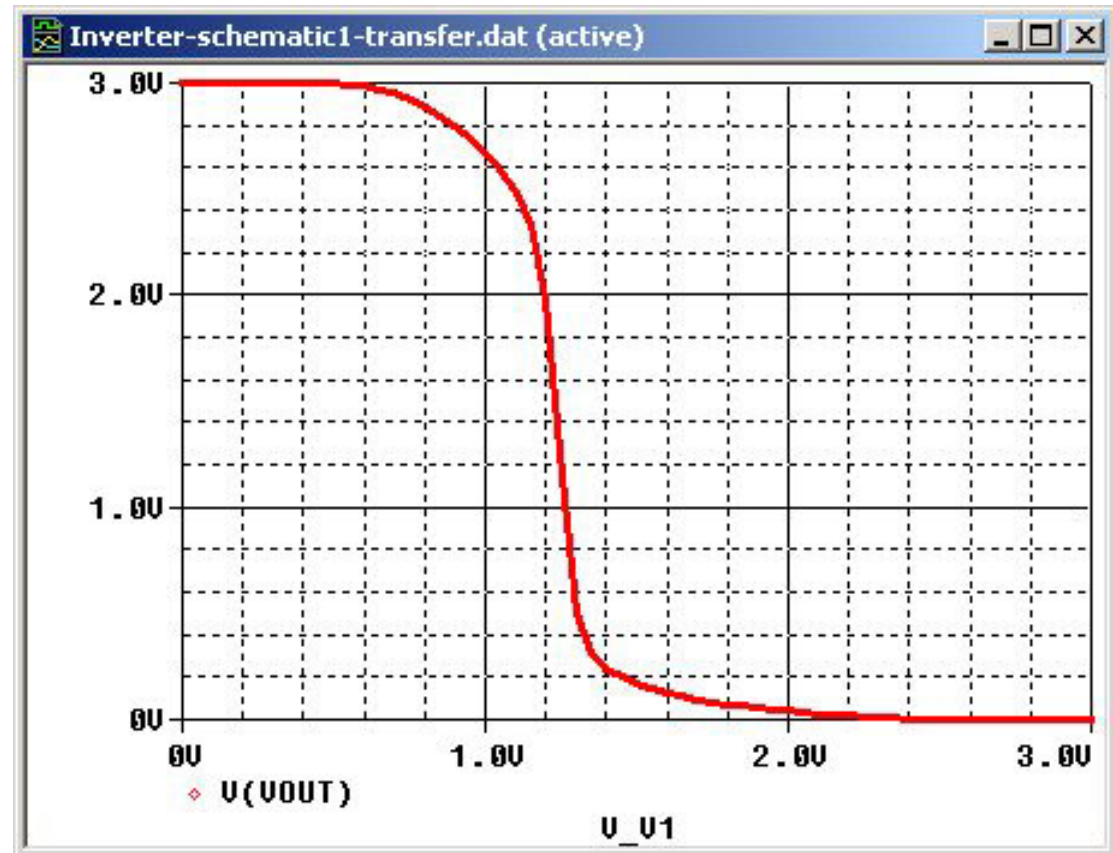
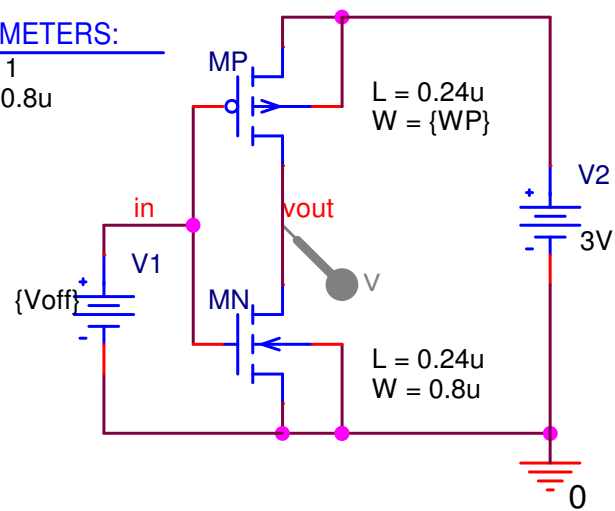


DC-Analyse: CMOS Inverter

Demo: CMOS_inverter.opj
Profile DC_transfer

PARAMETERS:

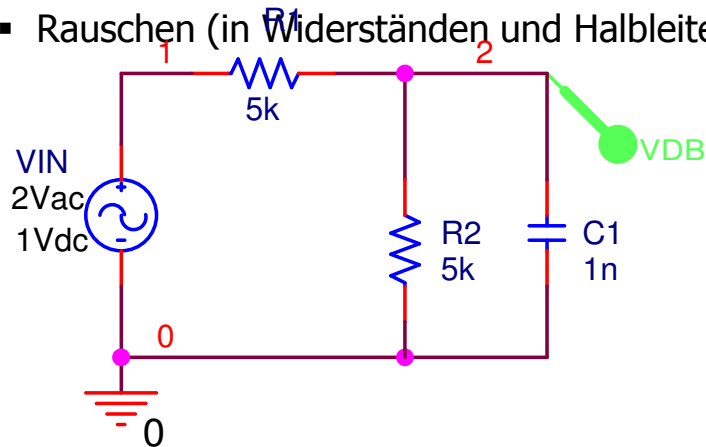
Voff = 1
WP = 0.8u



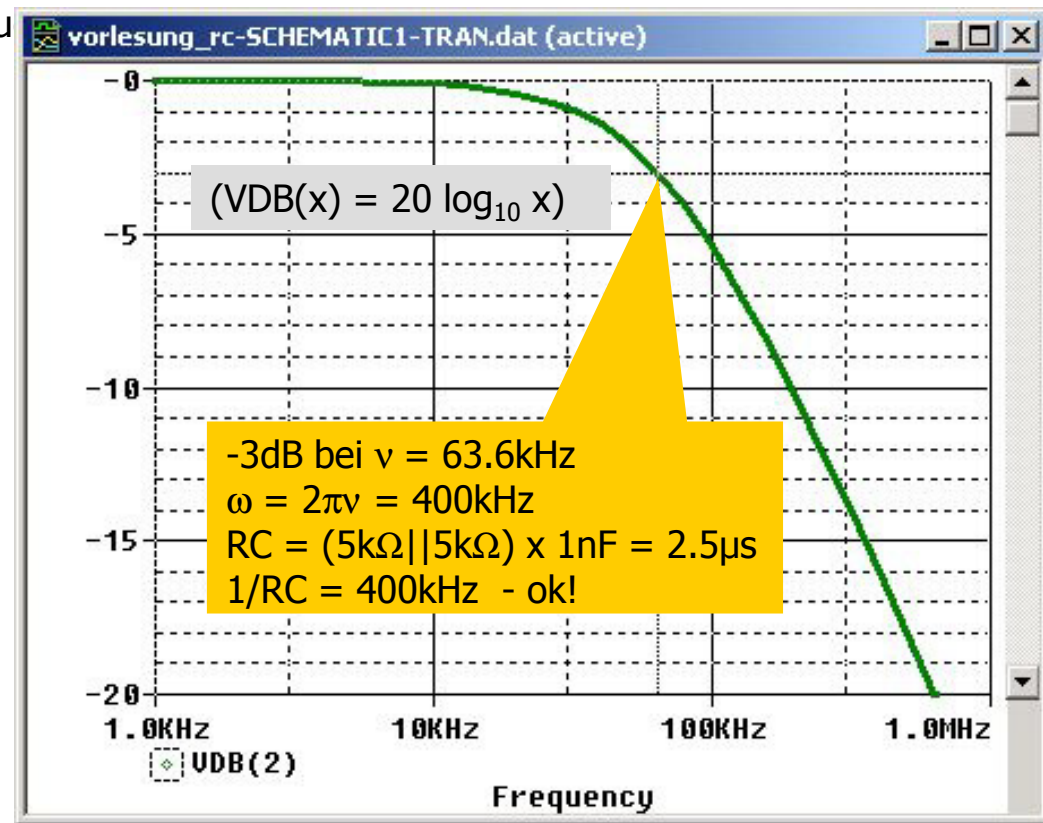
AC Analyse

Demo: Vorlesung_RC.opj

- Kleinsignal-Analyse (im Frequenzbereich). Variante davon ist Berechnung der Transfer Funktion (.TF)
- SPICE berechnet zunächst den **Arbeitspunkt** (.OP) und dadurch die **linearisierten Kleinsignalparameter** der Komponenten (Dioden, Transistoren...).
- Die Netzliste besteht dann nur noch aus Rs, Cs, Ls und Quellen. Sie wird als Matrix geschrieben und 'exakt' gelöst.
- Dann wird die Frequenz der AC-Signalquelle durchgefahen.
- Rauschen (in Widerständen und Halbleiterbau



```
VIN 1 0 DC 1 AC 2
R1 1 2 5k
R2 2 0 5K
C1 2 0 1N
.AC DEC 10 1k 1MEG
.END
```



AC-Analyse: CMOS Inverter

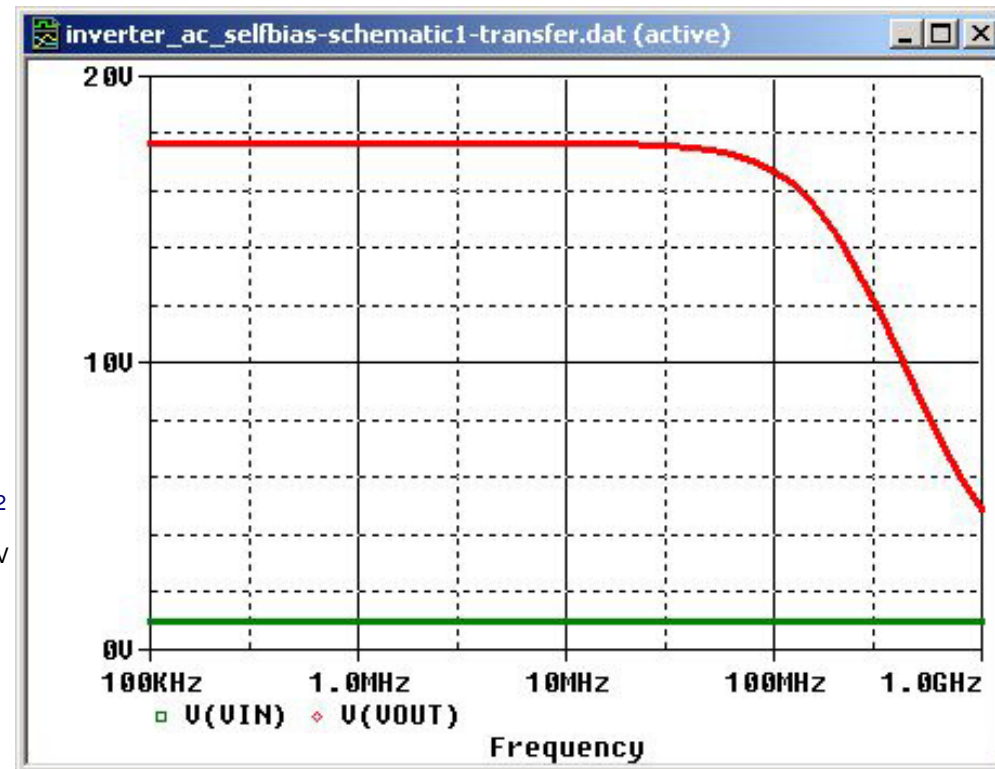
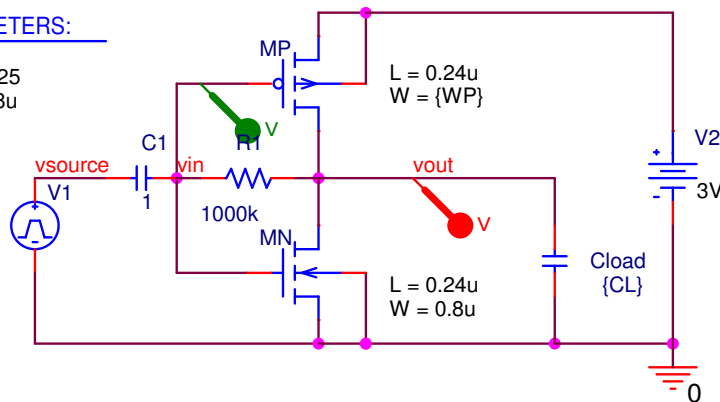
Demo: CMOS_Inverter_selfbias.opj

- Problem hier: DC – Wert des Eingangs ('Arbeitspunkt') – s. Demo!
- Lösung z.B. 'Self-Bias' – s. Demo

PARAMETERS:

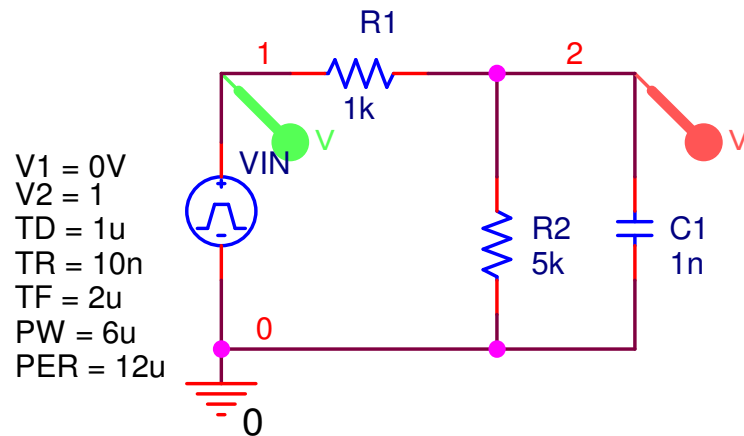
$CL = 10f$
 $V_{off} = 1.25$
 $WP = 0.8u$

