Lecture 5.
Netlist Grammar (brief)
& Parser Implementation

Guoyong Shi, PhD

shiguoyong@ic.sjtu.edu.cn

School of Microelectronics

Shanghai Jiao Tong University

Fall 2010

Outline

- **Netlist Grammar**
  - **Spice background**
  - **Spice devices and analyses**

- **Netlist parser implementation**
  - **Symbol table**
  - **Data structure for models & devices**
  - **Loading and solving flow in Spice**

*** This lecture can be used as a simple Spice simulator manual.**

SPICE

- Simulation Program with IC Emphasis
- First developed at UC Berkeley by Laurence Nagel (1972)
- First version contained simple diode, BJT and MOS transistor models
- Spawned many other programs with similar netlist syntax (HSPICE, PSpice, ...) as well as many other simulators (Eldo, Spectre, Sabre, ADS, ...)

SPICE History

1969-70	The CANCER Project (Ron Rohrer and Class project)
1970-72	CANCER Program (Ron Rohrer and Larry Nagel)
May 1972	SPICE I released to Public Domain
July 1975	SPICE 2A E. Cohen following Nagel’s Research
Fall 1975	SPICE 2C
Mid 1976	SPICE 2D New MOS Models
Jan 1979	SPICE 2E MOS Model Levels
Jan 1980	SPICE 2F SPICE in C, new MOS charge models
Sep 1980	SPICE 2G Pivoting in linear solver
Aug 1982	SPICE 2G
Aug 1986	SPICE 3 (Sparse package by Ken Kundert)
.....	
1990	SPICE not maintained anymore
until today	BSIM modeling continued (BSIM3/BSIM4)

The Latest spice3f4 Source

- http://embedded.eecs.berkeley.edu/pubs/downloads/spice/index.htm
- You may download all source code there.
- Try to install it in your computer.
  - Easier on Linux machine
  - Harder in cygwin
- Read the source code if you like challenges !!
- I'll introduce some of the algorithms used in SPICE in this course.

Spice Netlist Format

- ***The first line is supposed to be the title of a circuit;***
- ***The last line must be ".END".***
- **The order of the lines between the 1st and the last is arbitrary (except for the continuation lines.)**
- **A line is continued by entering a '+' (plus) in column 1 of the following line.**
- **A circuit is defined by a set of devices and their connections; a device is described by a name, nodes in the circuit, and values or models that specify the electric property of this device.**
- **A set of control lines that define the model parameters or the run controls.**
- **A name field must begin with a letter (A through Z) and cannot contain any delimiters.**
- ***Spice3 nodes names may be arbitrary character strings*.**

Spice Netlist Format

- **The ground node must be named '0'.**
- **The circuit cannot contain:**
  1. **a loop of voltage sources and/or inductors;**
  2. **a cutset of current sources and/or capacitors.**
- ***Each node in the circuit must have a DC path to the ground*.**
- **Every node must have at least two connections except for**
  - **transmission line nodes (to permit *unterminated* transmission lines) and**
  - **MOSFET substrate nodes (which have two internal connections.)**

Resistor

- **Lumped Resistor Grammar:**
  - **Rname <node> <node> <val>**
- **Examples**
  - **R1 1 2 100**
  - **RC1 12 17 1K**
- **Semiconductor Resistor Grammar:**
  - **Rname <node> <node> [<val>] <M_NAME> <L=len> <W=width> <TEMP=T>**
- **Example**
  - **Rm 3 7 RMODEL L=10u W=1u**

Resistor

- **Grammar**
  - **Rname <node> <node> [<val>] <M_NAME> <L=len> <W=width> <TEMP=T>**
- **Example (Resistor with model)**
  - **Rm 3 7 RMODEL L=10u W=1u**

- **If <val> is specified, it overrides the geometric info following the <val>**
- **If <M_NAME> is specified, then resistance is calculated from the process info in the model using the given length (L) and width (W).**

Capacitor

- **CXXXXXXX N+ N- VALUE < IC=INCOND >**
- **N+ and N- are the positive and negative element nodes, respectively. VALUE is the capacitance in Farads.**
- **The (optional) initial condition is the initial (time-zero) value of capacitor voltage (in Volts).**
- **Note that the initial conditions (if any) apply only if the *UIC option is specified on the .TRAN control line*.**
- **Examples:**
  - **CBYP 13 0 1UF**
  - **COSC 17 23 10U IC=3V**

Semiconductor Capacitors

- CXXXXXXX N1 N2 < VALUE> < MNAME > < L=LENGTH > < W=WIDTH> < IC=VAL >
- This allows for the calculation of the actual capacitance value from strictly geometric information and the specifications of the process.
- If VALUE is specified, it defines the capacitance.
- If MNAME is specified, then the capacitance is calculated from the process information in the model MNAME and the given LENGTH and WIDTH.
- If VALUE is not specified, then MNAME and LENGTH must be specified. If WIDTH is not specified, then it is taken from the default width given in the model.
- Either VALUE or MNAME, LENGTH, and WIDTH may be specified, but not both sets.
- Examples:
- CLOAD 2 10 10P
- CMOD 3 7 CMODEL L=10u W=1u

Inductor

- **LYYYYYYY N+ N- VALUE < IC=INCOND >**
- **N+ and N- are the positive and negative element nodes, respectively. VALUE is the inductance in Henries.**
- **The (optional) initial condition is the initial (time-zero) value of inductor current (in Amps) that flows from N+, through the inductor, to N-.**
- **Note that the initial conditions (if any) apply only if the *UIC option is specified on the .TRAN analysis line*.**
- **Examples:**
  - **LLINK 42 69 1UH**
  - **LSHUNT 23 51 10U IC=15.7MA**

Mutual Inductors

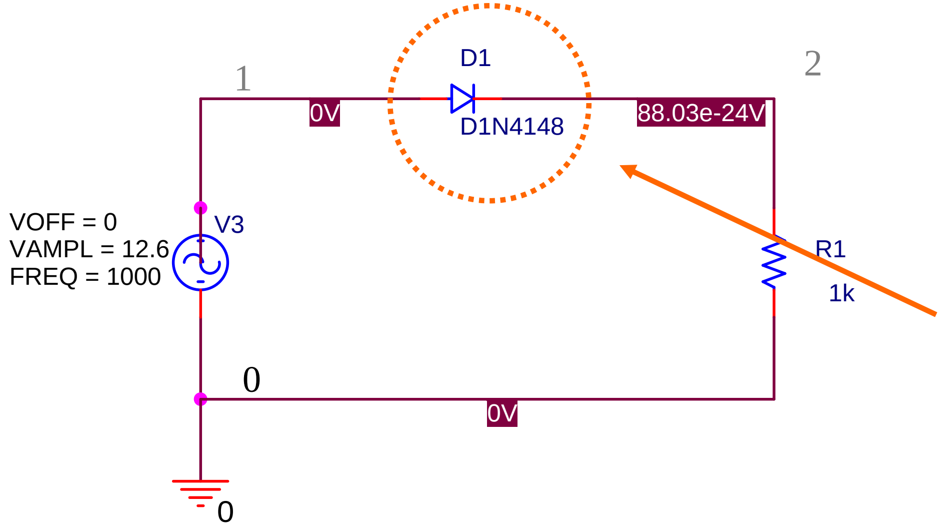
- **KXXXXXXX LYYYYYYY LZZZZZZZ VALUE**
- **LYYYYYYY and LZZZZZZZ are the names of the two coupled inductors, and VALUE is the coefficient of coupling, $0 < K \leq 1$.**
- ***Using the 'dot' convention, place a 'dot' on the first node of each inductor.***
- **Examples:**
  - **K43 LAA LBB 0.999**
  - **KXFRMR L1 L2 0.87**

Diode (D)

- DXXXXXXX N+ N- MNAME <AREA> <OFF> <IC=VD> <TEMP=T>
- Examples:
- DBRIDGE 2 10 DIODE1
  DCLMP 3 7 DMOD 3.0 IC=0.2

- MNAME is the model name,
- AREA is the area factor (default to 1.0),
- OFF indicates an (optional) starting condition on the device for dc analysis.

Diode

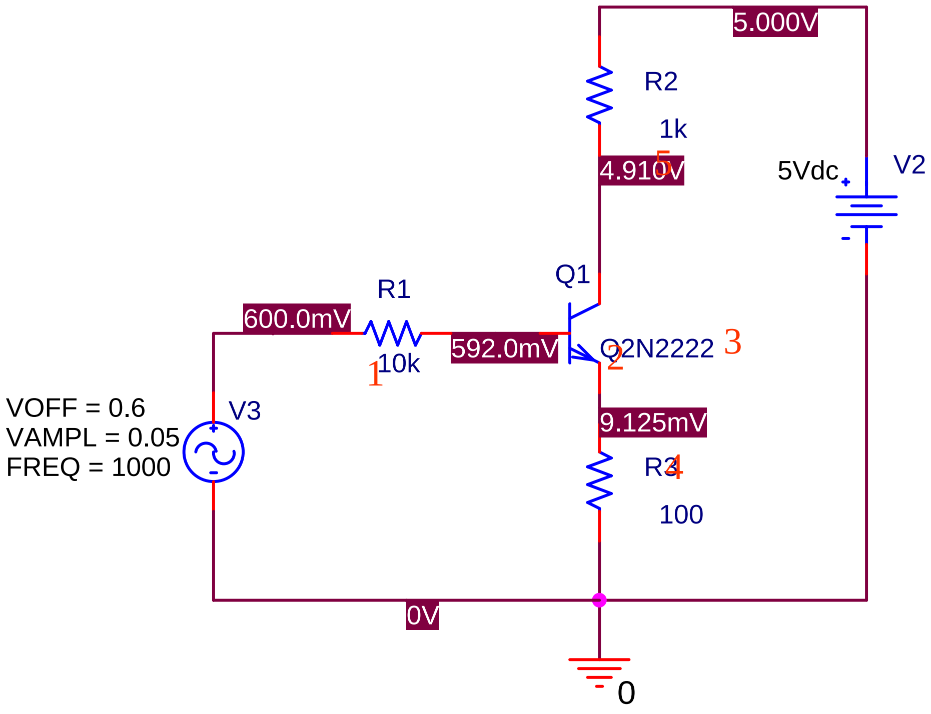


```
*DIODE01.CIR – Half-wave rectifier
.lib "nom.lib"
V1 1 0 SIN(0 12.6 1000)
D1 1 2 D1N4148
R1 2 0 1K
.TRAN 0.1M 10M 5M 0.01M
.PROBE
.END
```

BJT (Q)

- Bipolar Junction Transistors (BJTs)
- QXXXXXXX NC NB NE <NS> MNAME <AREA> <OFF> <IC=VBE, VCE> <TEMP=T>
- Examples:
- Q23   10 24 13   QMOD IC=0.6, 5.0
  Q50A 11 26   4 20 MOD1
- NC, NB, and NE are the collector, base, and emitter nodes, respectively. NS is the (optional) substrate node. If unspecified, ground is used.
- MNAME is the model name, AREA is the area factor, and OFF indicates an (optional) initial condition on the device for the DC analysis.
- If the area factor is omitted, a value of 1.0 is assumed.
- The (optional) initial condition specification using IC=VBE, VCE is intended for use with the UIC option on the .TRAN control line.
- The (optional) TEMP value is the temperature at which this device is to operate, and overrides the temperature specification on the .OPTION control line.

Transistor Circuit



```
* source ONE_TRANSISTOR
R_R1        1 2  10k
R_R2        3 5  1k
R_R3        0 4  100
V_V2        5 0 5Vdc
Q_Q1        3  2  4 Q2N2222
V_V3        1 0
+SIN 0.6 0.05 1000 0 0 0
```

JFET (J)

- **Junction Field-Effect Transistors (JFETs)**
- **JXXXXXXX ND NG NS MNAME < AREA > < OFF > < IC=VDS, VGS > < TEMP=T >**
- **Examples:**
- **J1 7 2 3 JM1 OFF**

Independent Voltage Sources

- VXXXXXXX N+ N- << DC > DC/TRAN VALUE> < AC < ACMAG < ACPHASE >>> < DISTOF1 < F1MAG < F1PHASE >>> < DISTOF2 < F2MAG < F2PHASE >>>

  Examples:
- VCC 10 0 DC 6
  VIN 13 2 0.001 AC 1 SIN(0 1 1MEG)

  0.001 is DC value. AC 1 is for AC analysis with magnitude 1.
- SIN(0 1 1MEG) is a for transient analysis.
- A source can be assigned values for DC, AC, and tran analysis.

- VMEAS 12 9
  VCARRIER 1 0 DISTOF1 0.1 -90.0
  VMODULATOR 2 0 DISTOF2 0.01

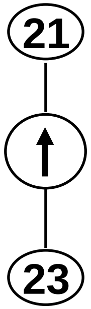
- Zero-valued voltage sources (representing short-circuits) can be used for measuring current (ammeters).

