SPICE:

User's Guide and Reference

July 2, 2007

Edition 1.3

Michael B. Steer

A manual created to support the development of fREEDA™ (http://www.freeda.org).

This document was prepared using LaTeX.
Diagrams were prepared using idraw, xfig and gnuplot.

Contents

List of Figures

List of Tables

Preface

This book was begun to develop a multi-version of a SPICE manual to guide the development of the multi-physics simulator fREEDA^TM, see http://www.freeda.org. The various version os Spice are not fully compatible which presents challenges in trying to make a new simulator compatible with Spice.

The contributions of Paul Franzon to Chapters 2 and 3 are gratefully acknowledged.

Michael Steer
North Carolina State University
email: mbs@ncsu.edu

Chapter 1

Introduction

1.1 Introduction

SPICE is a general-purpose circuit simulation program for nonlinear `DC`, nonlinear transient, and small-signal `AC` analyses. Circuits may contain resistors, capacitors, inductors, mutual inductors, independent voltage and current sources, dependent sources, transmission lines, switches, and the five most common semiconductor devices: diodes, BJTs, JFETs, MESFETs, and MOSFETs. SPICE was developed at the University of California at Berkeley and after many years of effort culminated in the landmark SPICE2G6 version. This was the last FORTRAN language version of SPICE distributed by UC Berkeley and its syntax and analysis options have become a standard for SPICE-like simulators. With few exceptions, all commercial versions and University versions of SPICE are upwards compatible with SPICE2G6 in that they support the complete syntax and analyses of SPICE2G6. Since SPICE2G6 was released SPICE3 was developed at UC Berkeley initially as a C language equivalent of SPICE2G6. The new capabilities of SPICE3 include pole-zero analysis, and new transistor models for MESFETs and for short and narrow channel MOSFETs as well improved numerical methods. Many commercial versions of SPICE are based directly on SPICE3. However there is a group of commercial SPICE-like simulators that have significant advances over SPICE2G6 and SPICE3 in the areas of enhanced input syntax, improved convergence, better device models and more analysis types. Many of these enhanced SPICE programs were completely rewritten and not ports of the Berkeley software. As can be expected, the effort put into these commercial programs is reflected in their price. The first SPICE version for personal computers was the commerical program PSPICE by MicroSim corporation. PSPICEnow has the largest customer base of all commerical SPICE programs. Consequently the syntax of PSPICE has become a second "standard". The PSPICE syntax is upwards compatible to the SPICE2G6 syntax. However, there are some incompatibilities between the SPICE3 and PSPICE syntaxes as PSPICE was released before SPICE3 became available. The effect of this development is that all SPICE simulators (including commercial programs) accept a SPICE2G6 netlist but perhaps not a SPICE3 netlist. Conflicts with SPICE3 generally exist in the naming of additional elements and in the use of new models.

1.2 How to use this book

This manual was used as a guide in developing the fREEDA simulator (see http://www.freeda.org). If you are generally unfamiliar with how to use SPICE, or are not familiar with all of its features then Chapter 2 and Chapter 3 are provided to get you started. The aim in Chapter 2 is just to help you write, run, and understand your first SPICE file. In Chapter 3, each of the major types of analyses SPICE can do for you are introduced, by example. In contrast, Chapter **??** is intended for those wishing to understand how SPICE works internally. Chapter 5 describes in the format of the SPICE input file or netlist. Part II (Chapters 6 and 7) describe the syntax of the SPICE language and the predefined expressions provided within it. Part III

summarizes the SPICE syntax, statements and elements in a quick look-up form suitable for the experienced user. Chapter 9 presents more elaborate SPICE examples. Also provided is a quick reference guide to SPICE's error messages and their meaning (Appendix E).

1.3 What SPICE Does

Many different types of analyses are supported by different versionss of SPICE. Most versions allow all of the analysis types of SPICE2G6 plus a few additional analyses. One of the exceptions is the distortion analysis which proved to be unreliable in SPICE2G6. The table below identifies the analyses that are common to virtually all SPICE programs and the extended analyses by the SPICE3 and PSPICE versions included in this book.