$AC = 1$
 $V1 = 1$
 $V2 = 0$
 $TD = 0.5n$
 $TR = 0.1n$
 $TF = 1n$
 $PW = 10n$
 $PER = 20n$
 $DC = \{V_{off}\}$

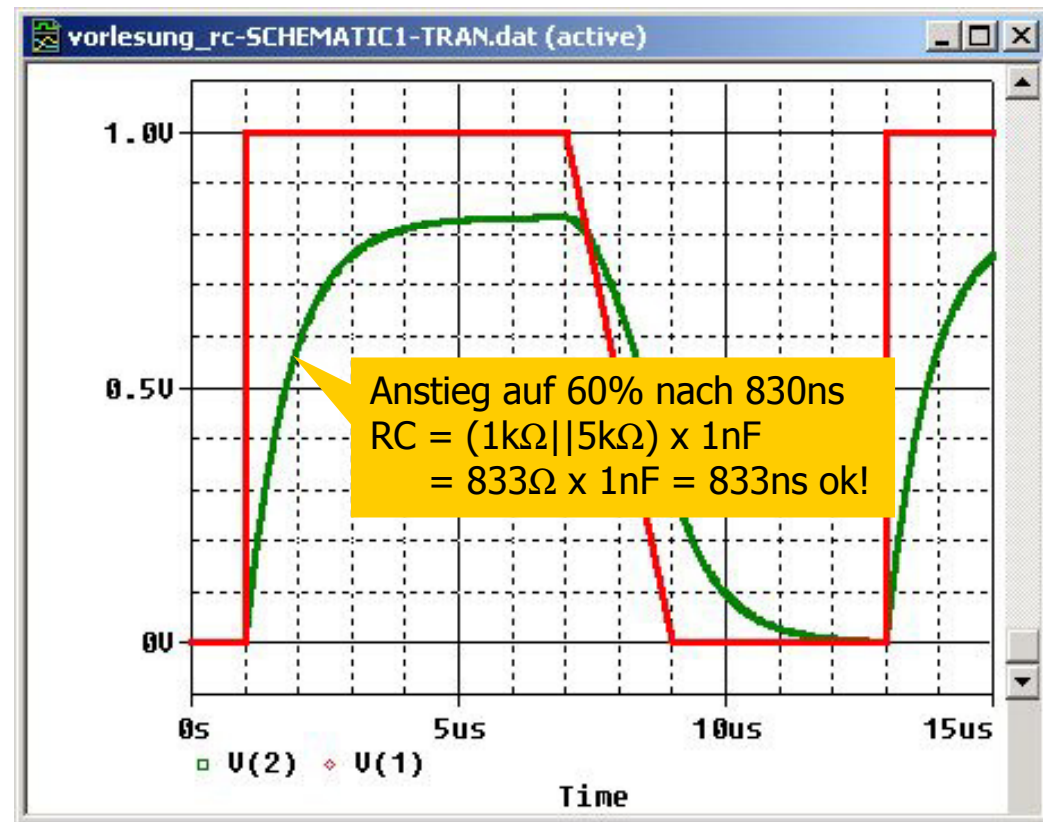


Transienten Analyse

- Dynamische Großsignalanalyse (im Zeitbereich)
- .OP wird berechnet, dann werden die Signalquellen mit der Zeit verändert. Die Signale werden über numerische Integration berechnet. Die Weite der Zeitschritte wird dynamisch so variiert, daß bei schnellen Signaländerungen (Flanken) viele Werte berechnet werden.
- Eine automatische Berechnung der Fouriertransformation ist möglich (.FOUR) – Klirrfaktor, THD etc.



```
V_VIN 1 0 DC 1 AC 1 PULSE 0V 1
+ 1u 10n 2u 6u 12u
R1 1 2 1k
R2 2 0 5K
C1 2 0 1N
.TRAN 0 15U 0 10N
.END
```



Transienten-Analyse CMOS Inverter

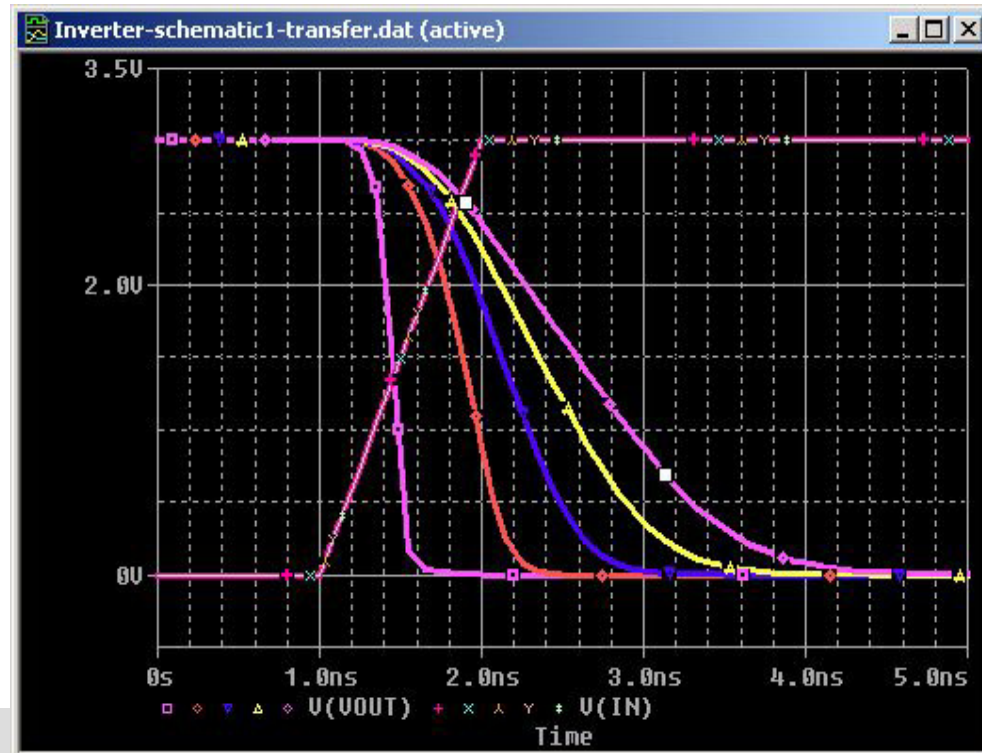
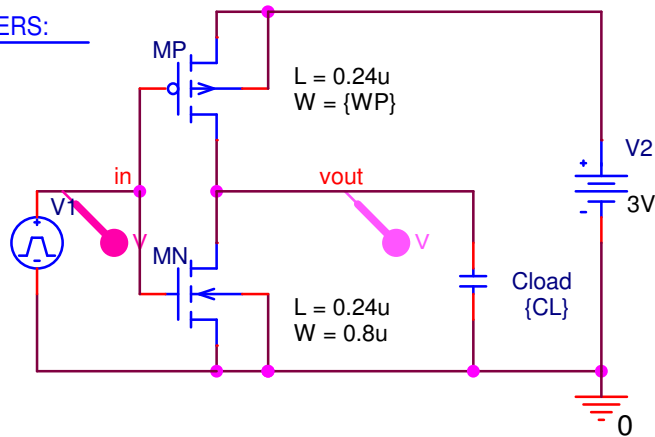
Vorlesung_Inverter.opj

- Ausgangssignal als Funktion der Lastkapazität:

PARAMETERS:

CL = 10f
Voff = 1
WP = 0.8u

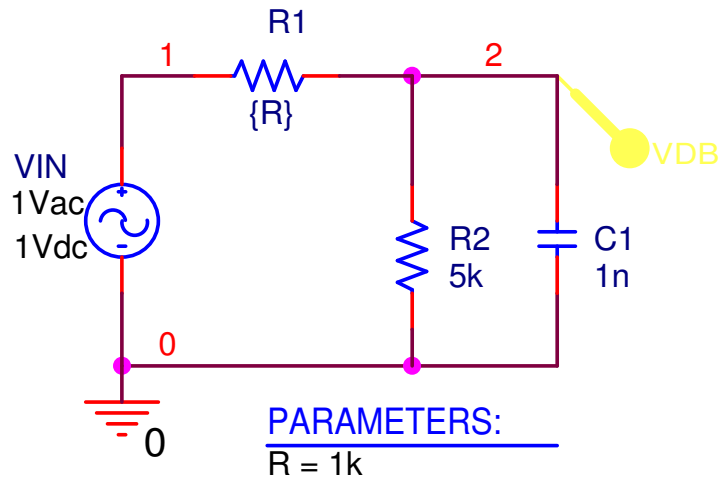
V1 = 0
V2 = 3
TD = 1n
TR = 1n
TF = 1n
PW = 10n
PER = 20n
DC = {Voff}



```
.PARAM      Voff=1 CL=10f WP=0.8u
V_V2       N00048 0 3V
C_Cload    0 VOUT {CL}
M_MN       VOUT IN 0 0 nmos25 L=0.24u W=0.8u AD=0.64p AS=0.64p PD=3.2u PS=3.2u M=1
M_MP       N00048 IN VOUT N00048 pmos25 L=0.24u W={WP} AD=0.64p AS=0.64p PD=3.6u PS=3.6u M=1
V_V1       IN 0 DC {Voff} PULSE 0 3 1n 1n 1n 10n 20n
.TRAN      0 2ns 0 0.1n
.STEP LIN PARAM CL 0 400f 100f
.PROBE
```

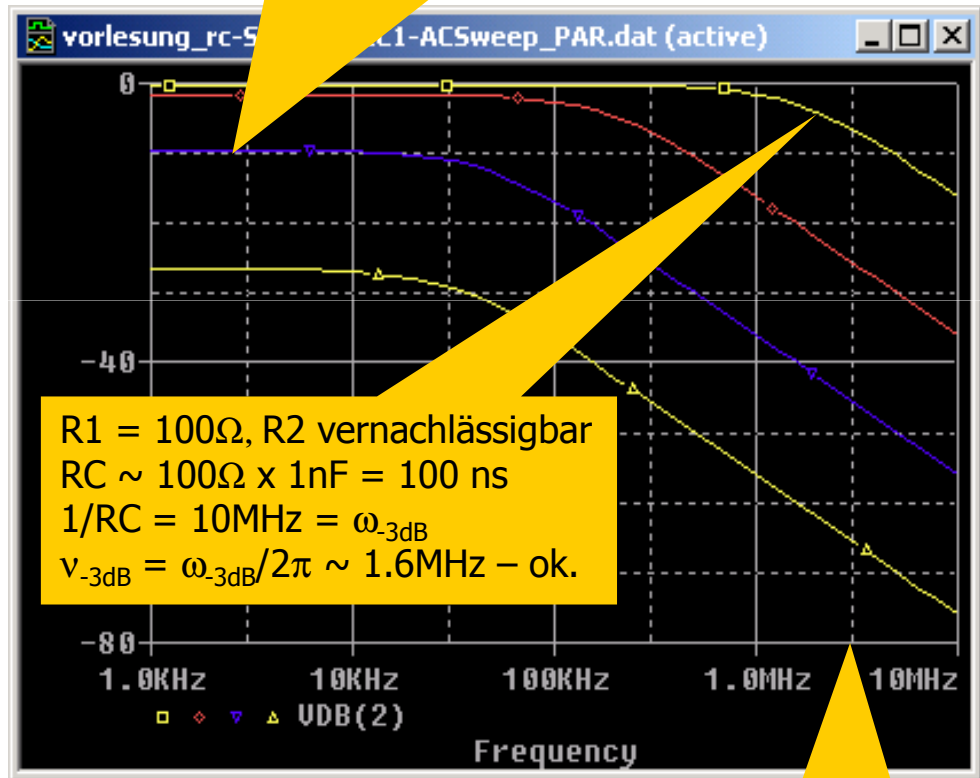

Variation von Parametern

- Parameter von Bauelementen können automatisch variiert werden. Hier mit Liste: 100, 1k, 10k, 100k



```
VIN 1 0 DC 1 AC 1
.PARAM R=1k
R1 1 2 {R}
R2 2 0 5K
C1 2 0 1N
.AC DEC 10 1k 10MEG
.STEP PARAM R LIST 100 1k 10k 100k
.END
```

R1 = 10kΩ
 Amplitude bei $v \rightarrow 0$:
 $V_{AC} \times 5k\Omega / 15k\Omega = 0.33V = -9.5dB$



R1 = 100Ω, R2 vernachlässigbar
 $RC \sim 100\Omega \times 1nF = 100 \text{ ns}$
 $1/RC = 10MHz = \omega_{3dB}$
 $v_{-3dB} = \omega_{3dB}/2\pi \sim 1.6MHz - \text{ok.}$