Independent Current Sources

- **General form:**

  IYYYYYYY N+ N- << DC> DC/TRAN VALUE> < AC < ACMAG < ACPHASE >>> < DISTOF1 < F1MAG > F1PHASE>>> < DISTOF2 < F2MAG < F2PHASE >>>

- **Positive current is assumed to flow from the positive (+) node, *through the source*, to the negative (-) node.**
- **A current source of positive value forces current to flow out of the N+ node, *through the source*, and into the N- node.**



- **Examples:**
  ISRC 23 21 AC 0.333 45.0 SFFM(0 1 10K 5 1K)

- **SFFM stands for *Frequency-Modulated Sinusoidal Function*.**
- **Its standard form is**

  SFFM(V0 Va fc mdi fs)
- **Its mathematical form is**

  v(t) = V0 + Va * [sin(2*pi*fc*t + mdi*sin(2*pi*fs*t))].

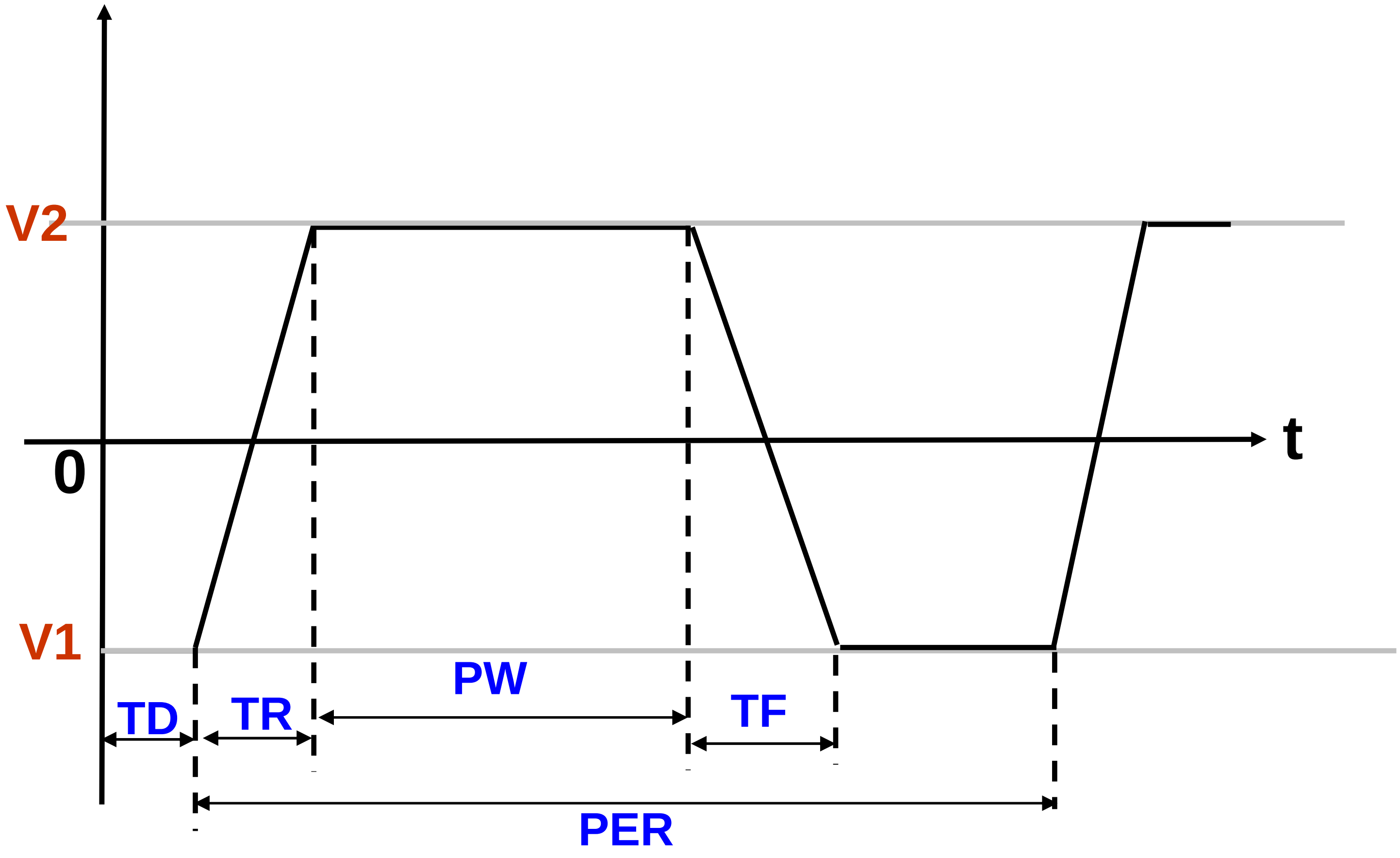
- **IIN1 1 5 AC 1 DISTOF1 DISTOF2 0.001**

Time-Dependent Independent Sources

- **Any independent source can be assigned a time-dependent value for transient analysis.**
- **If a source is assigned a time-dependent value, *the time-zero value is used for DC analysis*.**
- **There are five independent source functions:**
  - **Pulse,**
  - **Exponential,**
  - **Sinusoidal,**
  - **Piece-wise linear, and**
  - **Single-frequency FM.**
- **If parameters other than source values are omitted or set to zero, the default values shown are assumed.**

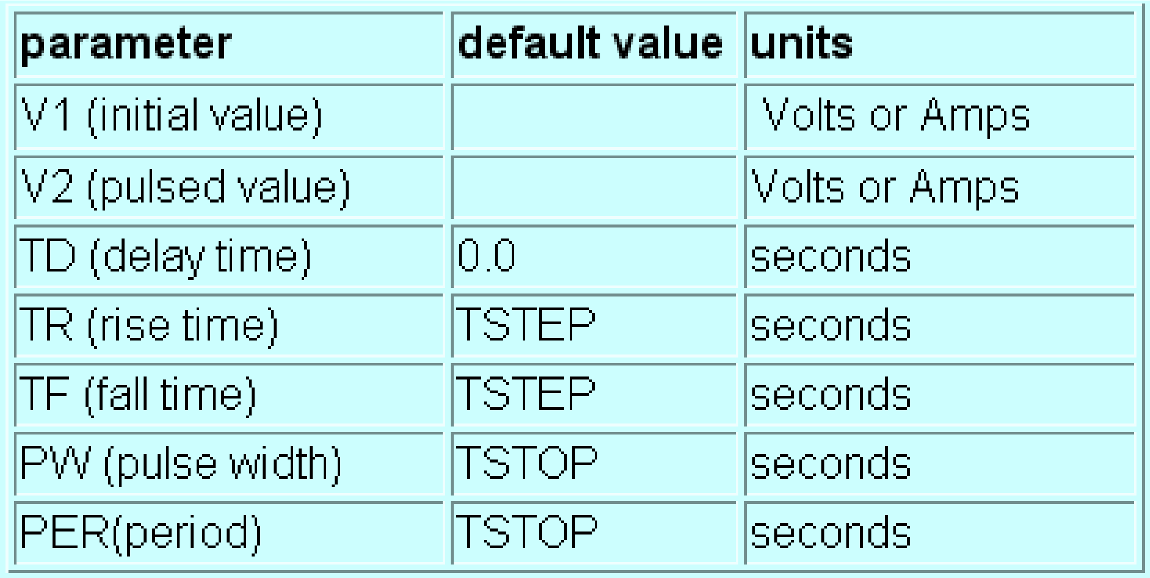
PULSE

- PULSE(V1 V2 TD TR TF PW PER)



PULSE

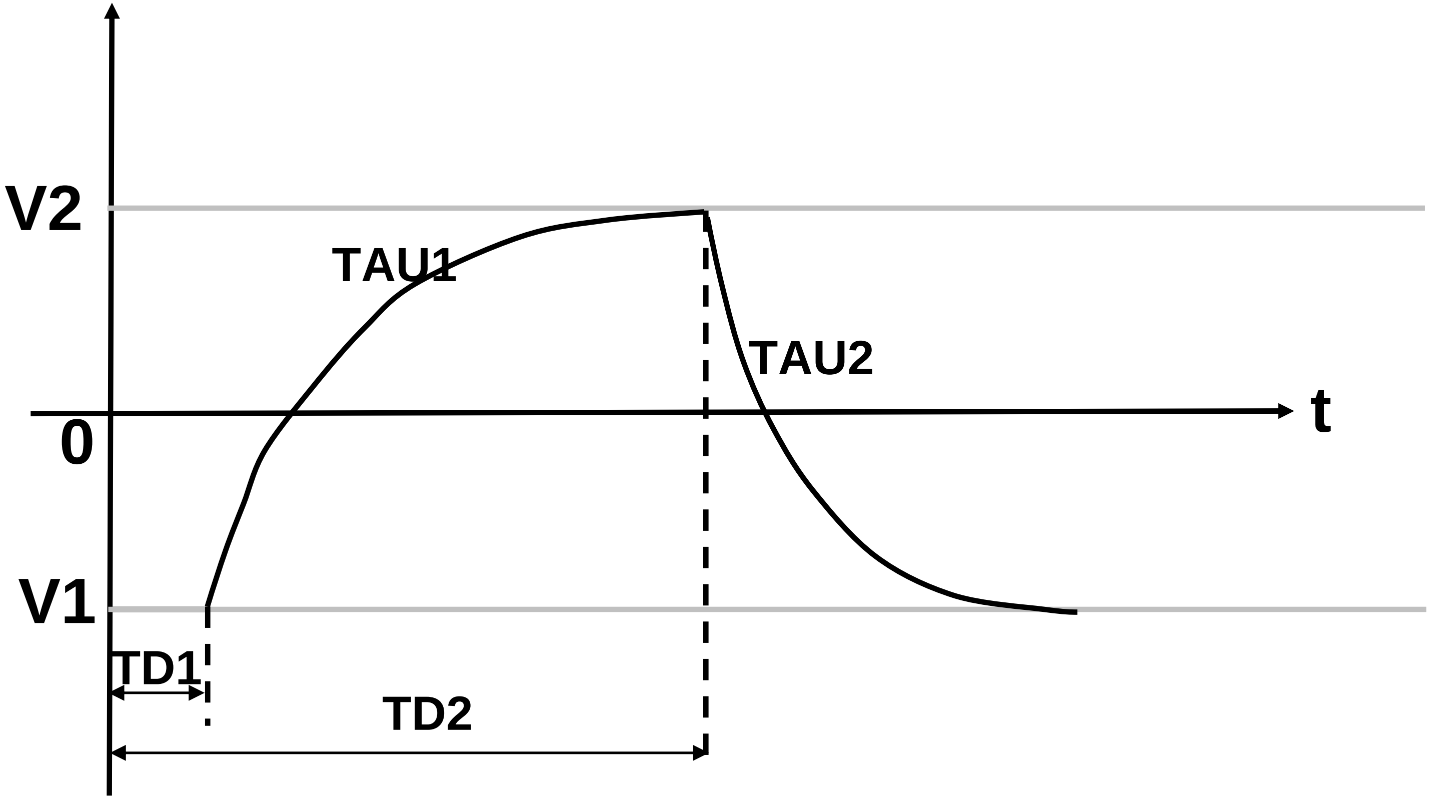
- **VIN 3 0 PULSE (-1 1 2NS 2NS 2NS 50NS 100NS)**
- **VIN 3 0 PULSE -1 1 2NS 2NS 2NS 50NS 100NS**
  - **(without parentheses)**



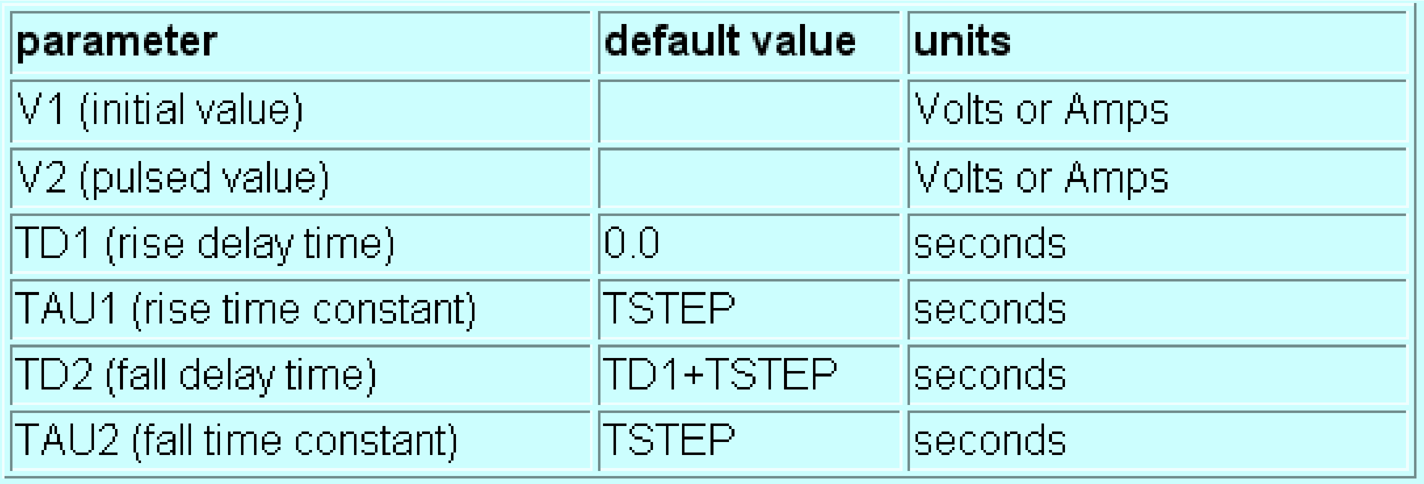
parameter	default value	units
V1 (initial value)		Volts or Amps
V2 (pulsed value)		Volts or Amps
TD (delay time)	0.0	seconds
TR (rise time)	TSTEP	seconds
TF (fall time)	TSTEP	seconds
PW (pulse width)	TSTOP	seconds
PER(period)	TSTOP	seconds