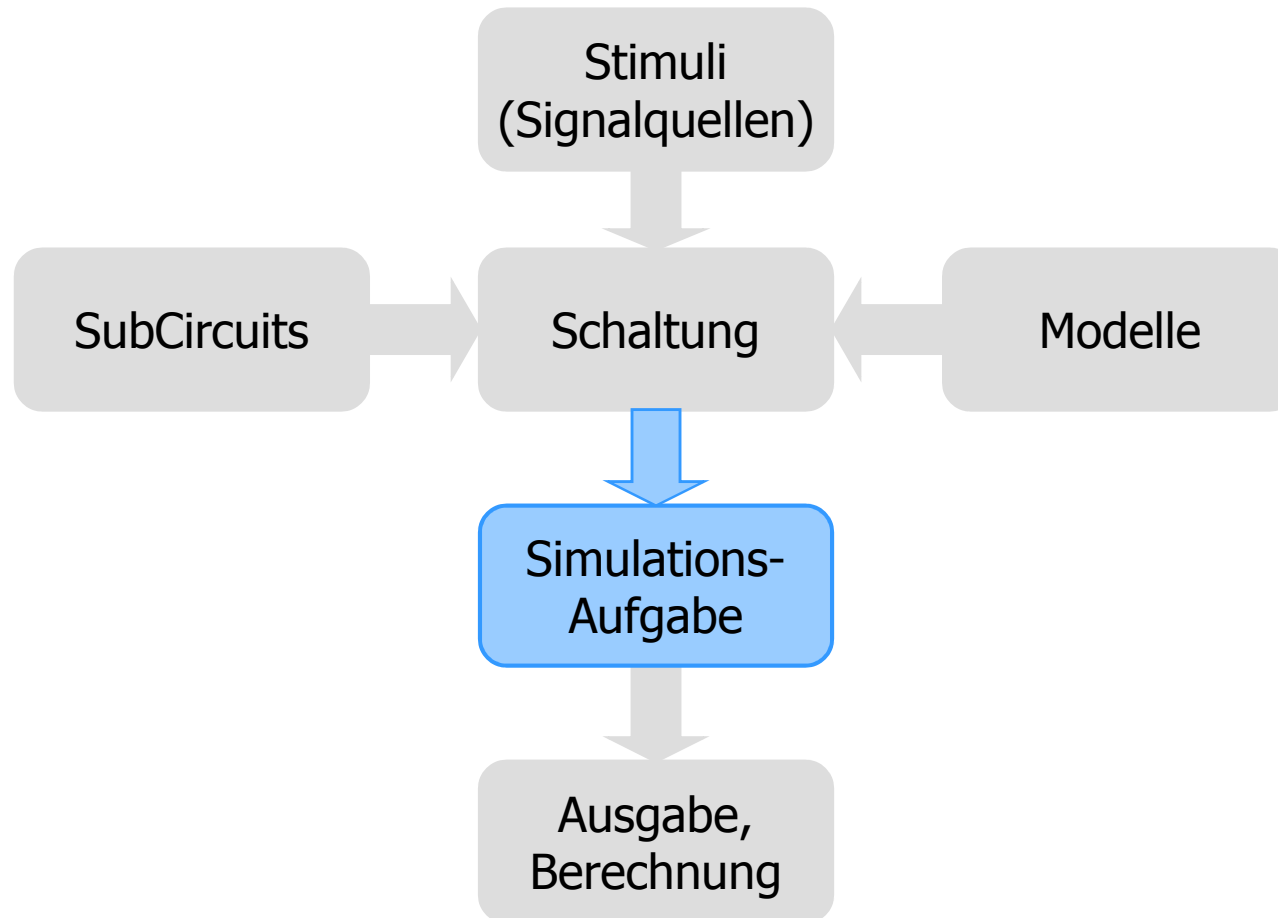
$1 \text{ MHz} \times 10^{0.5} \sim 3.16MHz$

SPICE

Grundelemente und Syntax

SPICE Syntax, globale Elemente

Erste Zeile in File	Wird ignoriert, sollte Projektnamen enthalten
* (manchmal ;)	Achtung: Diese Zeile fehlt in der Netzliste
+	Kommentar
.END	Fortsetzungszeile (Zeilen dürfen manchmal maximal 80 Zeichen lang sein)
T,G,MEG,K,M,U,N,P,F	Markiert Ende des Files (weiter Zeilen werden ignoriert)
3.5e4 oder 4.1e-14	Einheiten
	Zahlen
.INCLUDE FILENAME	File einfügen
.TEMP WERT	Temperatur (Grad Celsius), Default ist TNOM = 27°C. Mehrere Werte sind möglich. Dann wird sie Simulation für jeden Wert wiederholt.
.PARAM NAME=WERT ...	Weist dem Parameter NAME einen Wert zu. PARAM kann dann im File an verschiedenen Stellen benutzt werden.
.OPTION NAME ...	Spezifiziert Optionen für den Simulator. Erlaubte Parameter hängen vom Simulator ab. Beispiele: NOMOD unterdrückt den Ausdruck der Modellparameter, was das File lesbarer macht.
.OPTION NAME=WERT ...	ABSTOL=IVALUE Stromauflösung, Voreinstellung: 1pA RELTOL=VALUE Relative Auflösung von Spannung und Strom (Def.:1e-3)
.IC VN1 = WERT1 VN2 =...	Legt (Spannungs-)Anfangsbedingungen ('Initial Condition') für angegebene Knoten fest. Wichtig z.T. bei Kondensatoren, bistabilen Elementen
.DATA ...	Legt Datenwerte für spätere Benutzung fest (s. Beispiel später)



SPICE Analyse I

.OP

Berechne DC-Arbeitspunkt. Ergebnis wird im .out file ausgegeben.
z.B. zum prüfen, ob Transistoren im richtigen Bereich (lin/sat) arbeiten.

.DC QUELLE START STOP STEP

DC-Analyse. Die QUELLE wird durchgefahren und jeweils der DC Arbeitspunkt bestimmt. Schachtelung ist möglich
(.DC Q1 START1 STOP1 STEP1 Q2 START2 STOP2 STEP2)

.AC TYPE NPOINTS FSTART FSTOP

AC-Analyse.
TYPE kann LIN,DEC oder OCT sein.
Signalquellen müssen dazu AC Signale ausgeben (**beliebter Fehler!**)
Die ACMAG und ACPHASE Attribute aller Quellen werden berücksichtigt.
Zur einfacheren Interpretation setzt man oft die AC Amplitude auf 1V

.TRAN TSTEP TSTOP <TSTART> <TMAX> <UIC>

TSTART Anfangsverzögerung
TMAX maximaler Zeitschritt (SPICE paßt Schritte automatisch an die Änderung der Signale an...)
UIC falls angegeben wird vor der Simulation kein .OP ausgeführt.
In PSPICE heißt der Parameter SKIPBP

.NOISE VN SOURCE

Zusammen mit .AC Sweep. Die Rauschbeiträge aller Bauteile werden generiert und die quadratische Summe am Knoten VN ermittelt. Ein Signal/Rausch-Abstand vom Eingang SOURCE wird berechnet.

SPICE Analyse II – HSPICE Extras

.PZ VN/I(DEVICE) SOURCE

Findet Pol- und Nullstellen (!).

.ALTER

Startet Simulation neu, Zeilen nach .ALTER ersetzen vorhanden Zeilen

.SWEEP PARAM ARGUMENTS

Verändert PARAM und führt Analyse neu durch.

Kann an andere Kommandos angehängt werden. ARGUMENTS =
DEC/OCT/LIN START STOP STEP oder **POI N LISTE** oder **DATA=NAME**

Beispiele:

.TRAN 10N 1U SWEEP TEMP -55 75 10

Führt eine Transientensimulation von -55°C bis $+75^{\circ}\text{C}$ ins Schritten von 10K durch.

.AC DEC 6 1MEG 1G SWEEP load POI 3 1pf 5pf 10pf

Führt eine AC-Analyse mit 6 Punkten pro Dekade von 1MHz bis 1GHz durch. Dabei nimmt der Parameter load die Werte 1 pF, 5 pF und 10 pF an.

.PARAM W1=50u W2=50u L=1u CAP=0

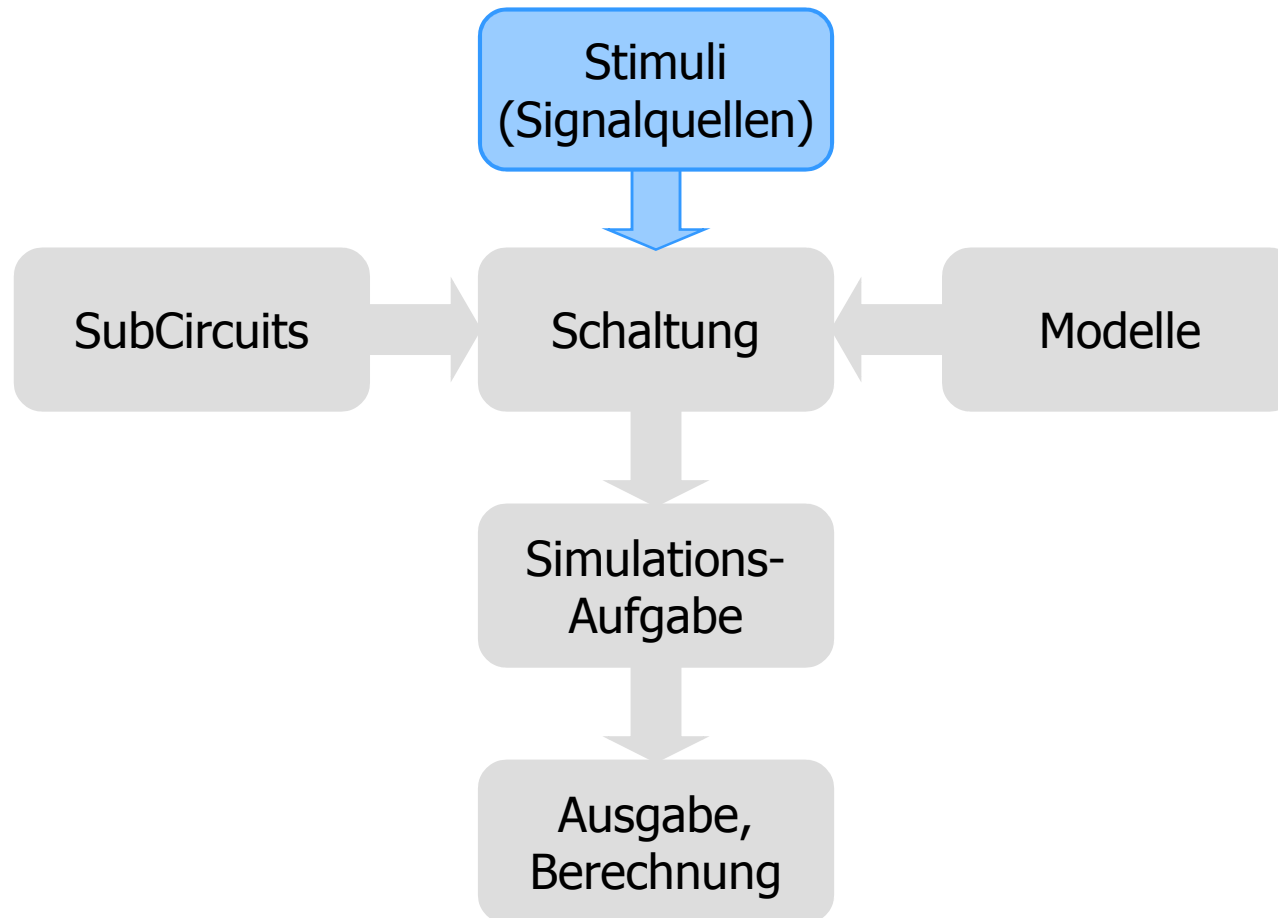
.TRAN 1n 100n SWEEP DATA=d1

.DATA d1

W1	W2	L	CAP
50u	40u	1.0u	1.2pf
25u	20u	0.8u	0.9pf

.ENDDATA

Macht zwei Transientenanalysen mit den im .DATA Feld d1 angegebenen Satz von Parametern



einfache Signalquellen

Vxx N+ N- DC WERT AC AMP,⟨PHASE⟩

Spannungsquelle

Ixx N+ N- DC WERT AC AMP,⟨PHASE⟩

Stromquelle. Optionale Phase des AC Signals in Grad (z.B. 90) (Das Komma ist nicht immer notwendig...)

Vxx N+ N- SIN OFF AMP FREQ DEL ⟨a⟩ ⟨PHASE⟩

Sinussignal (gedämpft)

$T = t - \text{DEL}$

$V(t) = \text{OFF} + \text{AMP} \exp[-a T] \sin[2\pi \text{FREQ} T + (\text{PHASE}/360)]$

Vxx N+ N- PULSE V0 V1 DEL TR TF WIDTH PERIOD

Rechtecksignal (periodisch)

Springt zur Zeit DEL von V0 auf V1.

TR = Anstiegszeit, TF = Abfallzeit.

Einmaliges Signal durch PERIOD >> Simulationsintervall.

Vxx N+ N- PWL T0 V0 T1 V1 T2 V2

Stückweise lineares Signal ('PieceWise Linear').

Ti Vi sind die Eckpunkte des Kurvenverlaufs

Beispiele:

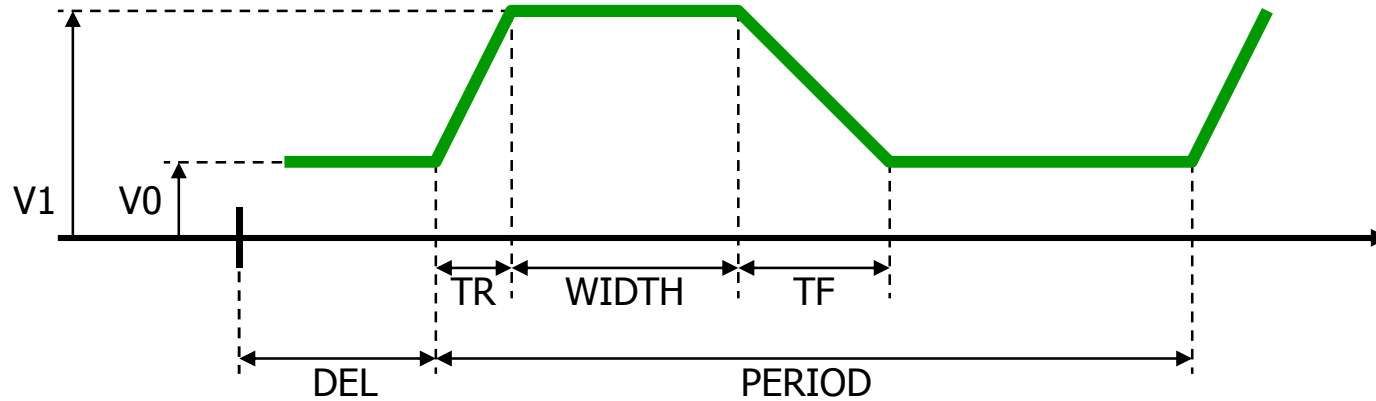
V2 in 0 VCC AC 1 PWL 0 0 10N VCC 15N VCC 20N 0

Spannungsquelle V2 hat eine DC Spannung, die durch Parameter VCC gegeben ist, und einen AC Anteil von 1V. Im Zeitbereich erzeugt sie einen Puls, der von 10ns bis 15ns auf VCC springt. Der Arbeitspunkt (.DC) wird NICHT aus dem DC Wert, sondern aus dem PWL Wert (hier 0V bei t=0) berechnet.

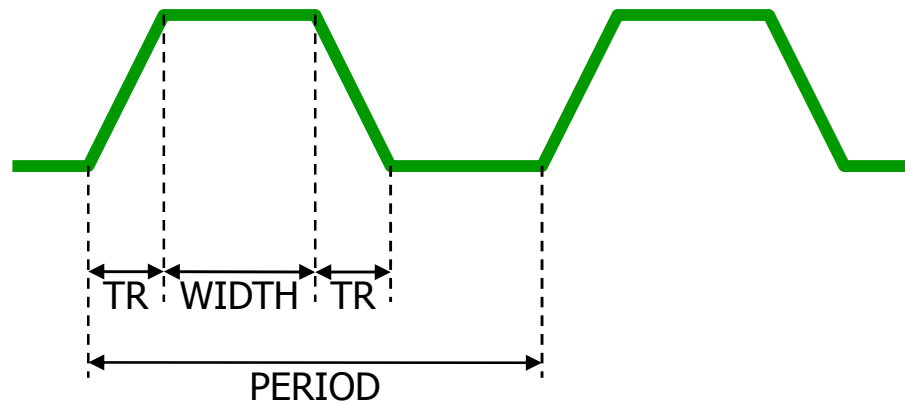
VIN 13 2 0.001 AC 1 SIN 0 1 1MEG

Spannungsquelle VIN hat eine DC Spannung von 1mV, eine AC Spannung von 1V. Im Zeitbereich erzeugt sie eine Sinusspannung um 0V mit einer Amplitude von 1 V mit der Frequenz $\nu = 1 \text{ MHz}$.

Detail: Vpulse



- Um (bei $TR=TF$) exakt gleiche Tastverhältnisse zu bekommen, muss $TR + WIDTH = PERIOD / 2$ sein:



gesteuerte Signalquellen

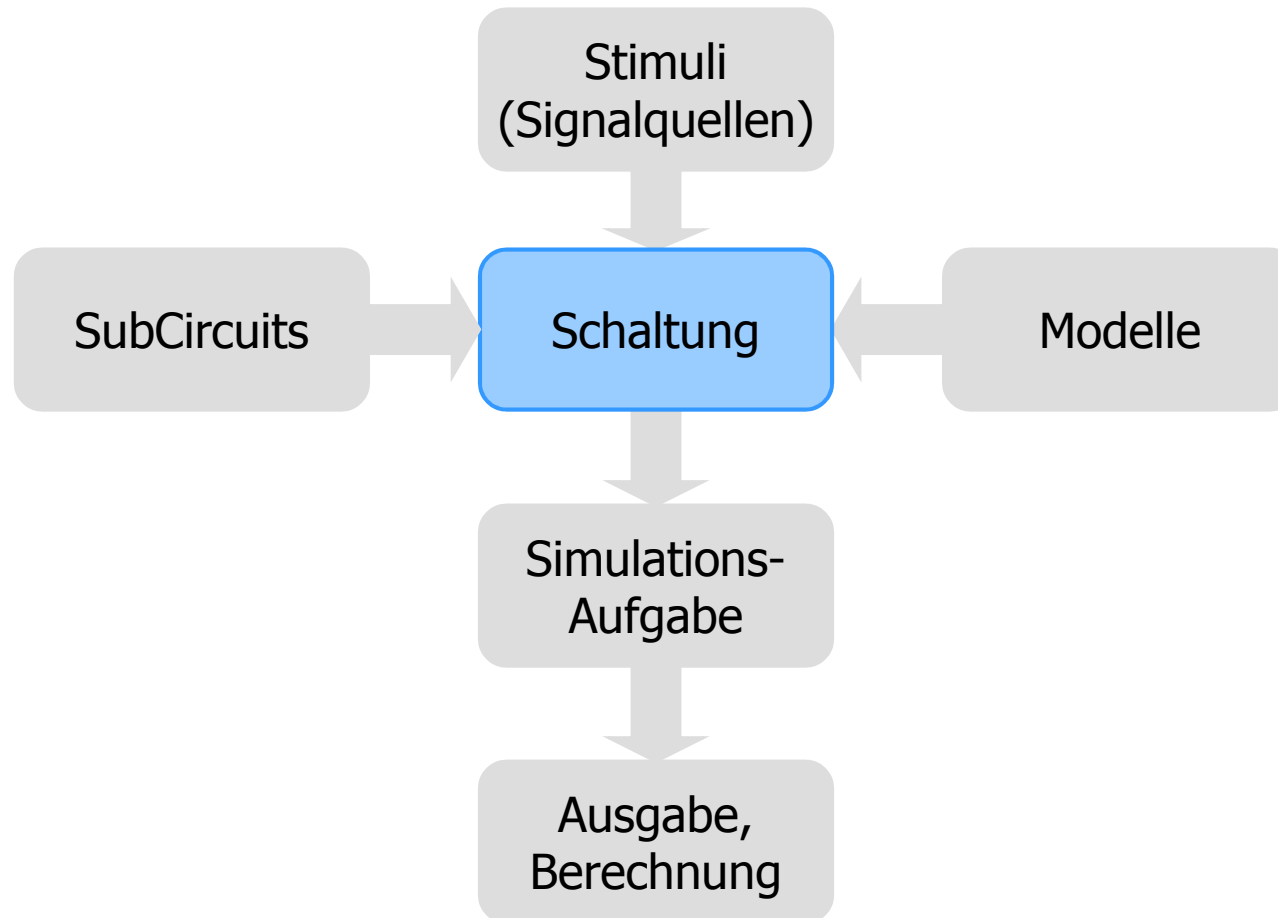
- E_{xx} NO+ NO- NC+ NC- VERST** Spannungsgesteuerte **Spannungs**quelle (VCVS)
NO+ und NO- sind die Ausgänge
NC+ und NC- sind die Eingänge zur Steuerung ($R_{in} = \infty$)
VERST ist die Verstärkung
Verwendung z.B. für idealen Operationsverstärker, zur Entkopplung
- G_{xx} NO+ NO- NC+ NC- VERST** Spannungsgesteuerte **Strom**quelle (VCCS)
VERST ist die Verstärkung (Transkonduktanz [A/V])
Verwendung z.B. als 'idealer' Transistor
- H_{xx} NO+ NO- QUELLE VERST** **Strom**gesteuerte **Spannungs**quelle (CCVS)
NO+ und NO- sind die Ausgänge
QUELLE ist der Name einer Spannungsquelle mit Spannung 0V,
an der der Strom gemessen wird.
VERST in [V/A].
- F_{xx} NO+ NO- QUELLE VERST** **Strom**gesteuerte **Strom**quelle (CCCS)

Weitere Parameter sind z.B. **MAX=WERT MIN=WERT**
Verzögerungselemente, z.B. **E_{xx} NO+ NO- DELAY in+ in- TD=VERZ**
Die Transfer Charakteristik kann auch durch Polynome angegeben werden

Beispiel idealisierter Operationsverstärker:

E1 out 0 inp inn 100000 MAX=VCC MIN=0

Verstärkung ist 100000, Ausgangsspannung ist auf 0...VCC begrenzt.



Elementare Komponenten

Rxx N+ N- WERT

Rxx N+ N- MODEL WERT

Widerstand. Temperaturabhängigkeit ist parametrisiert

Widerstand mit komplizierterem Modell (z.B Temperatur-
Abhängigkeit)

Cxx N+ N- WERT <IC=WERT>

Lxx N+ N- WERT <IC=WERT>

Kondensator, IC= Anfangsspannung

Spule, IC = Anfangsstrom

Dxx NAn NKat MODEL <FLÄCHE>

Qxx NC NB NE <NSUB> MODEL <FLÄCHE>

Mxx ND NG NS <NB> MODEL <W=...> ...

Diode mit Modell MODEL (z.B. 1N4148)

Bipolarer Transistor mit Modell MODEL, Emitterfläche

MOS Transistor. Viele optionale Parameter

Demo IC:
WienOszillator

Beispiele:

Q13 15 3 0 1 NPNSTRONG 1.5

; bipolarer NPN mit Emitterfläche 1.5 x nominal

M1 out in VDD VDD PNOM L=25u W=12u ; PMOS

M28 1 2 0 0 NWEAK

; NMOS

+ L=0.24u W=0.8u M=2

; Geometrie, Multiplikator

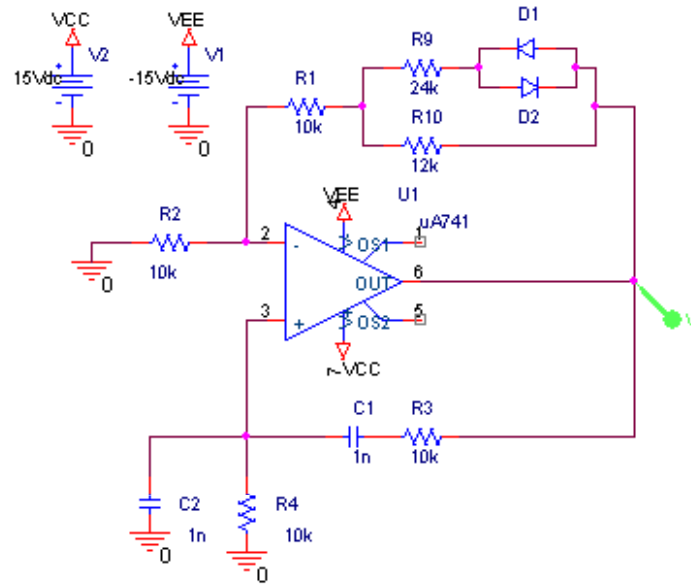
+ AD=288p AS=288p PD=60u PS=60u

; Kapazitäten (Area und Sidewall für Drain, Source)

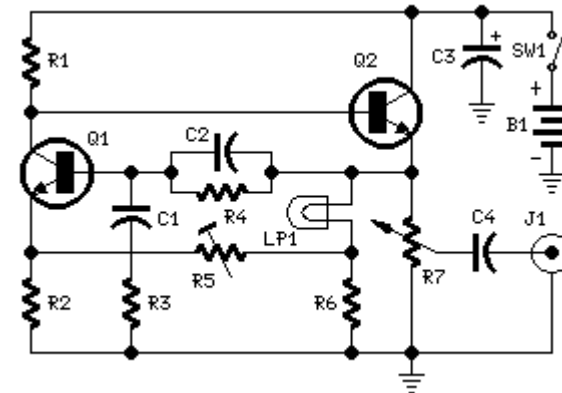
+ NRD=14 NRS=24 NRG=10

; Widerstände (Drain, Source, Gate) (*RSH)