EXPONENTIAL

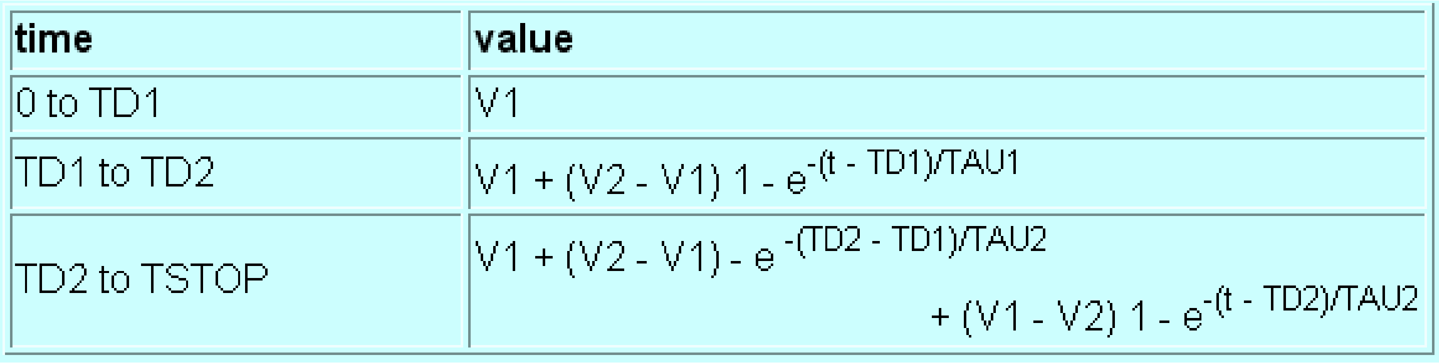
- **EXP(V1 V2 TD1 TAU1 TD2 TAU2)**
- **VIN 3 0 EXP(-4 -1 2NS 30NS 60NS 40NS)**



EXPONENTIAL



parameter	default value	units
V1 (initial value)		Volts or Amps
V2 (pulsed value)		Volts or Amps
TD1 (rise delay time)	0.0	seconds
TAU1 (rise time constant)	TSTEP	seconds
TD2 (fall delay time)	TD1+TSTEP	seconds
TAU2 (fall time constant)	TSTEP	seconds



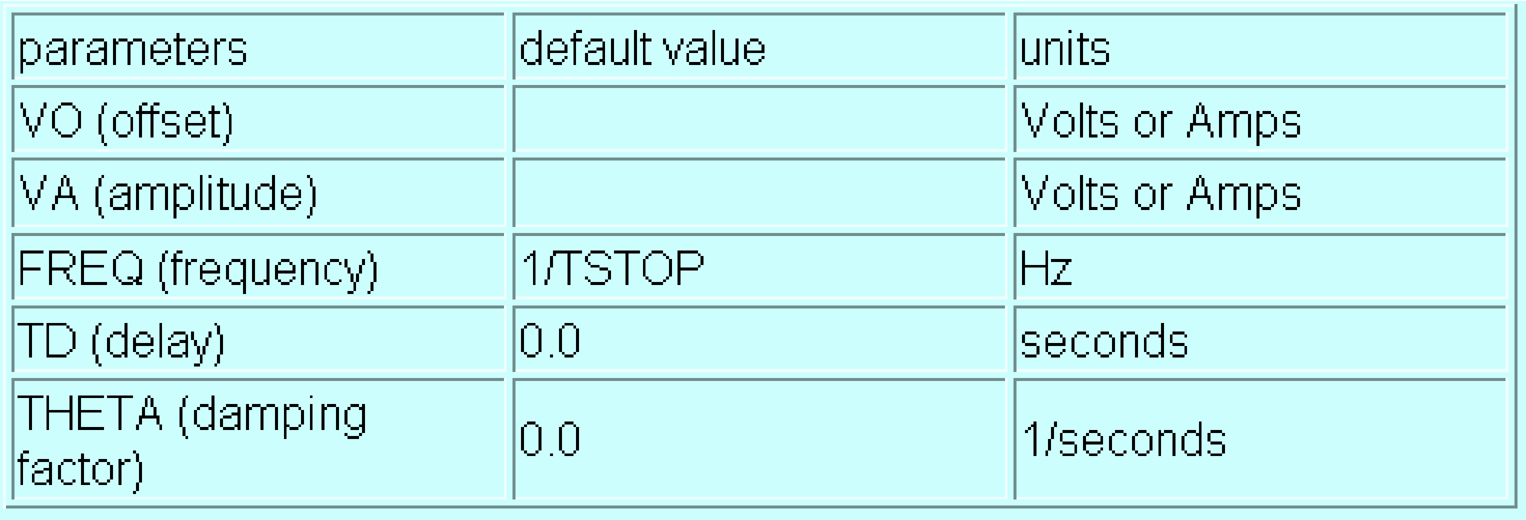
time	value
0 to TD1	V1
TD1 to TD2	$V1 + (V2 - V1)\ 1 - e^{-(t - TD1)/TAU1}$
TD2 to TSTOP	$V1 + (V2 - V1) - e^{-(TD2 - TD1)/TAU2}$ $+ (V1 - V2)\ 1 - e^{-(t - TD2)/TAU2}$

SINUSOIDAL

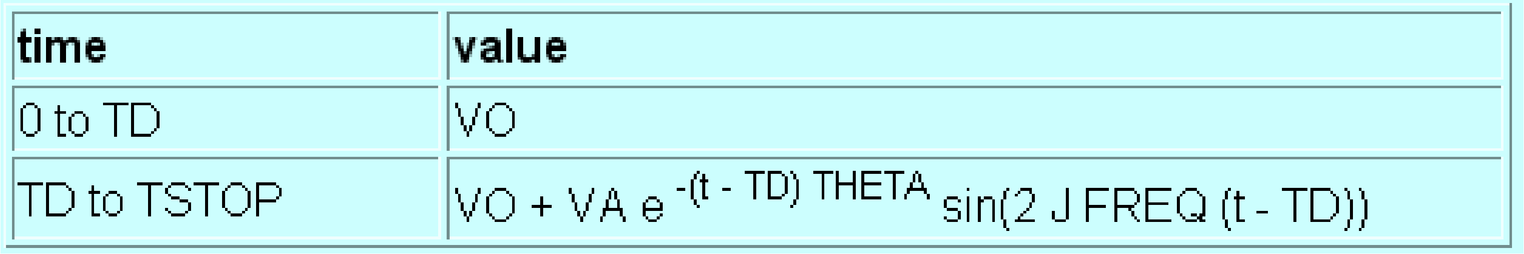
- **SIN(V0 VA FREQ TD THETA)**
- **VIN 3 0 SIN(0 1 100MEG 1NS 1E10)**

$$V_{IN} = \begin{cases} V_0, & 0 \le t < TD \\ V_0 + V_A * e^{-(t-TD)*THETA} \sin\left(2\pi f(t-TD)\right), & t \ge TD \end{cases}$$

SINUSOIDAL



parameters	default value	units
VO (offset)		Volts or Amps
VA (amplitude)		Volts or Amps
FREQ (frequency)	1/TSTOP	Hz
TD (delay)	0.0	seconds
THETA (damping factor)	0.0	1/seconds



time	**value**
0 to TD	VO
TD to TSTOP	$VO + VA\ e^{-(t - TD)\ THETA} \sin(2\ J\ FREQ\ (t - TD))$

Piecewise Linear

- General Form:

  PWL(T1 V1 [T2 V2 T3 V3 T4 V4 ...])

- Examples:

  VCLOCK 7 5 PWL(0 -7 10NS -7 11NS -3 17NS -3 18NS -7 50NS -7)

- Each pair of values (Ti, Vi) specifies that the value of the source is Vi (in Volts or Amps) at time=Ti.
- The value of the source at intermediate values of time is determined by using linear interpolation on the input values.

Linear Dependent Sources

1. **Linear Voltage-Controlled Voltage Sources**
   - **VCVS (E-Element)**
2. **Linear Current-Controlled Current Sources**
   - **CCCS (F-Element)**
3. **Linear Voltage-Controlled Current Sources**
   - **VCCS (G-Element)**
4. **Linear Current-Controlled Voltage Sources**
   - **CCVS (H-Element)**

VCVS (E)

- General form:

  EXXXXXXX N+ N- NC+ NC- VALUE

- Examples:

  E1 2 3 14 1 2.0

- N+ is the positive node, and N- is the negative node.
- NC+ and NC- are the positive and negative controlling nodes, respectively.
- VALUE is the voltage gain.

CCCS (F)

- General form:

  FXXXXXXX N+ N- VNAM VALUE

- Examples:

  F1 13 5 VSENS 5

- N+ and N- are the positive and negative nodes, respectively.
- Current flow is from the positive node, through the source, to the negative node.

- VNAM is the name of *a voltage source through which the controlling current flows*.
- The voltage source could be zero voltage, acting as an ammeter. The direction of positive controlling current flow is from the positive node, through the source, to the negative node of VNAM.
- VALUE is the current gain.

VCCS (G)

- **General form:**

  GXXXXXXX N+ N- NC+ NC- VALUE

- **Examples:**

  G1 2 0 5 0 0.1MMHO

- **N+ and N- are the nodes of the *controlled* branch.**
- **Current flows from N+, through the source, to N-.**
- **NC+ and NC- are the nodes of the *controlling* branch.**

- **VALUE is the transconductance (in mhos).**

CCVS (H)

- **General form:**

  HXXXXXXX N+ N- VNAM VALUE

- **Examples:**

  HX 5 17 VZ 0.5K

- **N+ and N- are the positive and negative nodes, respectively.**
- ***VNAM is the name of a voltage source through which the controlling current flows*.**
- **The direction of positive controlling current flow is from the positive node, through the source, to the negative node of VNAM.**
- **VALUE is the transresistance (in ohms).**

MOSFET Device

- MXXXXXXX ND NG NS NB MNAME <L=VAL> <W=VAL> <AD=VAL> < AS=VAL> <PD=VAL> <PS=VAL> <NRD=VAL> <NRS=VAL> < OFF> < IC=VDS, VGS, VBS> <TEMP=T>

- Examples:

  M1 24 2 0 20 TYPE1   (* use all default parameters *)

  M2 2 17 6 10 MODM L=5U W=2U
  M3 2 9 3 0 MOD1 L=10U W=5U AD=100P AS=100P PD=40U PS=40U

- ND, NG, NS, and NB are the drain, gate, source, and bulk (substrate) nodes, respectively.
- MNAME is the model name.
- L and W are the channel length and width, in meters.
- AD and AS are the areas of the drain and source diffusions, in 2 meters.
- Note that the suffix U specifies microns (1e-6 m) and P sq-microns (1e-12 m2).
- ......

MOSFET Device (cont'd)

- MXXXXXXX ND NG NS NB MNAME < L=VAL> < W=VAL> < AD=VAL> < AS=VAL> < PD=VAL> < PS=VAL> < NRD=VAL> < NRS=VAL> < OFF> < IC=VDS, VGS, VBS> < TEMP=T>

- If any of L, W, AD, or AS are not specified, default values are used.
- PD and PS are the perimeters of the drain and source junctions, in meters.
- NRD and NRS designate the equivalent number of squares of the drain and source diffusions; these values multiply the sheet resistance RSH specified on the .MODEL control line for an accurate representation of the parasitic series drain and source resistance of each transistor.
- PD and PS default to 0.0 while NRD and NRS to 1.0.

MOSFET Device (cont'd)

- MXXXXXXX ND NG NS NB MNAME < L=VAL> < W=VAL> < AD=VAL> < AS=VAL> < PD=VAL> < PS=VAL> < NRD=VAL> < NRS=VAL> < OFF> < IC=VDS, VGS, VBS> < TEMP=T>

- OFF indicates an (optional) initial condition on the device for DC analysis.
- The (optional) initial condition specification using IC=VDS, VGS, VBS is intended for use with the UIC option on the .TRAN control line, when a transient analysis requires specific initial conditions.
- TEMP value is the device operating temperature. It overrides the temperature specified in .OPTIONS.
- TEMP specification is ONLY valid for level 1, 2, 3, and 6 MOSFETs, not for level 4 (BSIM1) or 5 (BSIM2) devices.

.MODEL

- .MODEL MNAME MTYPE (PNAME1=PVAL1 PNAME2=PVAL2 ...)
- .MODEL MNAME MTYPE PNAME1=PVAL1 PNAME2=PVAL2 ...
  - ---- parentheses (...) can be omitted
- Example:
  - .MODEL MOD1 NPN (BF=50 IS=1E-13 VAF=50)
  - .MODEL MOD1 NPN BF=50 VAF=50 IS=1.E-12 RB=100 CJC=.5PF TF=.6NS
  - .MODEL MOD1 NMOS LEVEL=1 VTO=-2 NSUB=1.0E15 UO=550

- Many devices can use one model definition.

- MNAME is the model name
- MTYPE is one of the following 15 types:

Device Model Types

MTYPE	DESCRIPTION
R	Semiconductor resistor model
C	Semiconductor capacitor model
D	Diode model
NPN	NPN BJT model
PNP	PNP BJT model
NJF	N-channel JFET model
PJF	P-channel JFET model
NMOS	N-channel MOSFET model
PMOS	P-channel MOSFET model
NMF	N-channel MESFET model
PMF	P-channel MESFET model
SW	Voltage controlled switch
CSW	Current controlled switch
URC	Uniform distributed RC model
LTRA	Lossy transmission line model

Refer to Spice 3 Manual for model parameters and default values.

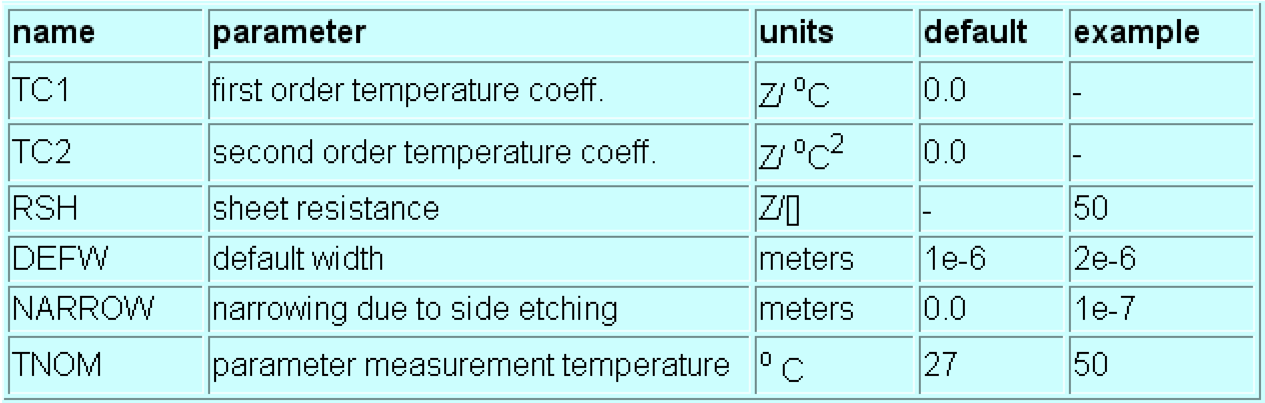
Resistor Model

- The sheet resistance (RSH) is used to determine the nominal resistance by the formula

$$R = RSH \frac{L - NARROW}{W - NARROW}$$

- DEFW is a default value for W.
- If either RSH or L is not specified, then the default resistance value of 1k Ω is used.
- After the nominal resistance is calculated, it is adjusted for temperature by the formula:

$$R(T) = R(T0)\,[1 + TC1\,(T - T0) + TC2\,(T-T0)^2\,]$$



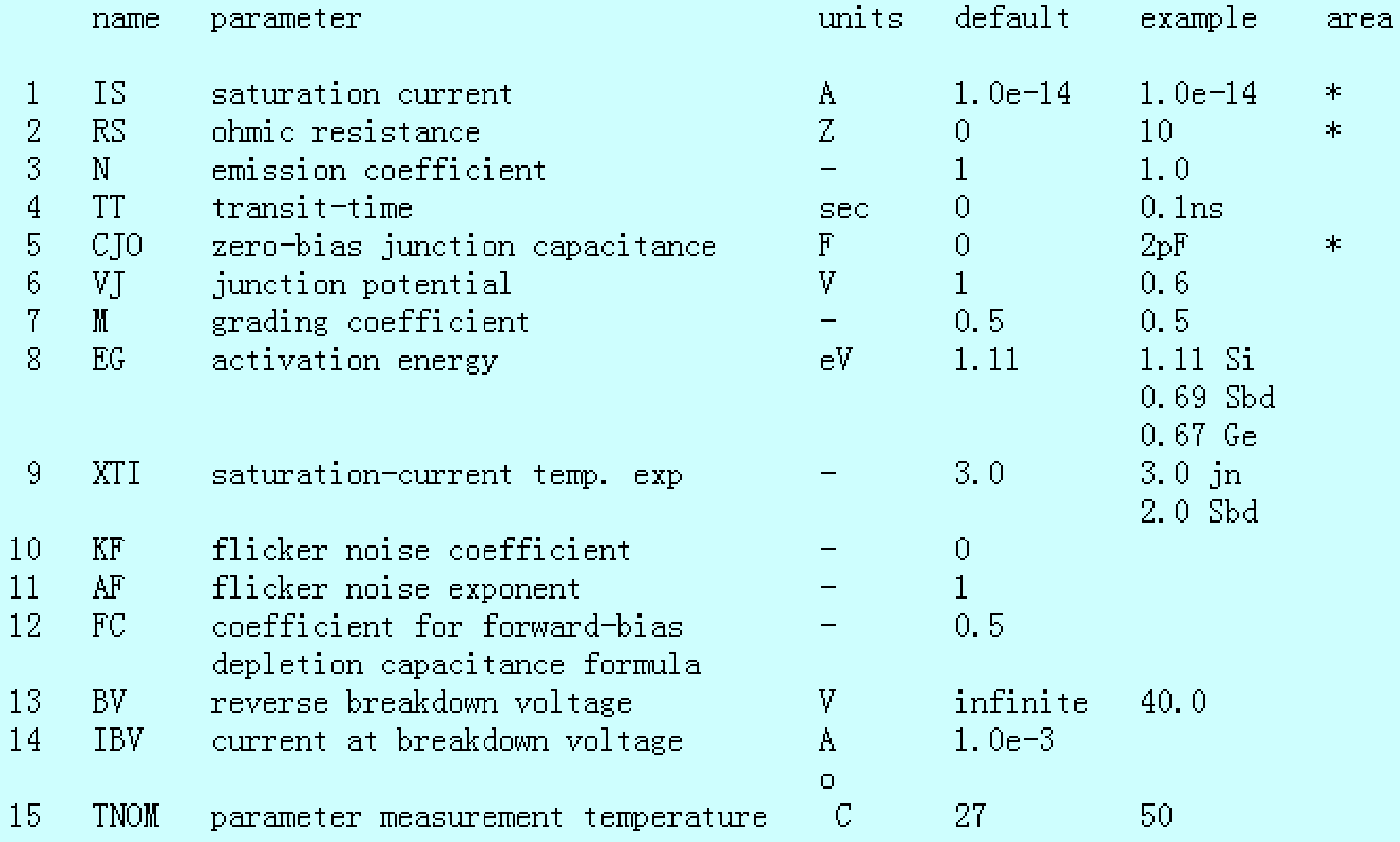
name	parameter	units	default	example
TC1	first order temperature coeff.	Z/ °C	0.0	-
TC2	second order temperature coeff.	$Z/\,{}^{o}C^2$	0.0	-
RSH	sheet resistance	Z/□	-	50
DEFW	default width	meters	1e-6	2e-6
NARROW	narrowing due to side etching	meters	0.0	1e-7
TNOM	parameter measurement temperature	° C	27	50

Diode Model (D)

.MODEL DIODE1 D

- IS (saturation current) and N (emission coeff) determines the DC characteristics.
- RS (ohmic resistance).
- TT (a transit time) models the charge storage effects.
- CJO (junction cap), VJ (junction potential), and M (grading coeff) model the nonlinear depletion layer capacitance.
- EG (activation energy) and XTI model the temperature dependence of the saturation current.
- TNOM is the nominal temperature (a circuit-wide default value given by .OPTIONS).
- Reverse breakdown voltage and current are determined by the parameters BV and IBV.

Diode Model Parameters



	name	parameter	units	default	example	area
1	IS	saturation current	A	1.0e-14	1.0e-14	*
2	RS	ohmic resistance	Z	0	10	*
3	N	emission coefficient	-	1	1.0	
4	TT	transit-time	sec	0	0.1ns	
5	CJO	zero-bias junction capacitance	F	0	2pF	*
6	VJ	junction potential	V	1	0.6	
7	M	grading coefficient	-	0.5	0.5	
8	EG	activation energy	eV	1.11	1.11 Si 0.69 Sbd 0.67 Ge	
9	XTI	saturation-current temp. exp	-	3.0	3.0 jn 2.0 Sbd	
10	KF	flicker noise coefficient	-	0		
11	AF	flicker noise exponent	-	1		
12	FC	coefficient for forward-bias depletion capacitance formula	-	0.5		
13	BV	reverse breakdown voltage	V	infinite	40.0	
14	IBV	current at breakdown voltage	A	1.0e-3		
15	TNOM	parameter measurement temperature	^{o}C	27	50	

BJT Models (NPN/PNP)

- The BJT model is an adaptation of the integral charge control model of Gummel and Poon.
- This modified Gummel-Poon model extends the original model to include several effects at high bias levels.
- The model automatically simplifies to the simpler Ebers-Moll model when certain parameters are not specified.
- The parameter names used in the modified Gummel-Poon model have been chosen to be more easily understood by the program user, and to reflect better both physical and circuit design thinking.

MOSFET Models

- **MOSFET Models (NMOS/PMOS)**

- **SPICE provides four MOSFET device models, which differ in the formulation of the I-V characteristic:**

  - **LEVEL=1 -> MOS1 (Shichman-Hodges)**
  - **LEVEL=2 -> MOS2**
  - **LEVEL=3 -> MOS3 (a semi-empirical model)**
  - **LEVEL=4 -> BSIM1**
  - **LEVEL=5 -> BSIM2**
  - **LEVEL=6 -> MOS6**

MOSFET Models

- The DC characteristics of the level 1 through level 3 MOSFETs are defined by the device parameters VTO, KP, LAMBDA, PHI and GAMMA.
- These parameters are computed by SPICE if process parameters (NSUB, TOX, ...) are given, but user-specified values always override.
- Charge storage is modeled overlap capacitances CGSO, CGDO, and CGBO which are distributed among the gate, source, drain, and bulk regions, and bottom and periphery capacitances which vary as the MJ and MJSW power of junction voltage respectively, and are determined by the parameters CBD, CBS, CJ, CJSW, MJ, MJSW and PB.
- These voltage-dependent capacitances are included only if TOX is specified in the input description and they are represented using Meyer's formulation.

cf. “Spice 3 Manual”

MOSFET Models

- A discontinuity in the MOS level 3 model with respect to the KAPPA parameter has been detected ([1]).
- The supplied fix has been implemented in Spice3f2 and later.
- Since this fix may affect parameter fitting, the option "BADMOS3" may be set to use the old implementation (see the section on simulation variables and the ".OPTIONS" line).

[1] R. Saleh and A. Yang, Editors, Simulation and Modeling, IEEE Circuits and Devices, vol. 8, no. 3, pp. 7-8 and 49, May 1992.