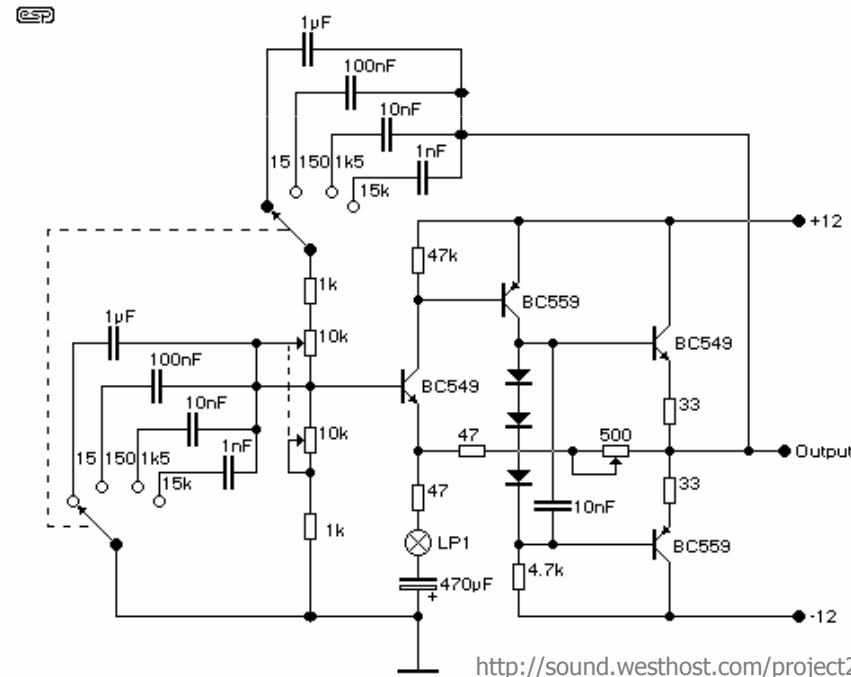
Schaltungsvarianten



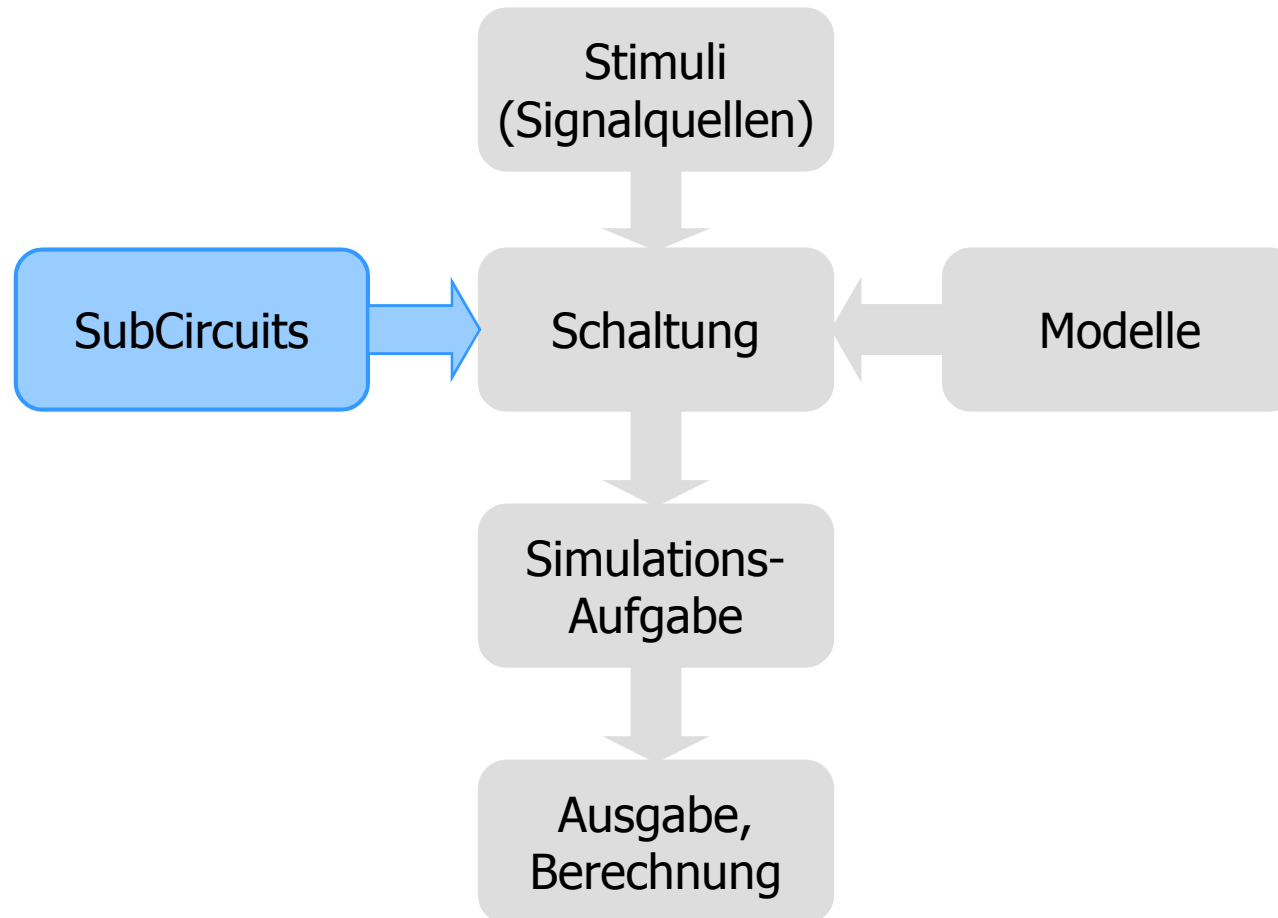
www.calvin.edu/~pribeiro/courses/engr332/Handouts



<http://www.redcircuits.com/Page13.htm>



<http://sound.westhost.com/project22.htm>



Subcircuits

- Subcircuits (.SUBCKTs) kapseln (komplexe) Schaltungsteile in einen Block
- Sie können mehrfach verwendet werden
- Wichtig für den Aufbau einer Design-**Hierarchie**

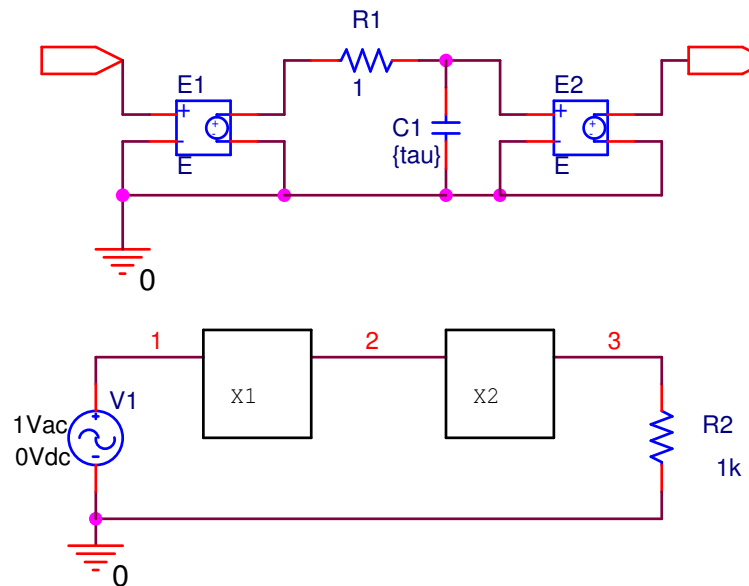
.SUBCKT SNAME N1 N2 ...
 + **PARAMS:** NAME = WERT, ...
 ... Definition der Schaltung ...
.ENDS <SNAME>

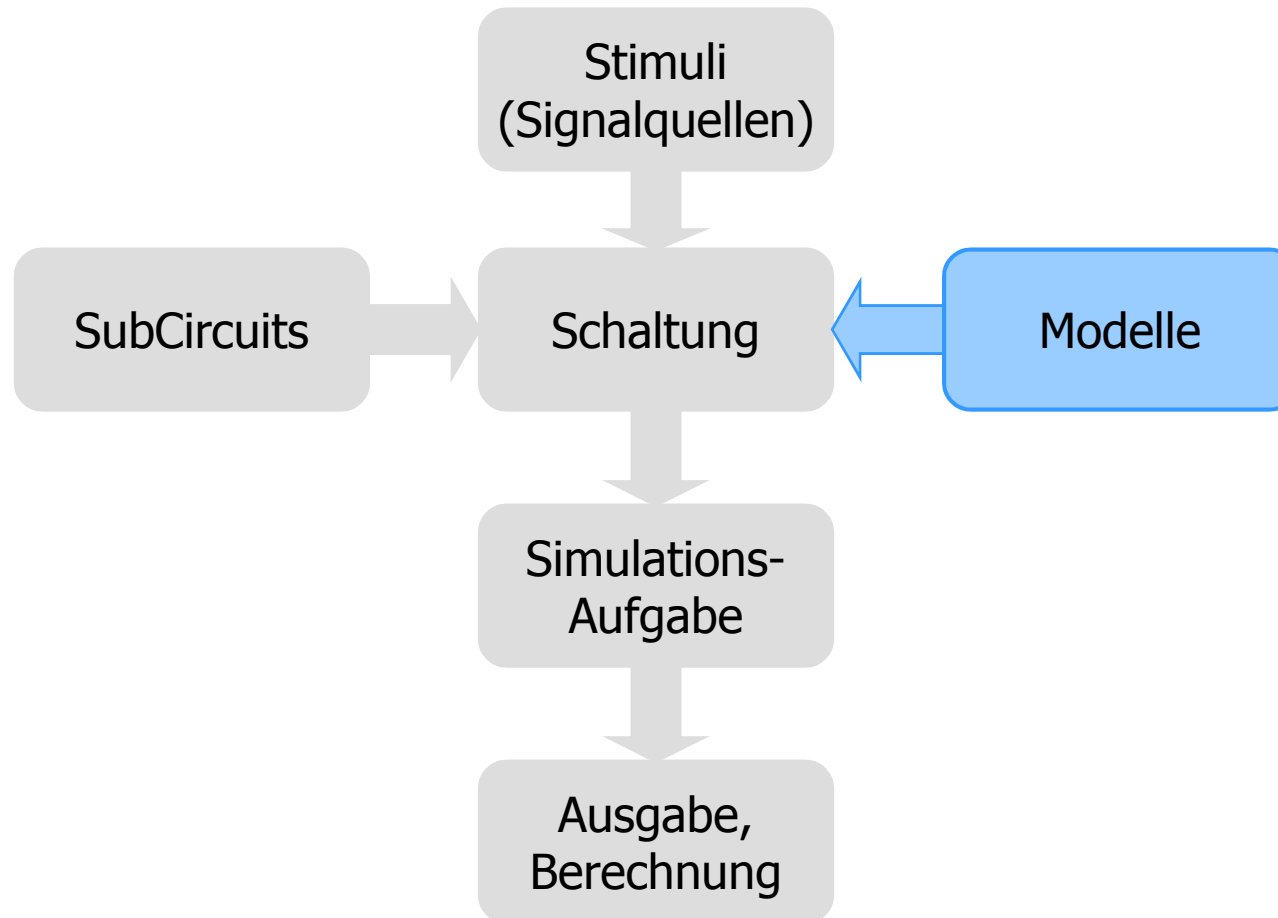
Definition des Subcircuit 'SNAME'
 SNAME ist der Bezeichner
 Ni sind die benötigten Knoten
 In der (optionalen) Liste können Parameter definiert werden

Xxx N1 N2 ... SNAME **PARAMS:** ...

Benutzung eines Subcircuits. Ni sind die Knoten, SNAME der Typ

```
* Kommentar: Filter mit SubCircuit
.SUBCKT FILTER IN OUT PARAMS: TAU=0
E1 1 0 IN 0 1
R1 1 2 1
C1 2 0 {TAU}
E2 OUT 0 2 0 1
.ENDS
V1 1 0 AC 1
X1 1 2 FILTER PARAMS: TAU=1U
X2 2 3 FILTER PARAMS: TAU=1U
R2 3 0 10MEG
.AC DEC 10 10K 10MEG
.PLOT AC VDB(3)
.END
```





Modelle: Beispiel Widerstand

- Sogenannte **Referenz-Modelle** sind feste Bestandteile von SPICE.
- Mit dem .MODEL Befehl werden daraus benutzerspezifische Modelle generiert (Variation von Parametern)

.MODEL NAME REFERENZ (parameterliste)

Deklaration des Modells NAME

Referenz = RES, D, NPN, PNP, NMOS, PMOS, ...

.MODEL TERM RES (VALUE = 50 TC1 = 0.1)

VALUE – WERT (multiplikativ)

TC1 – linearer Temperaturkoeffizient [Ω/K]

TNOM = 27°C

$$\text{Widerstandswert} = \langle \text{value} \rangle \cdot R \cdot [1 + TC1 \cdot (T - T_{\text{nom}}) + TC2 \cdot (T - T_{\text{nom}})^2]$$

* Beispiel Widerstandsmodell mit Temp-Koeff.