MOS Device Paras & Model Paras

- **MOS Level = 1**
- **Device Parameters (appearing behind a device card) :**

  TEMP; W; L; AS; AD; PS; PD; NRS; NRD; OFF; VBS; VDS; VGS; IC; L_SENS; W_SENS

  (example)

  M3 2 9 3 0 MOD1 L=10U W=5U AD=100P AS=100P PD=40U PS=40U

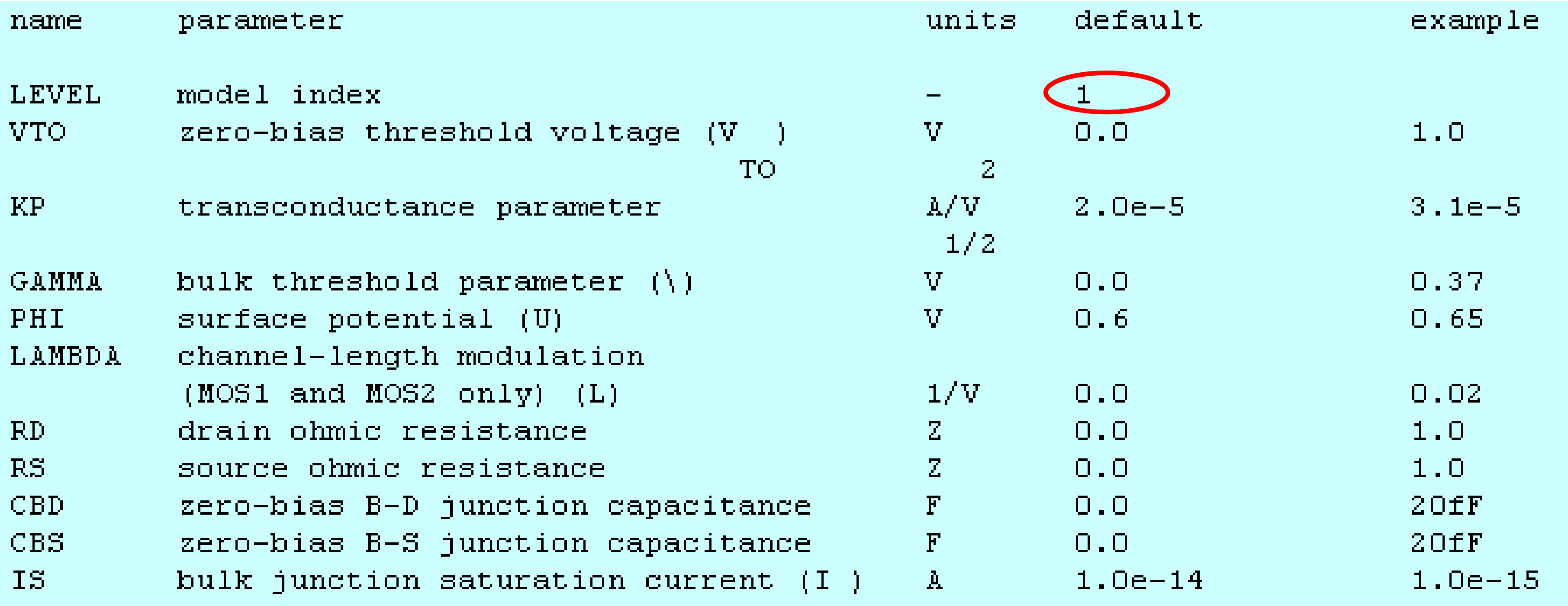
- **MOS Model Parameters (appearing behind a model card):**

  TYPE; LEVEL; VTO; KP; GAMMA; PHI; LAMBDA; RD; RS; CBD; CBS; IS; PB; CGSP; CGDO; CGBO; CJ; MJ; CJSW; MJSW; JS; TOX; LD; RSH; U0; FC; NSUB; TPG; NSS; TNOM;

  (example)

  .MODEL MOD1 NMOS VTO=-2 NSUB=1.0E15 U0=550

Sample MOS Model Parameters



name	parameter	units	default	example
LEVEL	model index	-	1	
VTO	zero-bias threshold voltage (V_{TO})	V	0.0	1.0
KP	transconductance parameter	A/V^2	2.0e-5	3.1e-5
GAMMA	bulk threshold parameter (\)	$V^{1/2}$	0.0	0.37
PHI	surface potential (U)	V	0.6	0.65
LAMBDA	channel-length modulation (MOS1 and MOS2 only) (L)	1/V	0.0	0.02
RD	drain ohmic resistance	Z	0.0	1.0
RS	source ohmic resistance	Z	0.0	1.0
CBD	zero-bias B-D junction capacitance	F	0.0	20fF
CBS	zero-bias B-S junction capacitance	F	0.0	20fF
IS	bulk junction saturation current (I)	A	1.0e-14	1.0e-15

The default model level is “level 1”.

BSIM1 & BSIM2 Models

- **The level 4 and level 5 (BSIM1 and BSIM2) parameters are all values obtained from process characterization, and can be generated automatically.**
- **J. Pierret [1] describes a means of generating a 'process' file, and the program Proc2Mod provided with SPICE3 converts this file into a sequence of BSIM1 ".MODEL" lines suitable for inclusion in a SPICE input file.**
- **Note that unlike the other models in SPICE, the BSIM model is designed for use with a process characterization system that provides all the parameters, thus there are no defaults for the parameters, and leaving one out is considered an error.**

- [1] J. R. Pierret, A MOS Parameter Extraction Program for the BSIM Model ERL Memo Nos. ERL M84/99 and M84/100, Electronics Research Laboratory University of California, Berkeley, November 1984

MOS Model Example

```
*MOSIS HPSCMOS26G 0.8um averaged SPICE LEVEL3 PARAMETERS
.MODEL  NMOS4   NMOS   LEVEL=3 ACM = 3
+ PHI =  0.6909   TOX = 1.73E-08  XJ = 0.200000U
+ TPG = 1   VTO = 0.7386   DELTA = 8.32E-01
+ LD = 1.44E-08  KP = 1.20E-04  UO = 602.1591
+ THETA = 1.28E-01 RSH = 1.54E+01  GAMMA = 0.6215
+ NSUB = 4.65E+16 NFS = 1.66E+12 VMAX = 1.85E+05
+ ETA    = 3.51E-02 KAPPA = 1.36E-01 CGDO = 2.82E-10
+ CGSO = 2.82E-10 CGBO = 4.01E-10 CJ = 2.84E-04
+ MJ =   0.9376   CJSW    = 5.76E-10 MJSW = 0.3158
+ PB = 0.7014
* Weff = Wdrawn - Delta_W
* The suggested Delta_W is 3.1800E-07
```

MOS Model Example

```
.MODEL  PMOS4  PMOS   LEVEL = 3   ACM = 3
+ PHI = 0.6909   TOX = 1.73E-08 XJ = 0.200000U
+ TPG = -1   VTO = -0.8776  DELTA = 1.02E+00
+ LD = 4.43E-09  KP = 3.30E-05  UO = 165.1182
+ THETA = 1.33E-01   RSH = 3.37E+01   GAMMA = 0.4764
+ NSUB = 2.74E+16   NFS = 2.21E+12   VMAX = 1.77E+05
+ ETA = 2.92E-02   KAPPA = 5.62E+00   CGDO = 2.91E-10
+ CGSO = 2.91E-10   CGBO = 4.04E-10   CJ = 5.51E-04
+ MJ = 0.5065   CJSW = 1.70E-10 MJSW = 0.256
+ PB = 0.8414
* Weff = Wdrawn - Delta_W
* The suggested Delta_W is 3.7420E-07
```

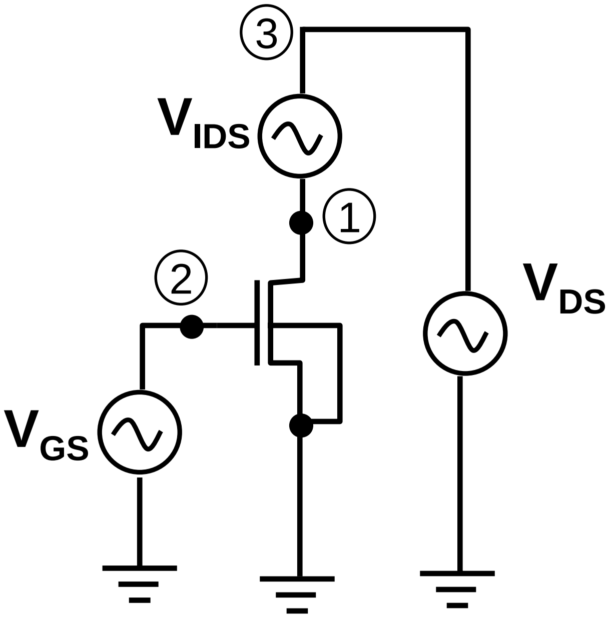
Six Levels of MOSFET Models

- **LEVEL=1 -> MOS1 (Shichman-Hodges)**
- **LEVEL=2 -> MOS2**
- **LEVEL=3 -> MOS3, a semi-empirical model**
- **LEVEL=4 -> BSIM1**
- **LEVEL=5 -> BSIM2**
- **LEVEL=6 -> MOS6**

- **The level 4 and level 5 (BSIM1 and BSIM2) parameters are all values obtained from process characterization, and can be generated automatically. (See Spice manual for more info.)**

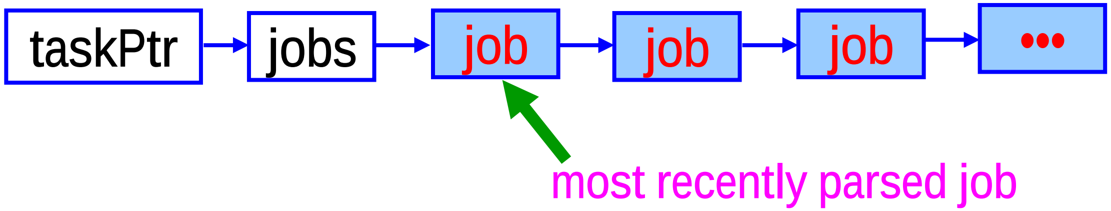
MOS Example

```
 *MOS OUTPUT CHARACTERISTICS
.OPTIONS NODE NOPAGE
VDS   3   0
VGS   2   0
M1     1   2   0   0   MOD1 L=4U W=6U AD=10P AS=10P
*VIDS MEASURES I_D.
*WE COULD HAVE USED VDS, BUT GET (–I_D)
VIDS  3  1
.MODEL MOD1 NMOS VTO=-2 NSUB=1.0E15 UO=550
.DC VDS 0 10 .5 VGS 0 5 1
.END
```



Analysis

- Analysis requests are specified as dot-commands in a netlist.
- Spice allows a number of different types of analysis, such as:
  - AC
  - DC Sweep
  - Distortion
  - Noise
  - Operating Point
  - Pole-Zero
  - Sensitivity
  - Transient
- Each analysis request is saved as an analysis job in Spice.
- All analysis jobs are managed as a linked-list in the task structure (TSKtask)



.OPTIONS

- **General form:**

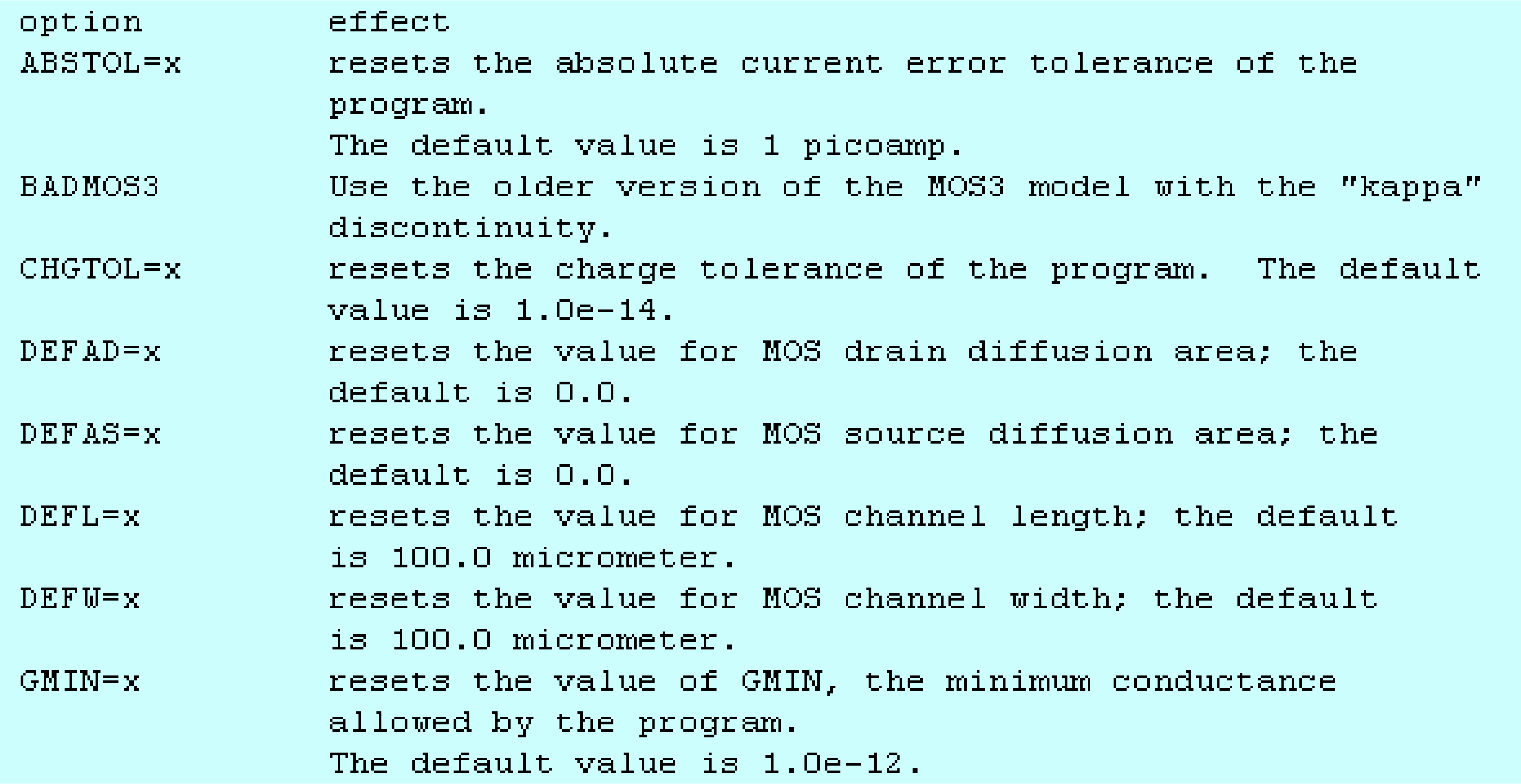
  .OPTIONS [OPT1 [OPT2 ... [OPT=OPTVAL ...]]]
- **Example:**

  .OPTIONS RELTOL=.005 TRTOL=8
- **The .OPTIONS card is used to set simulator variables.**
- **.OPTIONS parameters are used to control the accuracy, speed, or device default values.**
- **Parameters unset in .OPTIONS use default values.**
- **These parameters also can be changed via the "set" command.**

.OPTIONS

- **Parameters set in .option card are passed to the Task struct in Spice.**
- **.OPTIONS parameters are top-level parameters to be managed by the simulator.**
- **.OPTIONS RELTOL=.005 TRTOL=8**
- **RELTOL = relative error tolerance**
- **TRTOL = truncation tolerance; an estimate of the factor by which SPICE overestimates the actual truncation error (the default value is 7.0)**

Some .options Parameters



option	effect
ABSTOL=x	resets the absolute current error tolerance of the program. The default value is 1 picoamp.
BADMOS3	Use the older version of the MOS3 model with the "kappa" discontinuity.
CHGTOL=x	resets the charge tolerance of the program. The default value is 1.0e-14.
DEFAD=x	resets the value for MOS drain diffusion area; the default is 0.0.
DEFAS=x	resets the value for MOS source diffusion area; the default is 0.0.
DEFL=x	resets the value for MOS channel length; the default is 100.0 micrometer.
DEFW=x	resets the value for MOS channel width; the default is 100.0 micrometer.
GMIN=x	resets the value of GMIN, the minimum conductance allowed by the program. The default value is 1.0e-12.

See the Spice manual for more parameters.

More .options Parameters

- Other options in Spice2 emulation mode:

Option	Effect
ACCT	Print accounting and run time statistics
LIST	Print the summary listing of the input data
NOMOD	Suppresses the printout of the model parameters
NOPAGE	Suppresses page ejects
NODE	Print the node table.
OPTS	Print the option values.

.NODESET

- .NODESET is used to specify initial node voltage guesses.
- General form:
  .NODESET V(NODNUM)=VAL V(NODNUM)=VAL ...
- Examples:
  .NODESET V(12)=4.5 V(4)=2.23

- The nodeset line helps the program find the DC or initial transient solution by making a preliminary pass with the specified nodes held to the given voltages.

- The .NODESET line may be necessary for convergence on bistable or a-stable circuits. (In general, this line should not be necessary.)

.IC

- **.IC is used to set Initial Conditions for transient analysis.**
- **General form:**

  .IC V(NODNUM)=VAL V(NODNUM)=VAL ...
- **Examples:**

  .IC V(11)=5 V(4)=-5 V(2)=2.2
- **It has two different interpretations, depending on whether the UIC parameter is specified on the .TRAN control line.**
- **1) When the UIC parameter is specified on the .TRAN line, then the node voltages specified on the .IC control line are used to compute the capacitor, diode, BJT, JFET, and MOSFET initial conditions. This is equivalent to specifying the IC=... parameter on each device line, but is much more convenient. The IC=... parameter can still be specified and takes precedence over the .IC values. Since no dc bias (initial transient) solution is computed before the transient analysis, one should take care to specify all dc source voltages on the .IC control line if they are to be used to compute device initial conditions.**
- **2) *When the UIC parameter is not specified on the .TRAN* control line, the dc bias (initial transient) solution is computed before the transient analysis. In this case, the node voltages specified on the .IC control line is forced to the desired initial values during the bias solution. During transient analysis, the constraint on these node voltages is removed. This is the preferred method since it allows SPICE to compute a consistent dc solution.**

.AC

- **Small-Signal AC Analysis**

- **General form:**

```
.AC DEC ND FSTART FSTOP
.AC OCT NO FSTART FSTOP
.AC LIN   NP FSTART FSTOP
```

- **Examples:**

-
  ```
  .AC DEC 10   1    10K
  .AC DEC 10   1K 100MEG
  .AC LIN  100  1    100HZ
  ```

- **DEC stands for decade variation, and ND = # points per decade.**
- **OCT stands for octave variation, and NO = # points per octave.**
- **LIN   stands for linear variation,   and NP = # points.**
- **FSTART is the starting frequency, and FSTOP is the final frequency.**

- **Note that in order for this analysis to be meaningful, at least one independent source must have been specified with an AC value.**

.DC

- **DC Transfer Curve Calculation**
- **General form:**

```
.DC SRC1 START1 STOP1 INCR1 [SRC2 START2 STOP2 INCR2]
```

- **Examples:**

```
.DC VIN  0.25 5.0 0.25
.DC VDS 0     10    .5     VGS  0   5      1
.DC VCE 0     10    .25    IB       0   10U  1U
```

- **Defines DC transfer source and sweep limits (with capacitors open and inductors shorted).**
- **Both independent voltage and current source are allowed.**
- **The 2nd source (SRC2) is optional.**
- **In the case of two sources, the first source is swept over its range for each value of the second source. This option can be useful for obtaining semiconductor device output characteristics.**

DC Transfer Analysis

- **Grammar** sweep → ↓ fix one value

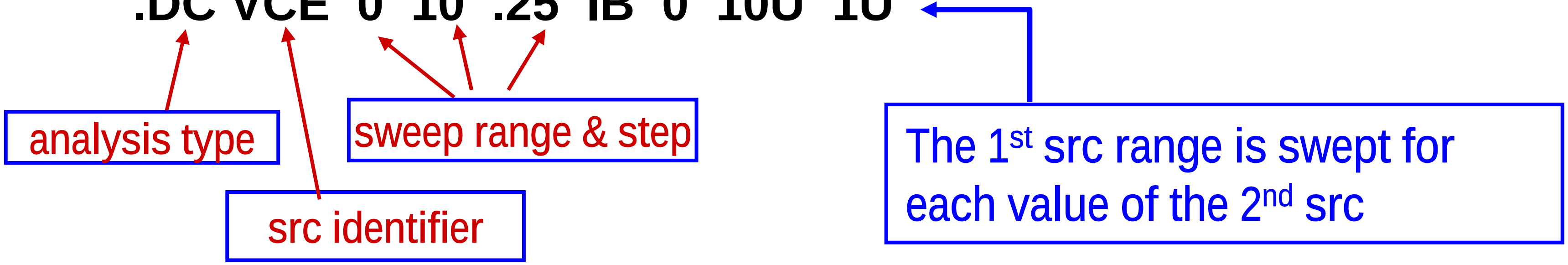
  .DC SRC VSTART VSTOP VINC [SRC2 VSTART2 VSTOP2 VINC2]

- **Examples**

  .DC VIN 0.25 5.0 0.25

  .DC VDS 0 10 .5 VGS 0 5 1

  .DC VCE 0 10 .25 IB 0 10U 1U



analysis type

src identifier

sweep range & step

The 1st src range is swept for each value of the 2nd src

.DISTO

- **Distortion Analysis**
- **General form:**

  .DISTO DEC ND FSTART FSTOP < F2OVERF1 >
  .DISTO OCT NO FSTART FSTOP < F2OVERF1 >
  .DISTO LIN NP FSTART FSTOP < F2OVERF1 >

- **Examples:**

  .DISTO DEC 10 1kHz 100Mhz
  .DISTO DEC 10 1kHz 100Mhz 0.9

- **The Disto line does a small-signal distortion analysis of the circuit.**
- **A multi-dimensional Volterra series analysis is done using multi-dimensional Taylor series to represent the nonlinearities at the operating point. Terms of up to third order are used in the series expansions.**

.NOISE

- Noise Analysis
- General form:

  .NOISE V(OUTPUT <,REF>) SRC (DEC | LIN | OCT) PTS FSTART FSTOP
  + < PTS_PER_SUMMARY >

- Examples:
- .NOISE V(5) VIN DEC 10 1kHZ 100Mhz
  .NOISE V(5,3) V1 OCT 8 1.0 1.0e6 1

- OUTPUT is the node at which the total output noise is desired;
- if REF is specified, then the noise voltage V(OUTPUT) - V(REF) is calculated. By default, REF is assumed to be ground.
- SRC is the name of an independent source to which input noise is referred.
- PTS, FSTART and FSTOP are .AC type parameters that specify the frequency range over which plots are desired.
- PTS_PER_SUMMARY is an optional integer; if specified, the noise contributions of each noise generator is produced every PTS_PER_SUMMARY frequency points.

.NOISE

- General form:

  .NOISE V(OUTPUT <,REF>) SRC (DEC | LIN | OCT)
  PTS FSTART FSTOP
  + < PTS_PER_SUMMARY >

- The .NOISE control line produces two plots –
  - one for the Noise Spectral Density curves and
  - one for the total Integrated Noise over the specified frequency range.
- All noise voltages/currents are in squared units (V^2 /Hz and A^2 /Hz for spectral density, V^2 and A^2 for integrated noise).

.OP

- .OP
- This command directs SPICE to determine the DC operating point of the circuit with inductors shorted and capacitors opened.
- Note: a DC analysis is automatically performed
  - prior to a transient analysis to determine the transient initial conditions, and
  - prior to an AC small-signal, Noise, and Pole-Zero analysis to determine the linearized, small-signal models for non- linear devices.

.PZ

- Pole-Zero Analysis
- General form:

```
.PZ NODE1 NODE2 NODE3 NODE4 CUR POL
.PZ NODE1 NODE2 NODE3 NODE4 CUR ZER
.PZ NODE1 NODE2 NODE3 NODE4 CUR PZ
.PZ NODE1 NODE2 NODE3 NODE4 VOL POL
.PZ NODE1 NODE2 NODE3 NODE4 VOL ZER
.PZ NODE1 NODE2 NODE3 NODE4 VOL PZ
```

- Examples:

```
.PZ 1 0 3 0 CUR POL
.PZ 2 3 5 0 VOL ZER
.PZ 4 1 4 1 CUR PZ
```

.PZ

- **General form:**

  .PZ NODE1 NODE2 NODE3 NODE4 CUR POL
  .PZ NODE1 NODE2 NODE3 NODE4 CUR ZER
  .PZ NODE1 NODE2 NODE3 NODE4 CUR PZ
  .PZ NODE1 NODE2 NODE3 NODE4 VOL POL
  .PZ NODE1 NODE2 NODE3 NODE4 VOL ZER
  .PZ NODE1 NODE2 NODE3 NODE4 VOL PZ

- **CUR stands for a transfer function of the type:**
  (output voltage) / (input current)
- **VOL stands for a transfer function of the type:**
  (output voltage) / (input voltage)
- **POL stands for pole analysis only,**
- **ZER for zero analysis only, and**
- **PZ for both.**
- **This feature is provided mainly because if there is a nonconvergence in finding poles or zeros, then, at least the other can be found.**
- **NODE1 and NODE2 are the two input nodes, and**
- **NODE3 and NODE4 are the two output nodes.**

.SENS

- DC or Small-Signal AC Sensitivity Analysis
- General form:

```
.SENS OUTVAR
.SENS OUTVAR AC DEC ND FSTART FSTOP
.SENS OUTVAR AC OCT NO FSTART FSTOP
.SENS OUTVAR AC LIN NP FSTART FSTOP
```

- Examples:

```
.SENS V(1,OUT)
.SENS V(OUT) AC DEC 10 100 100k
.SENS I(VTEST)
```

.SENS

- Calculates the sensitivity of OUTVAR to all non-zero device parameters.
- OUTVAR is a circuit variable (node voltage or voltage-source branch current).