.MODEL TERMINATOR RES (R=50 TC1=0.1)

V1 1 0 1

R1 1 2 50

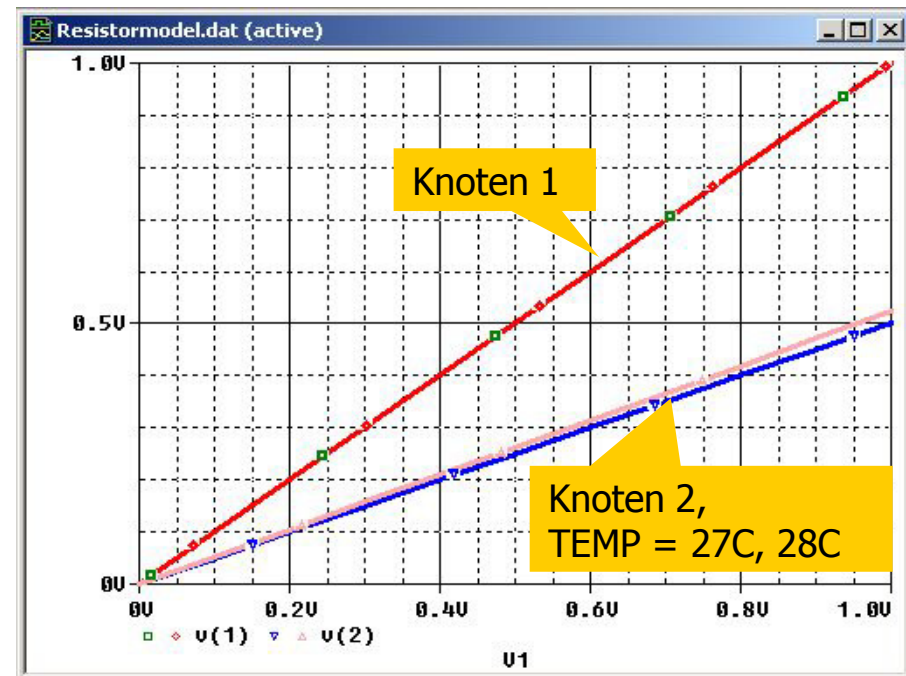
R2 2 0 TERMINATOR 1

.DC V1 0 1 0.1

.TEMP 27 28

.PRINT DC V(1) V(2)

.END

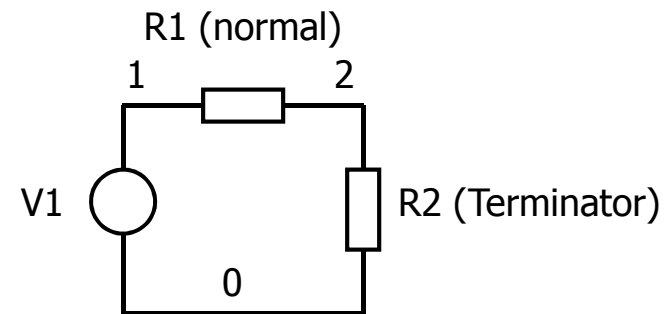


Kommandozeilen - SPICE

- Mit PSPICE ist auch ein Betrieb von der Kommandozeile aus möglich:

Demo: Kommandozeilen-Betrieb
Directory: C:\Program Files\OrCAD_Demo\Pspice)
Aufruf Sim: pspice -r <file.cir>
Aufruf Plot: pspice -c <file.cmd>

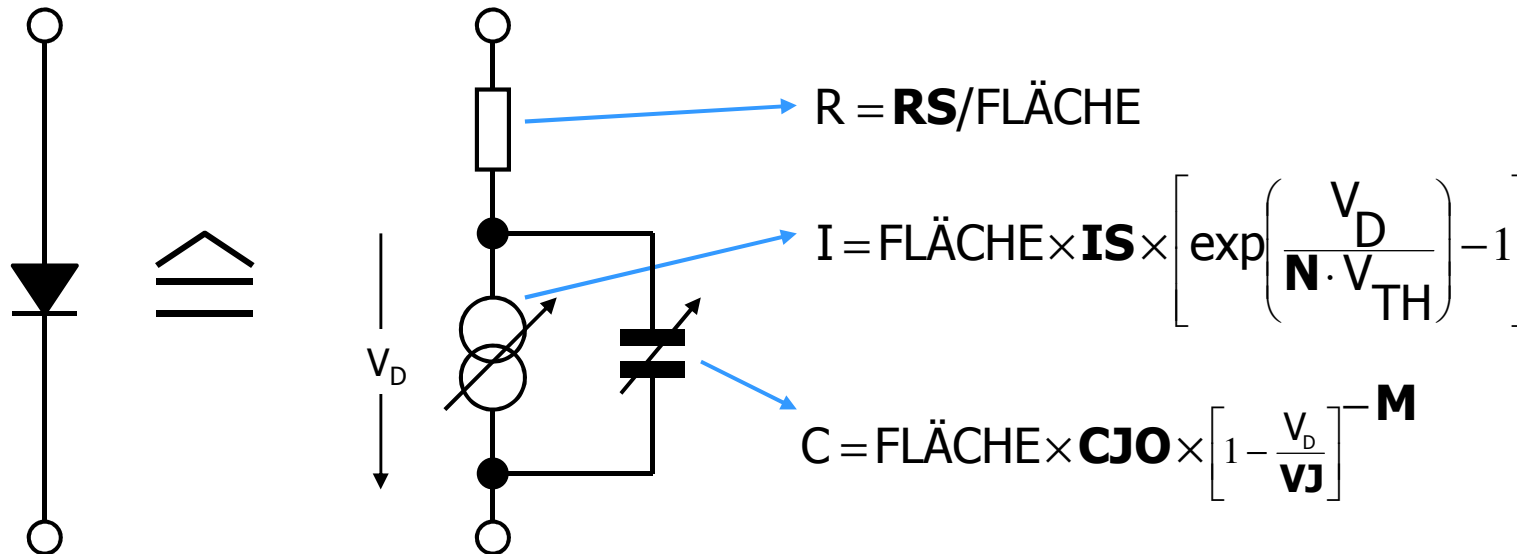
```
* Test of command line operation
* Beispiel Widerstandsmodell
.MODEL TERMINATOR RES ( R=50 TC1=0.1 )
V1 1 0 1V
R1 1 2 50
R2 2 0 TERMINATOR 1
.DC V1 0 1 0.1
.TEMP 27 28
.PROBE
.END
```



Modelle: Diode

Dxx NAnode NKathode MODEL <FLÄCHE>

Diode in der Netzliste. Fläche ist Multiplikator

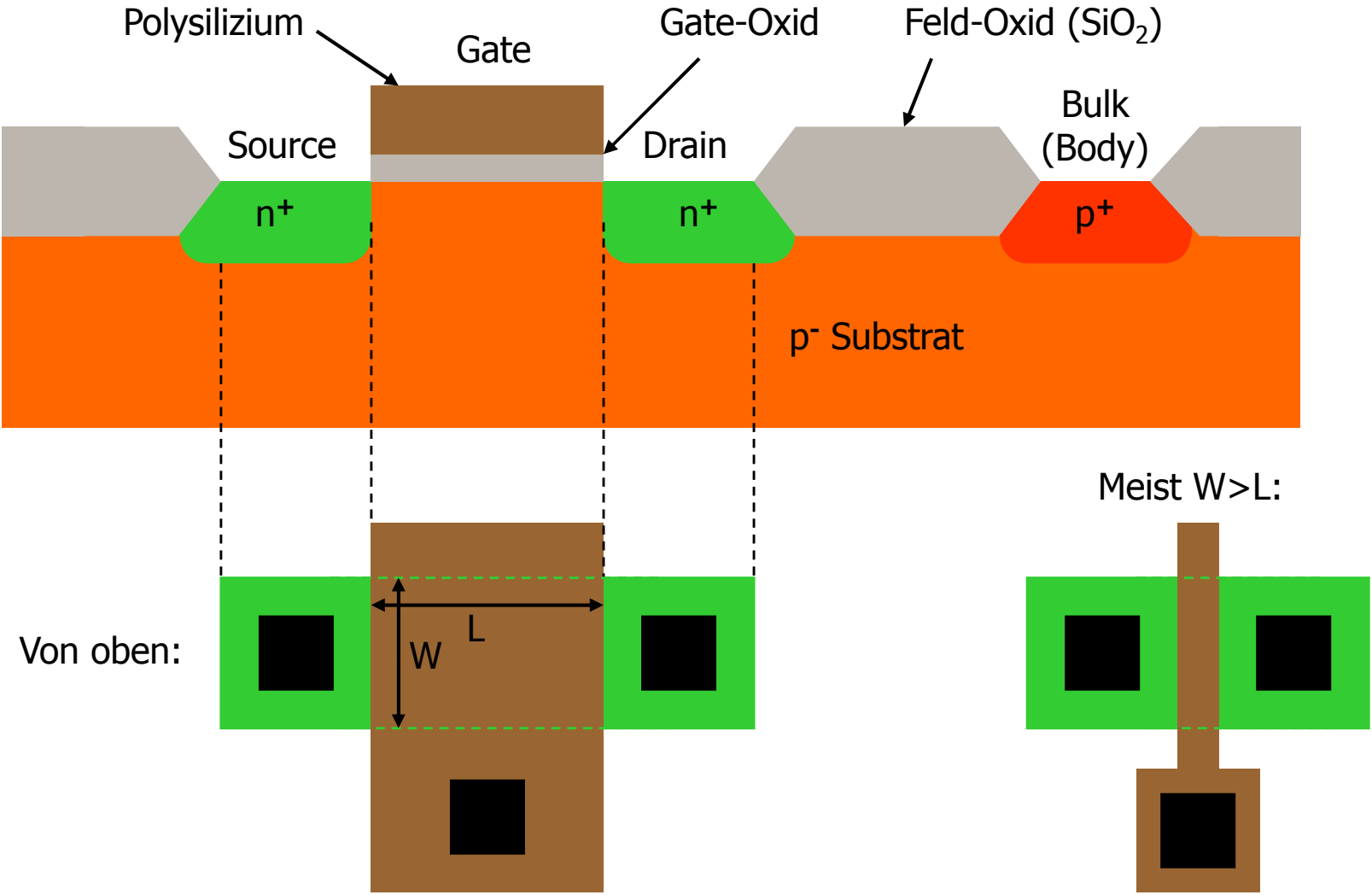


.MODEL D1N4148 D (IS=0.1PA, RS=16, CJO=2PF, ...)

- RS: Serienwiderstand
- IS: Sättigungsstrom
- N: Emissionskoeffizient (normalerweise: N=1)
- CJO: Sperrschichtkapazität bei $V_d=0V$ ($C_{junction,0}$)
- M: Kapazitätsexponent (0.3...0.5)
- VJ: Intrinsisches pn-Potential (typ. 1V)

Das 'volle' Modell (s. z.B. PSPICE Reference, S. 134) enthält noch viele Erweiterungen (Rekombinationsstrom, Leckstrom, Temperaturabhängigkeiten, Rauschen, ...)