- The form

  ```
  .SENS OUTVAR
  ```

  calculates sensitivity of the DC operating-point value of OUTVAR.
- The forms

  ```
  .SENS OUTVAR AC DEC ND FSTART FSTOP
  .SENS OUTVAR AC OCT NO FSTART FSTOP
  .SENS OUTVAR AC LIN   NP FSTART FSTOP
  ```

  calculates sensitivity of the AC values of OUTVAR.

- The parameters listed for AC sensitivity are the same as in an AC analysis.
- The output values are in dimensions of change in output per unit change of input (as opposed to percent change in output or per percent change of input).

.TRAN

- Transient Analysis
- General form:

  .TRAN TSTEP TSTOP < TSTART < TMAX >>
- Examples:

  .TRAN 1NS 100NS
  .TRAN 1NS 1000NS 500NS
  .TRAN 10NS 1US

- TSTEP is the printing or plotting increment for line- printer output. For use with the post-processor, TSTEP is the suggested computing increment.
- TSTOP is the final time, and TSTART is the initial time. The default of TSTART is zero.
- The transient analysis always begins at time zero. In the interval [0, TSTART], the circuit is analyzed (to reach a steady state), but no outputs are stored. In the interval [TSTART, TSTOP], the circuit is analyzed and outputs are stored.
- TMAX is the maximum step-size; for default, the program chooses either TSTEP or (TSTOP-TSTART) / 50.0, whichever is smaller.
- TMAX is useful when one wishes to guarantee a computing interval which is smaller than the printer increment, TSTEP.

.TRAN

- **Transient Analysis**
- **General form: .TRAN TSTEP TSTOP < TSTART < TMAX >>**
- **Examples:**
- **.TRAN 1NS 100NS**
  .TRAN 1NS 1000NS 500NS
  .TRAN 10NS 1US
- **UIC (use initial conditions) is an optional keyword which indicates that the user does not want SPICE to solve for the quiescent operating point before beginning the tran- sient analysis. If this keyword is specified, SPICE uses the values specified using IC=... on the various elements as the initial transient condition and proceeds with the analysis. If the .IC control line has been specified, then the node voltages on the .IC line are used to compute the initial conditions for the devices. Look at the description on the .IC control line for its interpretation when UIC is not specified.**

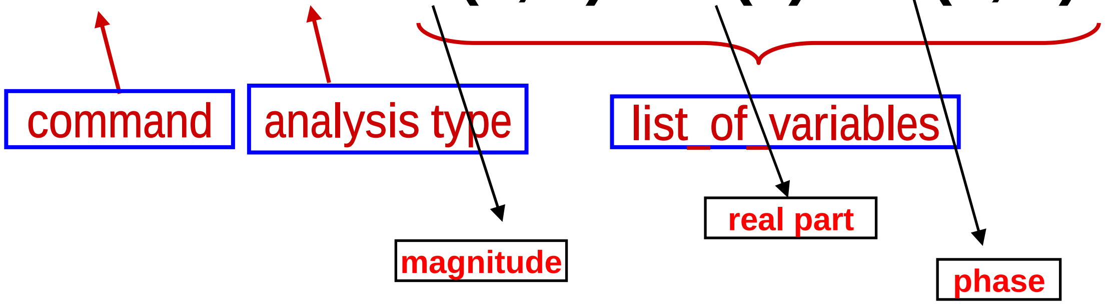
.PRINT

- **Grammar**

```
.PRINT  PRTYPE  OV1
```

- **Examples**

```
.PRINT  TRAN V(4)  I(VIN)
.PRINT  DC  V(2)  I(VSRC)  V(23, 17)
.PRINT  AC  VM(4, 2)  VR(7)  VP(8, 3)
```



command

analysis type

list_of_variables

magnitude

real part

phase

.PRINT

- .PRINT PRTYPE OV1 < OV2 ... OV8 >
- Examples:
- .PRINT TRAN V(4) I(VIN)
  .PRINT DC V(2) I(VSRC) V(23, 17)
  .PRINT AC VM(4, 2) VR(7) VP(8, 3)

- Tabular listing of one to eight output variables.
- PRTYPE is the type of the analysis (DC, AC, TRAN, NOISE, or DISTO).
- V(N1<,N2>) specifies the voltage difference between nodes N1 and N2. If N2 (and the preceding comma) is omitted, ground (0) is assumed. See the print command in the previous section for more details.
- The following five additional values can be accessed for the AC analysis by replacing the "V" in V(N1,N2) with:
- VR - real part
  VI - imaginary part
  VM - magnitude
  VP - phase
  VDB - 20 log10(magnitude)

.PRINT

- .PRINT PRTYPE OV1 < OV2 ... OV8 >

- I(VXXXXXXX) specifies the current flowing in the independent voltage source named VXXXXXXX.
- Positive current flows from the positive node, through the source, to the negative node.
- For the AC analysis, replacements for the letter I may be made in the same way as for voltage outputs.

- There is no limit on the number of .PRINT lines for each type of analysis.

Parser Implementation

Parser Implementation

- **The goal of a parser is to extract necessary information from a netlist.**
- **The extracted circuit element information is stored in a symbol table.**
- **The information stored in the symbol table will be used to check the symbol uniqueness, element validity, etc.**
- **While parsing, a list of device models and lists of model instances are created as well.**

SPICE Symbol Table

- **Data Structure**

```
typedef struct sINPtables{
    struct INPtab **INPsymtab;    /* symbol table for element names
& model names, etc. */
    struct INPnTab **INPtermsymtab;  /* symbol table for nodes */
    int INPsize;                /* size of symbol table */
    int INPtermsize;        /* size of node table */
    GENERIC *defAmod;    /* default model for A-elements */
    GENERIC *defBmod;    /* default model for B-elements */
    GENERIC *defCmod;
    … … …
    GENERIC *defZmod;     /* default model for Z-elements */
} INPtables;
```

<INPsymtab> & <INPtermsymtab>

```
/* For symbols like element names and model names */
struct INPtab {
    char *t_ent;
    struct INPtab *t_next;
};

/* For symbols like terminal nodes */
struct INPnTab {
    char *t_ent;
    GENERIC* t_node;
    struct INPnTab *t_next;
};
```

Symbol Table

- **Symbol Table is built as a class**
- **The Symbol Table class has the following methods:**
  - **initialize_table()**
  - **make_terminal()**
  - **insert_term()**
  - **remove_term()**
  - **insert_gnd ()**
  - **insert_entry()**
  - **remove_entry()**
  - **destroy_table()**
- **A hash function is implemented in Spice3 for quickly locating a table entry.**

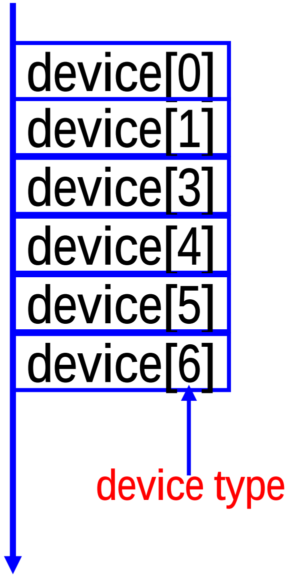
The Hash Function in Spice3

- **Spice3 implemented a hash function by summing the ASCII codes of the given string (mod table size).**

```
static int
hash(char *name,  int tsize)
{
        char *s;
        register int i = 0;
        /* s points to the name string */
        for (s = name; *s; s++)
                i += *s;     /* add up all ASCII codes */
        return (i % tsize);
}
```

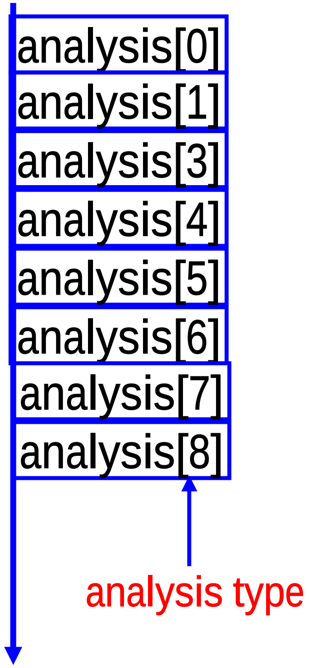
Simulator Construction

- **Two main categories of data structures**
  - **Device -> Model -> Instance**
  - **Analysis -> Job**



device[0]
device[1]
device[3]
device[4]
device[5]
device[6]

device type

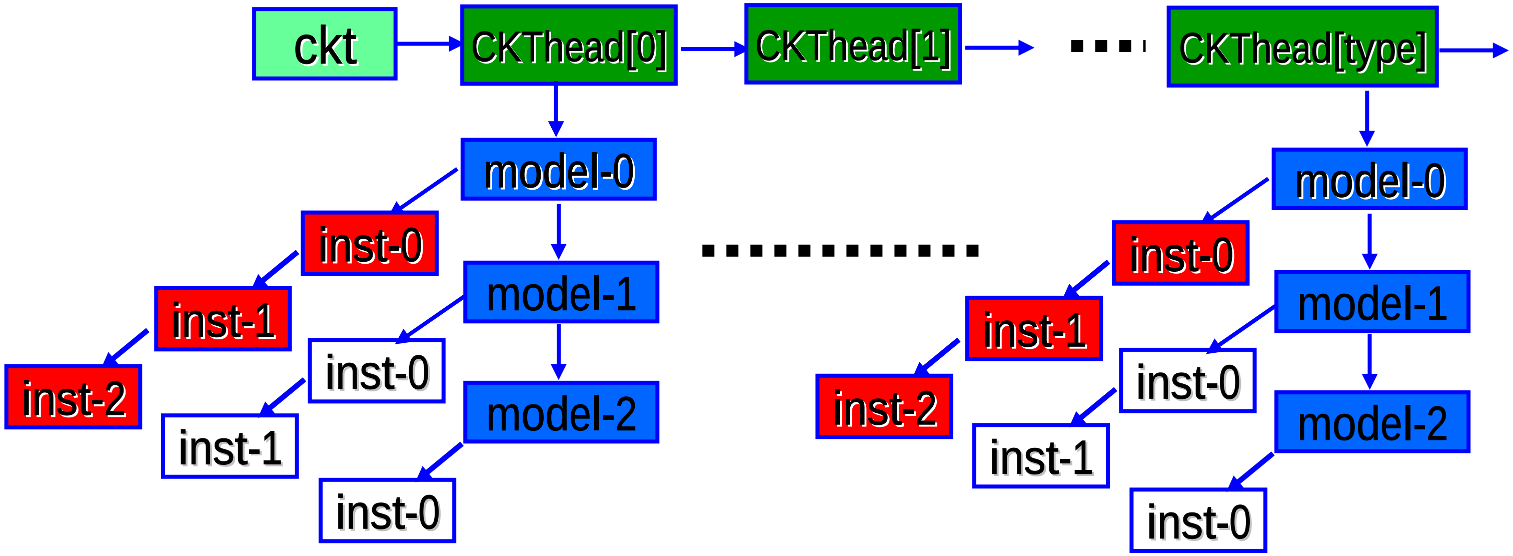


analysis[0]
analysis[1]
analysis[3]
analysis[4]
analysis[5]
analysis[6]
analysis[7]
analysis[8]

analysis type

Devices and Models

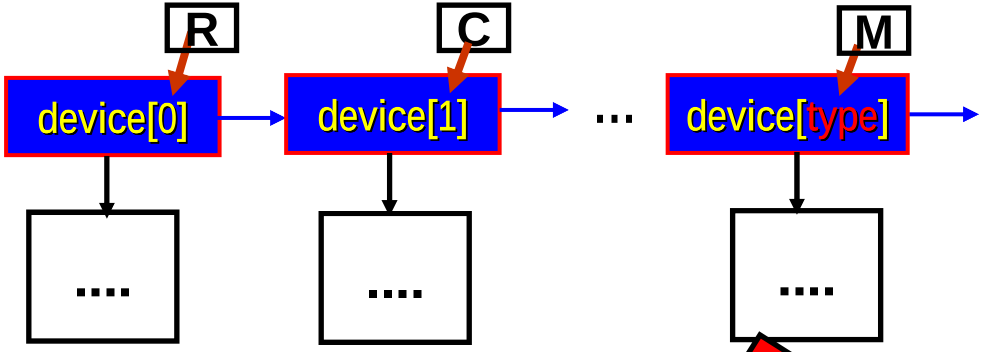
- One netlist would have a list of devices (identified by their types)
- Each type of device would have a list of models
- Each device model would have a list of instances



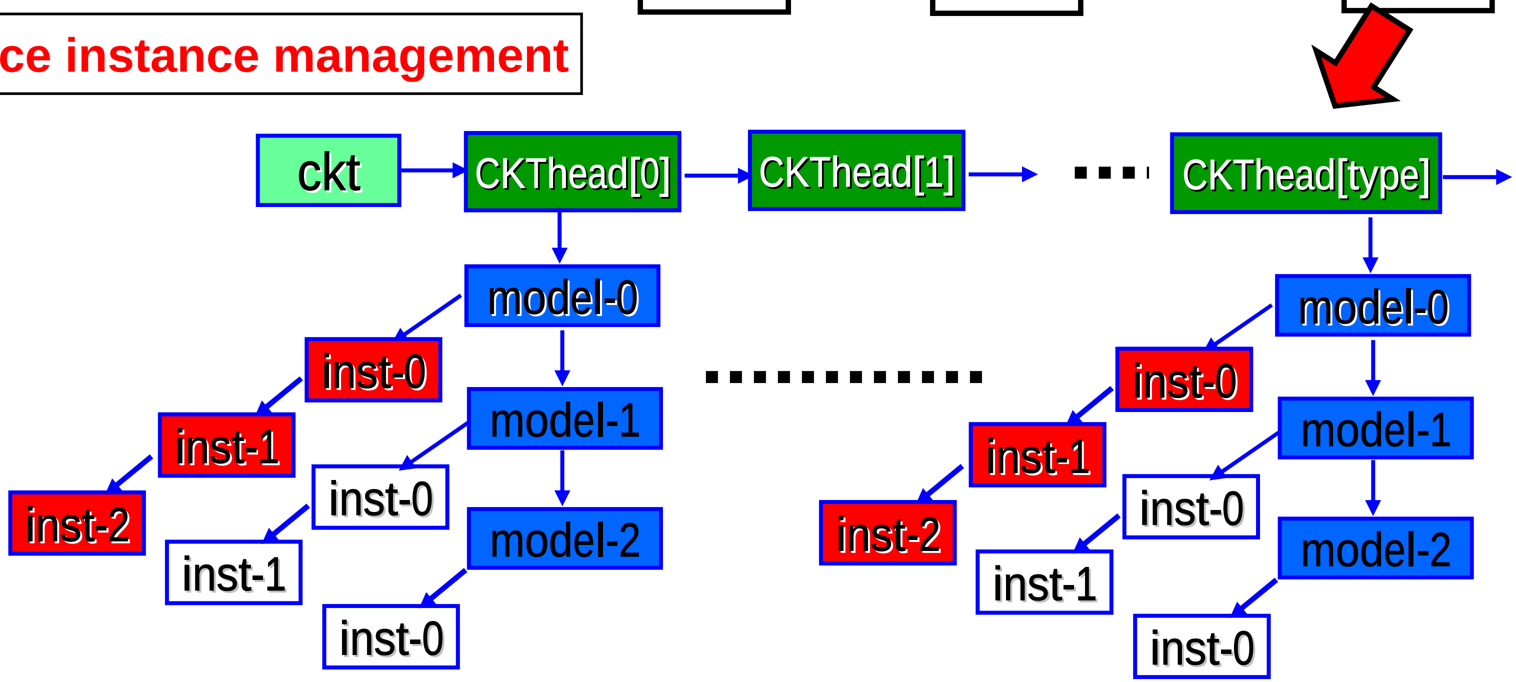
* This data structure will be used during circuit loading!

Device instance management in Spice

Device instances (parameters / routines) are managed in [type]s:

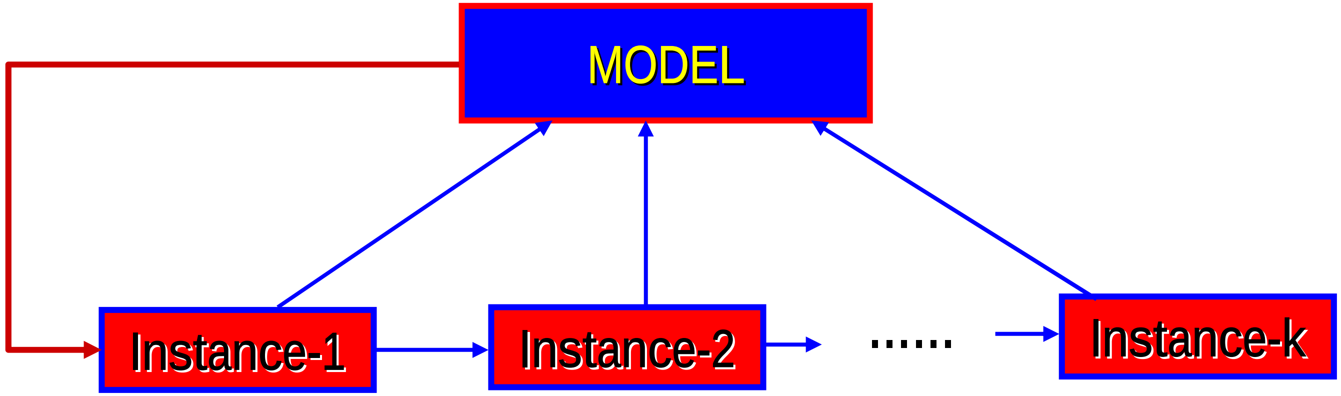


Device instance management



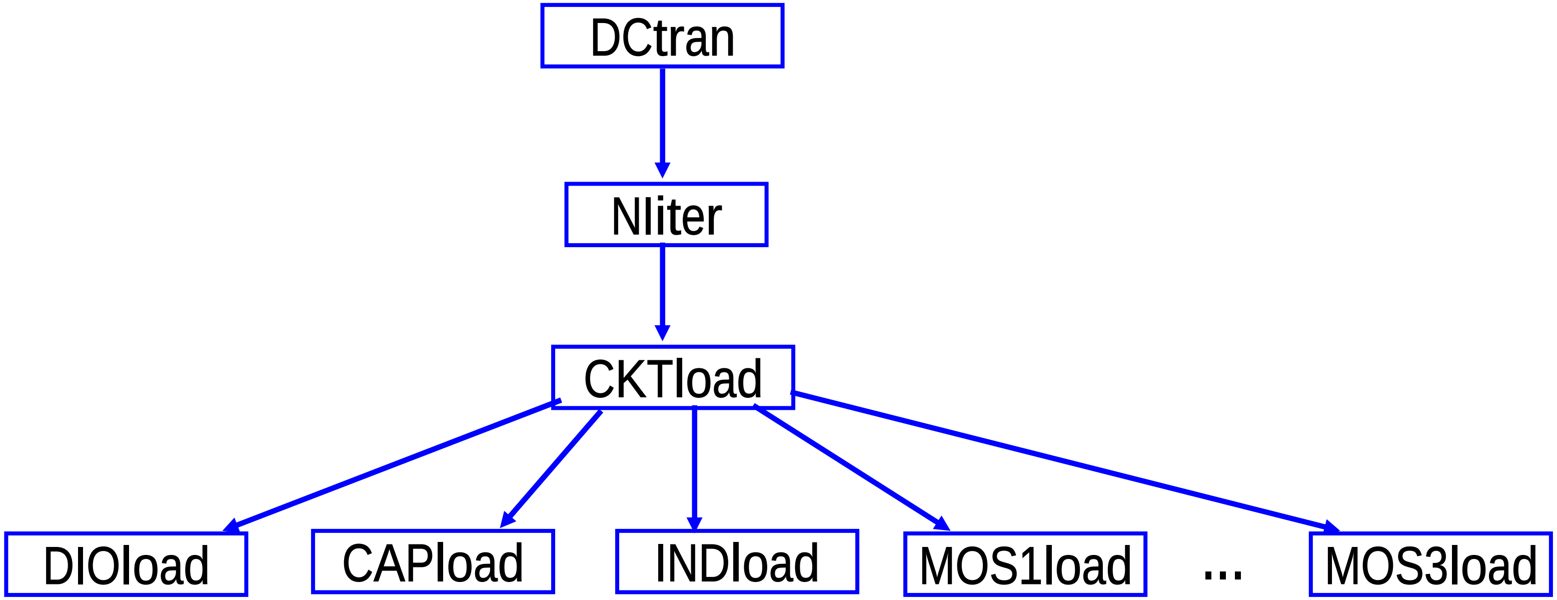
Model and Instances

- **The total num of device types is defined by the simulator** (in Spice3f4 this is defined in bconf.c)
- **Each model could have multiple device instances.**



Parameters for each device instance are set by INPpName() ([src/lib/inp/inppname.c])

Spice Transient Flow



DCtran

NIiter

CKTload

DIOload

CAPload

INDload

MOS1load

…

MOS3load

The TRAN Simulation Flow

- **TRAN is carried out in Spice by the *DCtran* function.**
- **For each TRAN point, *NIiter* (numerical integration) is called.**
- **Inside a for(...) loop in the *NIiter* function, the *CKTload* function is called which loads all devices for the new updated source.**
- **The loop in *NIiter* continues until convergence is reached.**

Simulation Complexity

- **Loading complexity – Loading all device stamps to the MNA matrix and RHS for many times to get one analysis done.**

- **Solving complexity – Solving the MNA linear system for many times**

Circuit Loading

- Circuit is loaded in different ways depending on the analysis types.
- For example, the MNA matrices are different for DC analysis and AC analysis.
- Each device (Res, Cap, ...) has its own load function (spice3).
- For complex devices such as MOS with high level models (BSIM3/4), an optimized loading function greatly speeds up simulation time.

Loading complexity

- Assume using MOS3 model.
- [MOS3load] will be invoked several hundreds to thousands of times by other functions.
- The total no. of calls to MOS3load depends on the no. of time points and the nonlinear iteration counts at each time in the simulation.

Solving complexity

- Other than the device model and instance calculation, the sparse matrix computation in SPICE3 is fairly time consuming.
- The linear algebra sparse matrix package used in SPICE3 was implemented by Dr. Kenneth Kundert around 1986 (when he was a PhD student at Berkeley.)

- K. Kundert, Sparse Matrix Techniques, in Circuit Analysis, Simulation and Design, Albert Ruehli (Ed.), North-Holland, 1986.

Table Extension

Available in HSPICE

```
*DIODE CIRCUIT
V1   1  0  DC  15V
R1  1  2  5K
R2  2  0  5K
R3  2  3  10K
R4  3  0  10K
GDIODE 3  4  TABLE  { V(3, 4) } = (0  0)  (0.1  0.13E-11)
+ (0.2  1.8E-11)  (0.3  24.1E-11)  (0.4  0.31E-8)  (0.5  4.31E-8)
+ (0.6  58.7E-8)  (0.7  7.8E-6)
R5  4  0  10K
.DC  V1  15  15  1
.PRINT  DC  I(R5)
.END
```

G: voltage controlled current source (VCCS)

self-controlled in this case

Freq Extension

Available in HSPICE

```
*FILTER CHARACTERISTIC
VIN  1  0  AC  1  0
R1  1  0  1K
EFILTER  2  0  FREQ  {V(1,0)} = (1.0K, -14, 107)  (1.9K, -9.6, 90)  (2.5K, -5.9, 72)
+ (4.0K, -3.3, 55)  (6.3K, -1.6, 39)  (10K, -0.7, 26)  (15.8K, -0.3, 17)
+ (25K, -0.1, 11)  (40K, -0.05, 7)  (63K, -0.02, 4)  (100K, -0.008, 3)
R2  2  0  1K
.AC DEC 5  1000  1.0E5
.PROBE  V(2)  V(1)
.END
```

(FREQ, MAG_DB, PHASE_DEG)

E: voltage controlled
voltage source (VCVS)

Behavioral Model

```
*VOLTAGE MULTIPLIER
V1  1  0  PWL(0  0  1MS  5V  3MS  –5V  5MS  5V  6MS  0)
.PARAM K = 0.4
V2  2  0  SIN(0  5  250  0  0  0)
*MULTIPLIER MODEL
Em  3  0 VALUE = {K*V(1,0)*V(2,0)}
R0  3  0  100
.TRAN  0.02MS   6MS
.PROBE
.END
```

Features of Modern Simulators (1)

- **Multiple-physics models:**
  - **Electrical (digital, analog, RF, microwave)**
  - **Electromagnetic**
  - **Thermal**
  - **Mechanical**
  - **Fluidic; ...**
- **A variety of analysis**
  - **Time-domain**
  - **Frequency-domain**
  - **Sensitivity**
  - **Noise**
  - **Harmonic Balance**
  - **Monte Carlo; ...**

Features of Modern Simulators (2)

- A variety of simulation algorithms
  - Linear solvers
  - Nonlinear solvers
  - Field solvers
  - Model order reduction
  - Symbolic algorithms
  - Automatic differentiation; ...

No-turn-in Exercise

- Try to download and compile the spice3f4 source code.
- Read the parser source code in spice 3f4.
- Attempt to design your circuit simulator:
  - What compiler tools you choose
  - Design a symbol table for your parser
  - Details on how the circuit information (devices/models/analyses) will be stored

- You don’t have to turn in this exercise!

Acknowledgement

- SPICE Developers at University of California, Berkeley
  - Wayne A. Christopher
  - Thomas L. Quarles
  - ...