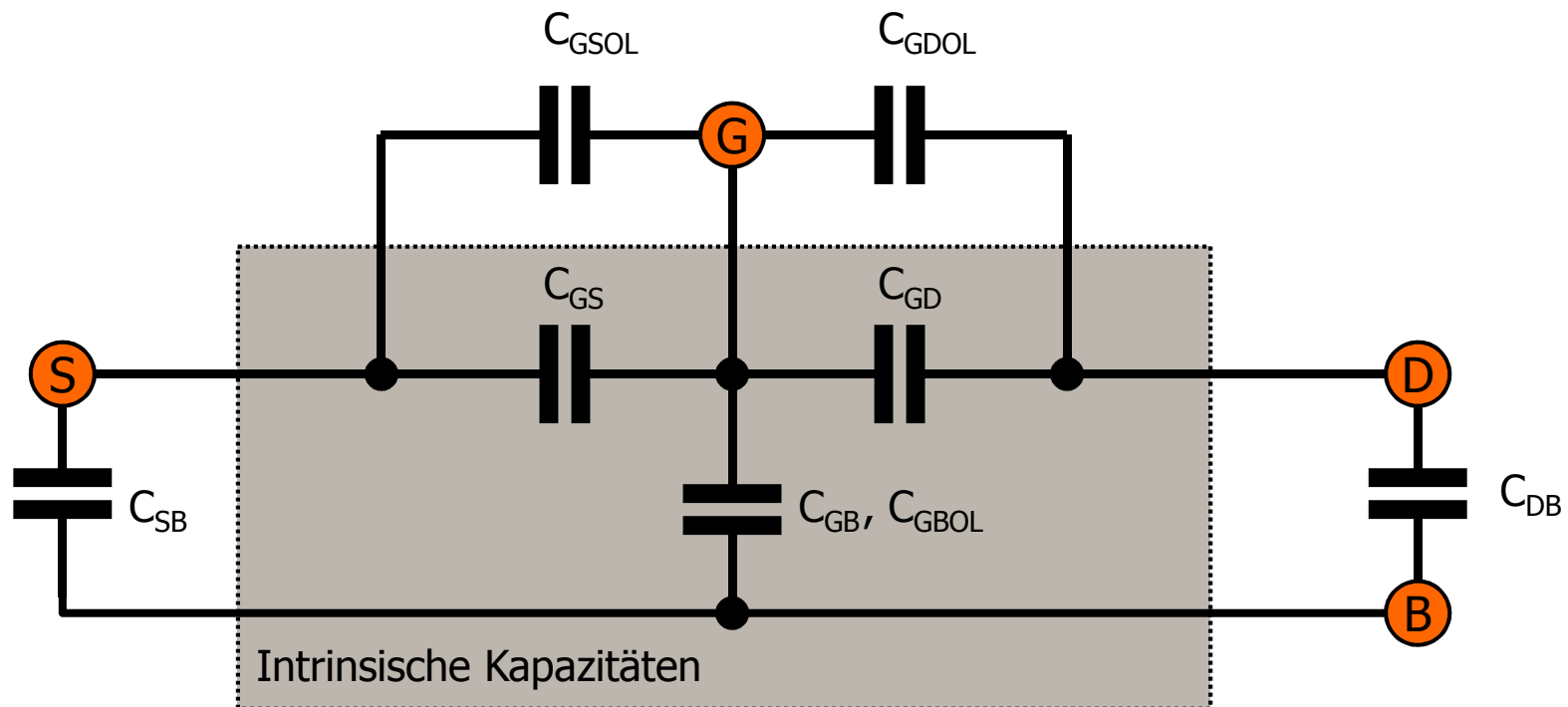
Modelle: MOS - Aufbau NMOS Transistors



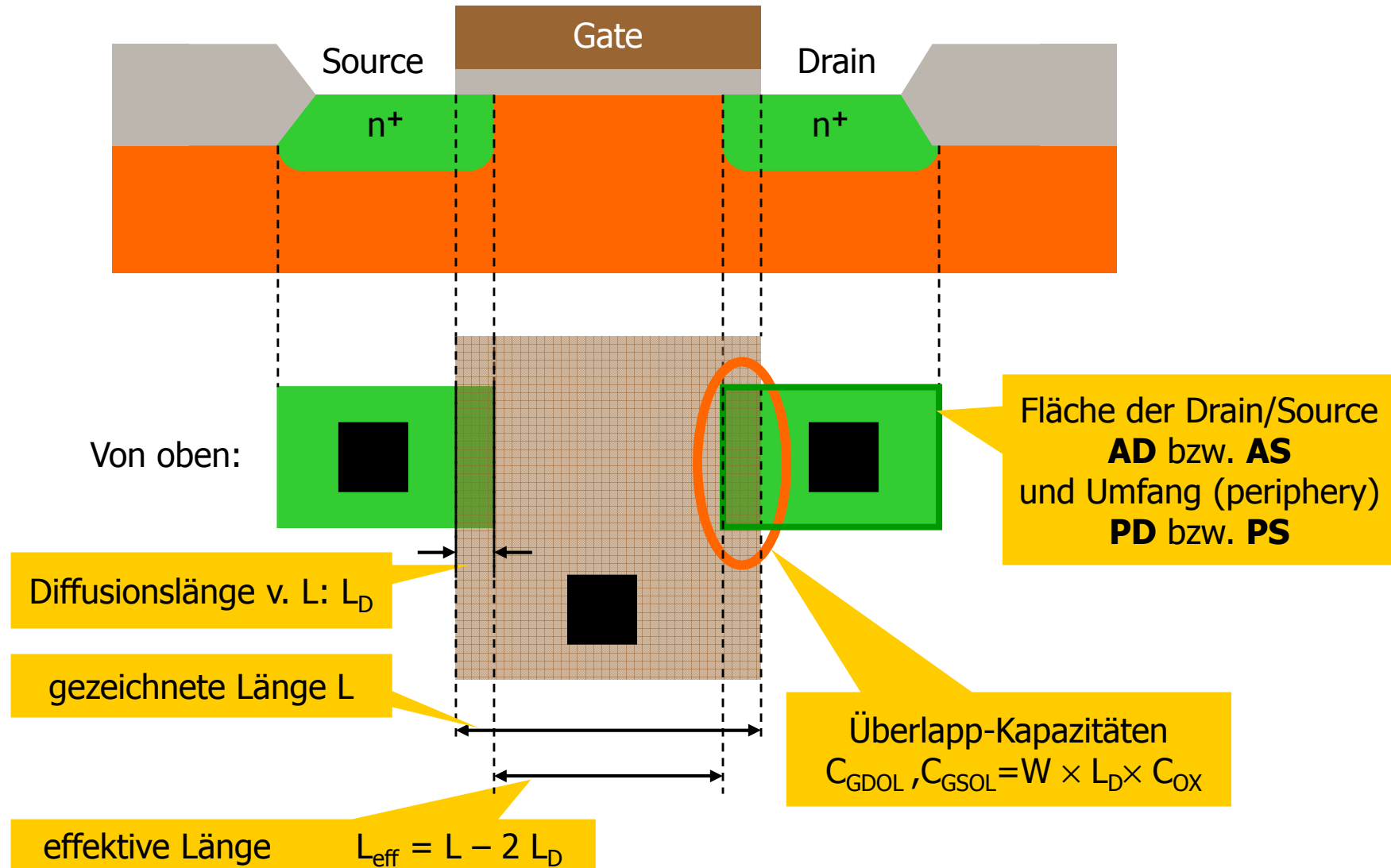
Kapazitäten im MOS Transistor

Man unterscheidet im Wesentlichen:

- **intrinsische Kapazitäten** vom Gate zum Kanal und ins Substrat. Sie hängen von der Gate-Spannung und vom Betriebszustand ab (MOS Struktur)
- Die **Sperrschichtkapazitäten** (C_{SB} , C_{DB}) von Drain und Source. Sie hängen von V_{DB} bzw V_{SB} ab.
- **Überlapp-Kapazitäten** (C_{GSOL} , C_{GDOL} , C_{GBOL}) 'direkt' vom Gate-'Metall' in die Drain/Source/Bulk-Gebiete
- Insgesamt also 5 Kapazitäten (C_{SD} ist klein). C_{GB} ist klein

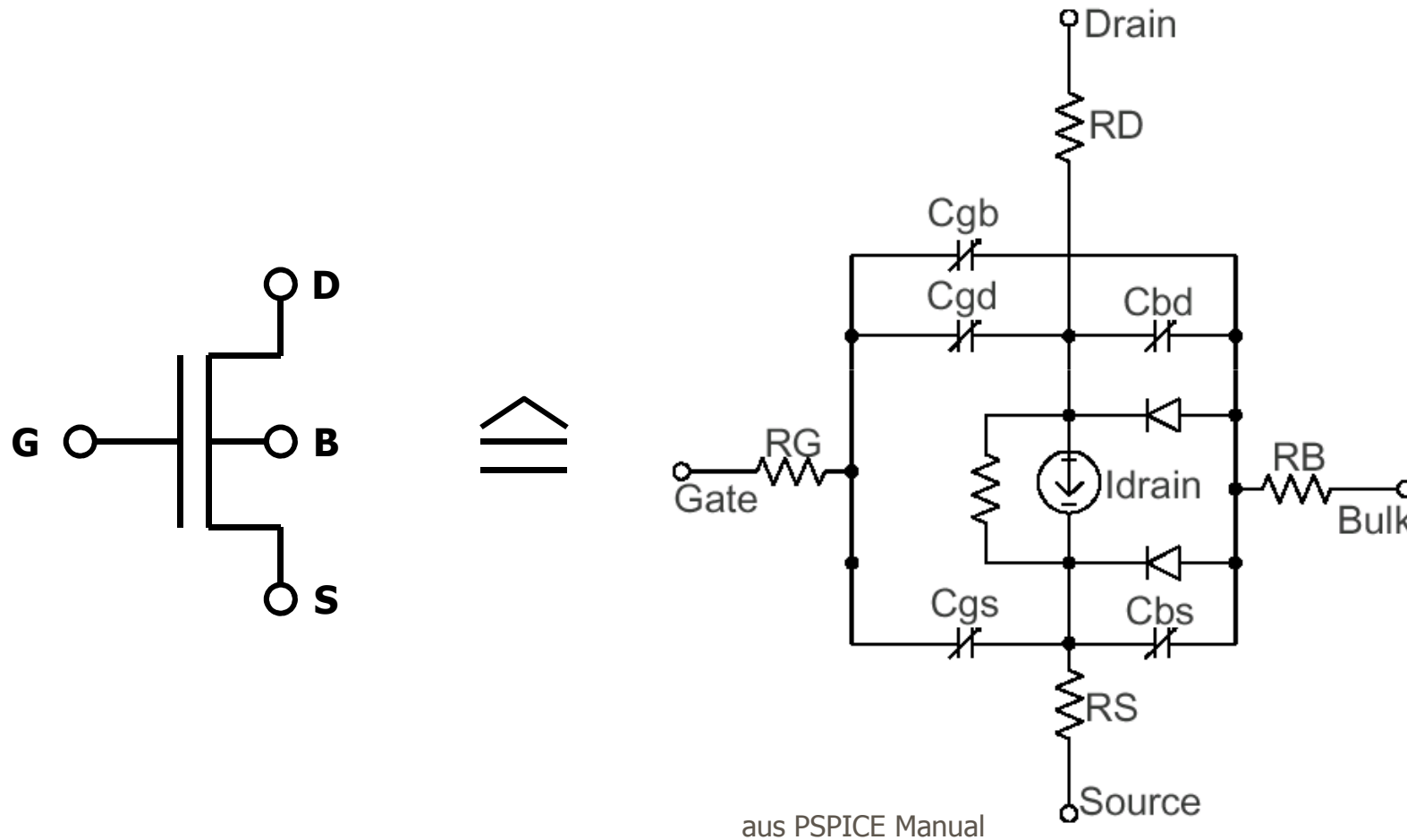


Überlappkapazität, Source/Drain Kapazität



Modelle: MOSFETs – mehr später

MOSFET Modell = FET + D/S-Dioden + Zugangswiderstände + Kapazitäten + (Rauschen, Leckströme,...)



Modelle: MOSFETs – mehr später

.MODEL Name NMOS/PMOS (LEVEL=...

LEVEL gibt den TYP des Modells an.
Es muß vom Simulator unterstützt werden !

W=...

Standardbreite des Gates

L=...

Standardlänge des Gates

AD=... AS=...

Fläche Drain/Source ('Area Drain') zur Kapazitätsberechnung

PD=... PS=...

Peripherie Drain/Source zur Kapazitätsberechnung

VTO=...

Schwellesspannung

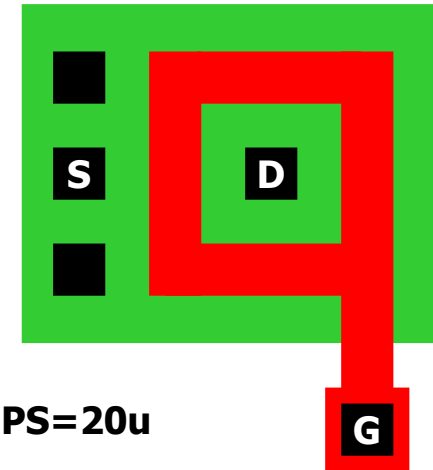
KP=...

Transkonduktanzparameter

TOX=...

Oxiddicke

)



Beispiele eines 'annular' Transistors (AD≠AS):

Mxx ND NG NS <NB> NSPECIAL W=10u L=1u AD=30p AS=200p PD=5u PS=20u

.MODEL NSPECIAL NMOS LEVEL=2 PHI=0.6 TOX=4.35E-8 XJ=0.2U TPG=1
+ VTO=.8756 DELTA=8.56 LD=2.4E-07 KP=4.55E-05 UO= 573.1 UEXP= 1.59E-1
+ UCRIT=5.9160E+4 RSH=1.03E+1 GAMMA=0.41 NSUB=3.31E+15 NFS=1.18E+12
+ VMAX=6.02E+4 LAMBDA=2.93E-2 CGDO=2.8E-1 CGSO=5.3E-10
+ CGBO=4.09E-10 CJ=1.1E-04 MJ=0.66 CJSW=2.2E-10 MJSW=0.18 PB=0.8

Modelle

- Manche SPICE Implementierungen bieten bis zu 20 verschiedenen Modell-Typen ('LEVEL=xx')
 - Wir verlassen uns (meist) auf die Empfehlungen der Hersteller und auf Experten...
-
- LEVEL=1 Shichman-Hodges
 - LEVEL=2 Grove-Frohman
 - LEVEL=3 Empirisches Modell
 - LEVEL=4 BSIM1 (Berkeley Short Channel IGFET) SPICE3, PSPICE
 - LEVEL=5 BSIM2 Jeng Model SPICE3
 - LEVEL=5 **BSIM3** (version1) PSPICE
 - LEVEL=6 BSIM3 (version2) PSPICE
 - LEVEL=6 MOS6 Sakurai-Newton - SPICE3
- Werden von allen Simulatoren unterstützt
- z.Z. 'Industriestandard'
-
- s. z.B. PSPICE Reference, S. 174

Level 1 Modell

$$I_D = K/2 W/L (V_{GS} - V_T)^2 (1 + \lambda V_{DS}), \quad \text{gilt in Sättigung } (V_D > V_{GS} - V_T)$$

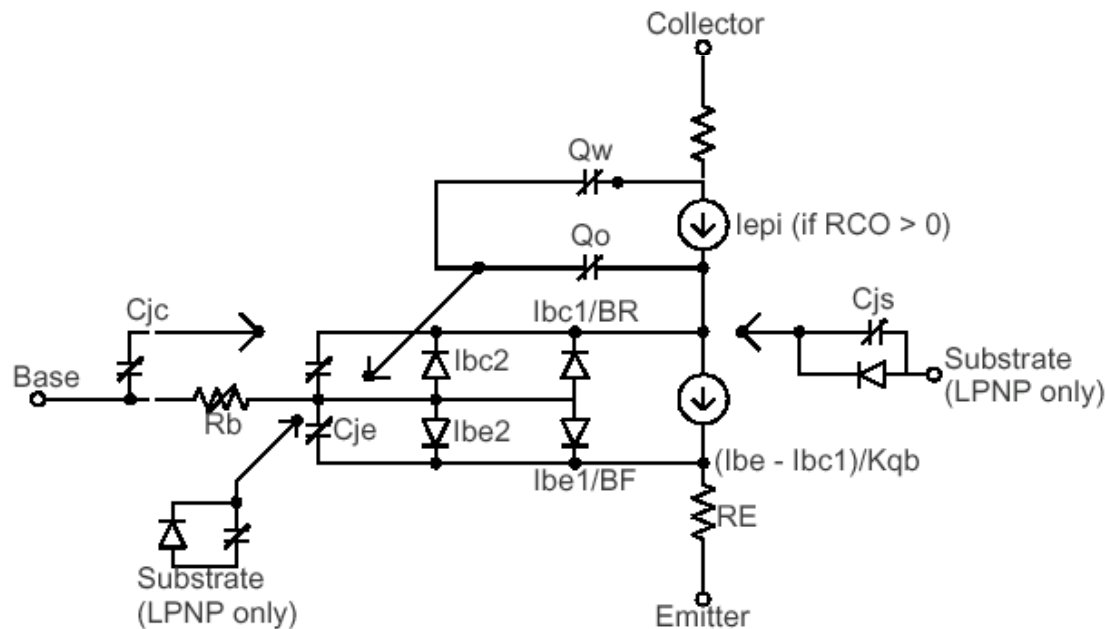
- Für Handrechnungen geeignet.
- Wichtigste SPICE Parameter:
.model mos1 nmos_simple
+ level=1
+ kp=100e-6 ; Transkonduktanzparameter
+ vto=1.0 ; Schwelle (negativ bei PMOS Transistoren)
+ lambda=0.1 ; Kanallängenmodulation, Ausgangswiderstand, Early-Effekt
+ gamma=0.5 ; Für Substrateffekt
+ cox=1e-3 ; Oxid-Kapazität (Fm⁻²)
+ tox=10e-9 ; Oxid-Dicke
- VT hängt von der Spannung des Substrats ab (später...)
- Bei einem Plot von sqrt(I_D) gegen V_{GS} sollte eine Gerade rauskommen!

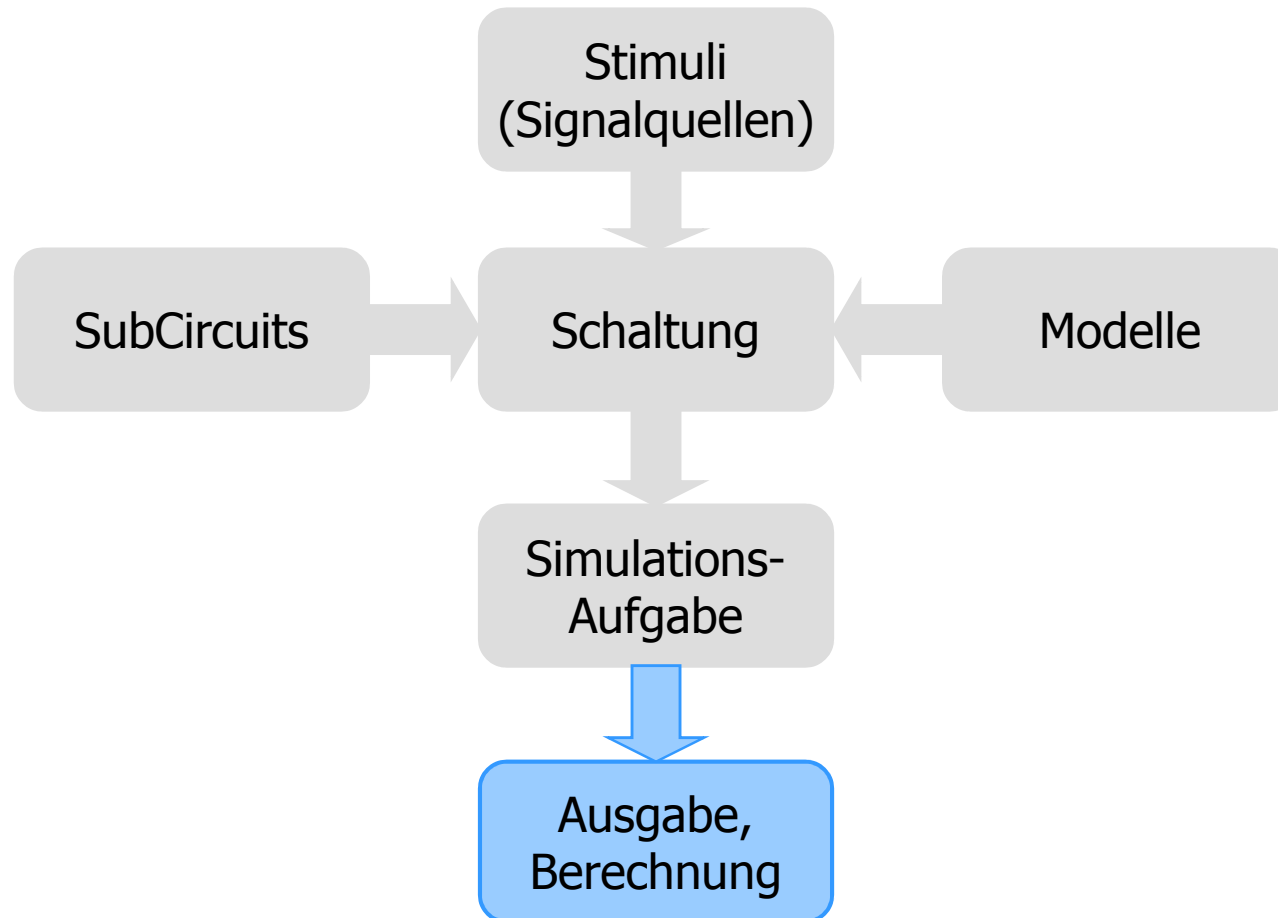
Modelle: Bipolare Transistoren

Für einen Bipolartransistor mit 'Einheitsfläche':

.MODEL Name NPN/PNP (BF=... Stromverstärkung Beta (Def. = 100)
 IS=... Sättigungsstrom (Def. = 1e-16 A)
 VAF=... Early - Spannung (Def. = unendlich)
 ...)

.MODEL Q2N2222A NPN (IS=14.34F VAF= 74.03 BF=255.9 XTI=3 EG=1.11
+ NE=1.307 ISE=14.34F IKF=.2847 XTB=1.5 BR=6.092 NC=2 ISC=0 IKR=0 RC=1 CJC=7.306P MJC=.3416
+ VJC=.75 FC=.5 CJE=22.01P MJE=.377 VJE=.75 TR=46.91N TF=411.1P ITF=.6 VTF=1.7 XTF=3 RB=10)





Ausgabe und Meßbefehle

.PRINT DC/AC/TRAN/NOISE V(4)

V(1,[M2])

VDB(n3)

VP(5,4)...

VBE(Q12)...

I(R3)

ID(M12)

P(R5)...

PAR('V(OUT)/V(IN)')

Druckt das DC/AC... Simulationsergebnis der Spannung an Knoten 4

Spannung zwischen Knoten 1 und Knoten 2

Spannung an Knoten n3 in Decibel

Phasendifferenz zwischen Knoten 5 und 3 (mit .AC)

Basis-Emitter-Spannung von Q12

Strom durch R3

Drain-Strom von MOS Transistor M12

Leistung, die in R5 anfällt

Verhältnis von Vout und Vin

.PLOT DC/AC/TRAN/NOISE Ausdruck <lo hi>...

Druckt einen Plot (mit Sternchen), lo/hi sind optionale min/max Achsenwerte

.PROBE DC/AC/TRAN/NOISE Ausdruck...

Speichert Daten in ein .out file (o.ä.) zur späteren grafischen Darstellung

Ein Beispiel für .MEASURE

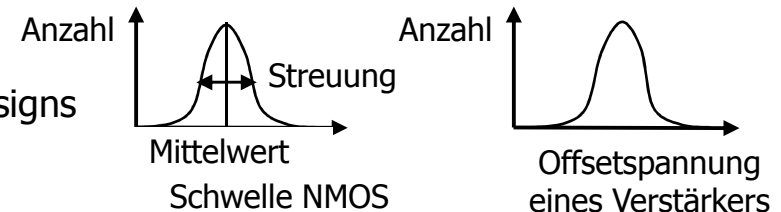
.MEAS TRAN rise TRIG V(1) VAL=.2 RISE=1 TARG V(1) VAL=.8 RISE=1

Mißt die Anstiegszeit des Signals V(1). TRIG startet die Messung, wenn V(1) steigend durch 20% geht, TARG legt den Stop bei 80% fest.

Spezielle Aufgabe: Monte-Carlo / Corner Sim.

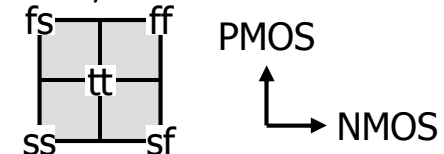
Monte-Carlo Simulation:

- Hierbei werden viele Simulationen wiederholt
- Bei jeder Simulation werden die Parameter der Bauelemente entsprechend ihrer (hoffentlich richtig bekannten) Streuungen variiert.
- Die Verteilung von Meßparametern wird analysiert
- Dies gibt wichtige Aussagen über die Robustheit eines Designs



Corner-Simulationen

- Die Hersteller garantieren alle Parameter nur innerhalb gewisser Grenzen. Ist z.B. das Gate-Oxid in einer Produktion besonders dünn, so sind die Transistoren schneller.
- Die Simulation wird daher nicht nur mit nominellen ('typical') Parametern gemacht, sondern auch mit schnellen ('fast') und langsamen ('slow').
- Dabei werden NMOS und PMOS separat variiert.
- Man simuliert also tt, ss, ff, sf, fs – Variationen
- Außerdem werden hohe / niedrige Betriebsspannung und unterschiedliche Temperaturen simuliert.



Automatische Optimierung

- SPICE kann Parameter variieren, bis eine/mehrere vorgegebene Meßgrößen optimiert sind

Weitere (in neueren Vollversionen von SPICE):

- Sensitivitäts-Analysen, Analyse von Bauteil-Belastung ('Stress'), ...

Schlußbemerkungen

- SPICE und Verwandte sind **wichtige Werkzeuge** im VLSI Design
- Syntax und Methoden der Simulatoren sind ähnlich (aber leider nicht ganz gleich)

- Wichtig ist die **Qualität der verwendeten Modelle**.
Sie müssen genau überprüft werden (Kapazitäten, Vergleich mit Messungen)

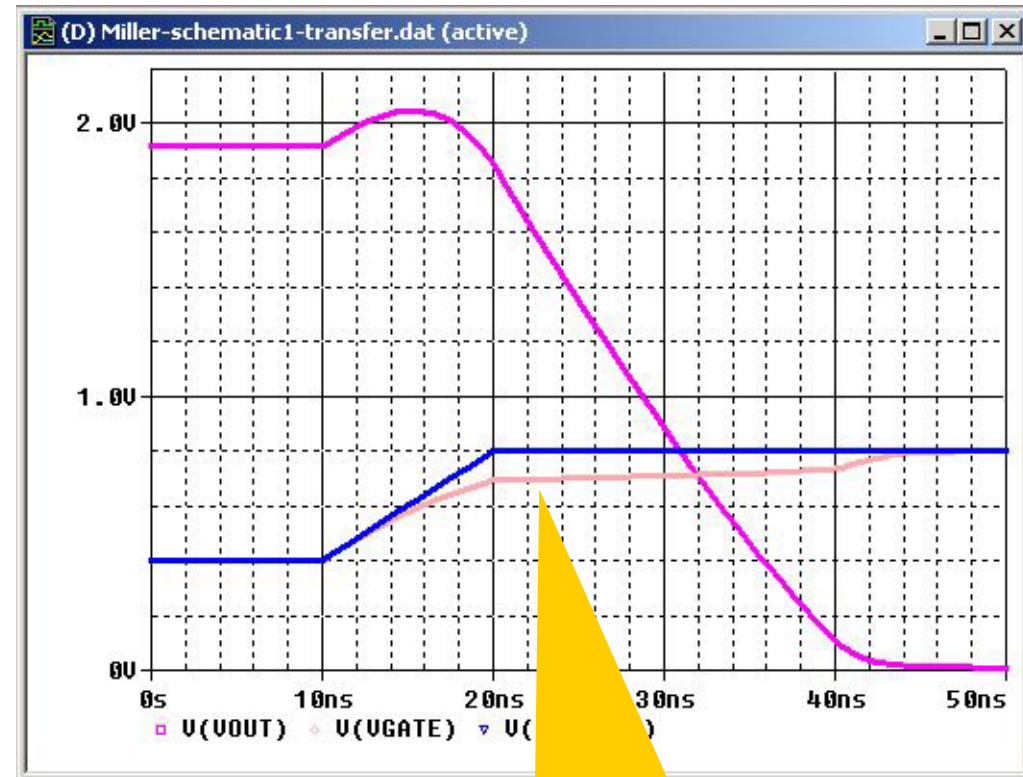
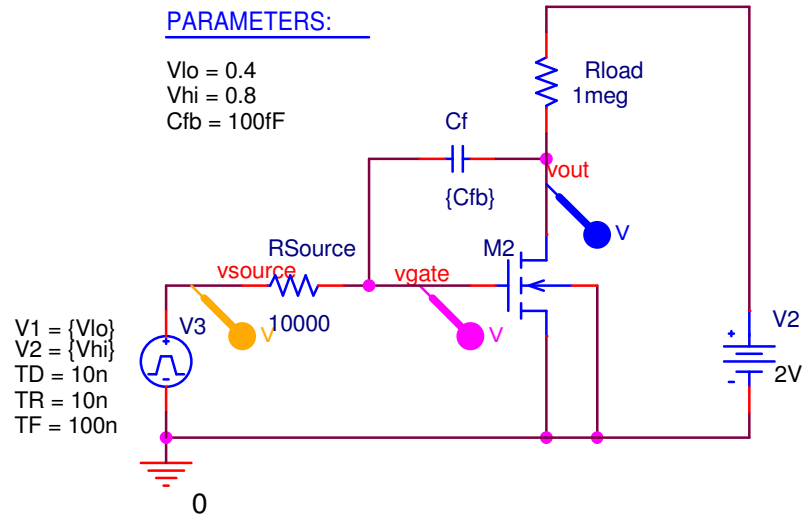
- Bei der Simulation kann man nicht genügend 'cross-checks' machen!

- Bei seltsamen Ergebnissen muß man u.a. die Parameter des Simulators verändern.
- Bei Transientensimulationen keine zu schnellen Anstiegszeiten verwenden!

- Ein Vergleich mit Rechnungen sollte, wenn möglich, gemacht werden

Noch ein Beispiel: Miller-Effekt

- Wir betrachten eine 'schlechte' Spannungsquelle (mit hohem Innenwiderstand), die einen Inverter ansteuert (aus NMOS + hochohmiger Last)



- Simulation: Cfb variieren, Vhigh variieren

Die Gatespannung steigt während der fallenden Ausgangsflanke nicht bis auf Vsource an