COMMON SPICE ANALYSES	
.AC	`AC` Analysis
.DC	`DC` Analysis
.FOUR	Fourier Analysis
.NOISE	Small-Signal Noise Analysis
.OP	Operating Point Analysis
.SENS	Sensitivity Analysis
.TRAN	Transient Analysis
ANALYSES SPECIFIC TO SPICE2G6	
.DISTO	Small-Signal Distortion Analysis
ANALYSES SPECIFIC TO SPICE3	
.DISTO	Small-Signal Distortion Analysis
.PZ	Pole-Zero Analysis
ANALYSES SPECIFIC TO PSPICE	
.TF	Transfer Function Specification PSPICE Only
.MC	Monte Carlo Analysis (PSPICE only)
.SAVEBIAS	Save Bias Conditions
.STEP	Parameteric Analysis
.WCASE	Sensistivity and Worst Case Analysis

1.4 justspice versions

This book is a manual for five versions of SPICE: SPICE2G6, SPICE3, PSPICEand HSPICE. For these the input syntax, and models are described. Particularly emphasis is given to SPICE3 and PSPICE as these are the most widley used SPICE versions. For these the graphical user interface is also described.

The syntax, analysis types, and elements of SPICE2G6form a common denominator with the capabilities of SPICE3and PSPICEbeing extensions. Adhering to the SPICE2G6 syntax ensures maximum portability of SPICE netlists. Major restrictions of this syntax compared to commercial versions include using integers to designate nodes. The SPICE3syntax is just a small extension of the SPICE2G6 syntax and is fully upwards compatible from SPICE2G6. The PSPICEsyntax is a considerable enhancement over the SPICE2G6syntax. Highlights of the enhanced syntax are that node names are allowed which greatly increases the readibility of the netlist, the use of symbolic expressions in place of numeric values, passing parameters to subcircuits, and many more analysis types. The PSPICE syntax has become a second "standard" syntax.

Part II of this book serves as a combined user and reference manual while Part III is a condensed reference manual aimed at the experiendenced user needing to check syntax. Descriptions of statements and elements are based on the SPICE2G6 syntax with the SPICE3 and PSPICE extensions clearly identified.

1.5 Documentation Conventions

In this manual the general forms of statements and elements use the following conventions to identify the type of input required:

1. Actual characters that must be typed by the user are in a typewriter font.
2. Input that must be replaced by a word or a numeric value is italicized.
3. Optional input is enclosed between square brackets " [ ] ".
4. Input that can be optionally repeated is followed by a string of dots "...".
5. As in SPICE input syntax a line is continued when a plus sign "+" appears in the first character position of the continued line.

As example the general form of resistor is

R*name* N_1 N_2
+ *ResistorValue* IC=V_R]

Here the first character on the first line is R which indicates that this line describes a resistor element. The full name of the resistor is R*name* where *name* can be replaced by any alphanumeric character string that uniquely identifies the element. Thus "R1", "Rgate" and "ROP_AMP_16_2" are names of resistors. It should be noted that SPICE does not distinguish between upper and lower case characters. *ResistorValue* must be replaced by the numeric value of the resistor possibly including a scale factor. Thus 1MEG, 1E6, and 1000000. The complete SPICE input syntax is described in Chapter 5. With the exception of the line continuation indicated by the leading + sign a SPICE element or statement must be fully contained on a single line.

Chapter 2

Getting Started

In this chapter, we simulate a small circuit in order to introduce you to SPICE. We describe the input file, or "circuit" file, showing you the generic structure of the file, and giving a number of examples. Though in each example we describe what is shown, we do not list all the options and variations for each item described. The reader is referred to the reference sections (chapters 6 and 7) for that. We then show you how to run SPICE and discuss the different features of the output file.

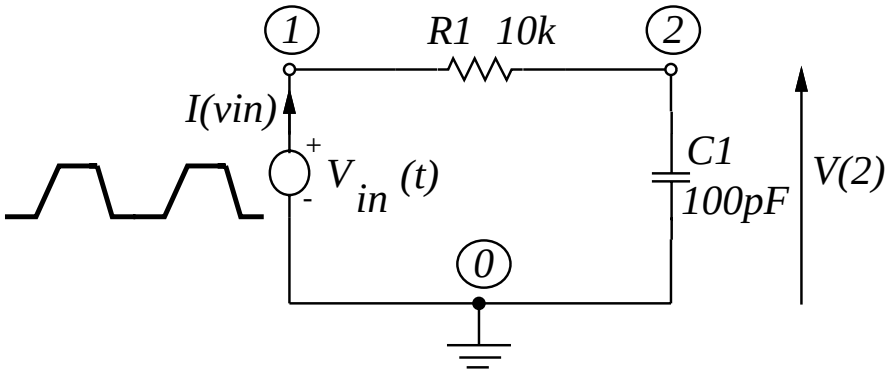
2.1 The Input File

A typical input file, and a schematic of the circuit and input waveform it is simulating, is shown in Figure 2.1. The input file is created with a text editor and is typically named something like 'test.cir'. The file is made up of five types of lines:

- A *title line*, up to 80 characters long, placed at the start of the file.
- An `.end` statement at the end of the file. This statement can be safely omitted in many simulators but its usage is recommended for compatibility purposes.
- Any number of *comment lines*, each starting with '*', can be placed anywhere after the title and before the end.
- Any number of *element lines* that describe the circuit to be simulated. The basic syntax of the element line is

  name node node ... value

  where *name* is the name that you assign to the element. The first character in the name identifies the type of circuit element being described, e.g. '`R`' for a resistor. From one to seven characters must then be added to the name to identify it uniquely, e.g. '`R1`', or '`Rpulldn`'. Numbers are usually used. The *node*'s identify the nodes in the circuit to which the terminals or 'leads' of the circuit element are connected. For example, one terminal of the capacitor '`C1`' is connected to the node numbered '`0`', which must be used for the ground (or common reference) node, and the other terminal is connected to node number '`2`'. In SPICE each circuit node is identified by a unique number. *value* describes the value(s) needed to describe the element.
- Any number of *control statement lines* that specify what type of circuit analysis is to be performed and how the results are to be reported.




```
Simple RC Network                                   Title Line
*
* This is a comment                                 Comments Lines
*
R1 1 2 500                                          Element Lines
C1 2 0 5p
vin 1 0 pulse (0 1.5 4ns 3ns 5ns 2ns 17ns)
*
.tran 500ps 34ns                                    Control Statement Lines
.print tran v(1) v(2) i(vin)
*
.end                                                End Line
```

Figure 2.1: Example circuit and corresponding input file.

The elements and control statement lines can be written in any order, even intermixed.

The first two element lines describe a 1 kΩ resistor and a 5 fF capacitor. Almost-standard metric prefixes are used in SPICE, the prefix abbreviation, the full metric name, and the represented scale factors being as follows:

Spice Prefix	Metric equivalent	Scale
F	femto	10^{-15}
P	pico	10^{-12}
N	nano	10^{-9}
U	micro	10^{-6}
M	milli	10^{-3}
K	kilo	10^{+3}
MEG	mega	10^{+6}
G	giga	10^{+9}
T	tera	10^{+12}

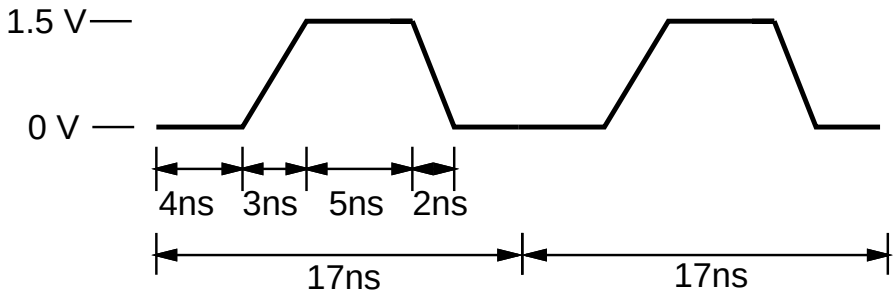
As SPICE does not differentiate between upper and lower case, '`MEG`' (or '`meg`') is used for 'mega' instead of the standard metric upper case 'M'.

The value of an element is specified in terms of the conventionally accepted units, e.g. resistance in Ohms, capacitance in Farads, and inductance in Henries. If you wish you can spell it out more fully, e.g.

```
C1 0 2 5f           or
C1 0 2 5fF          or
C1 0 2 5fFarad      or even
C1 0 2 5fthingies
```

The last alternative is allowed as SPICE actually ignores whatever follows the '`f`' and assumes Farads.

The element named '`Vin`' is an example of an independent voltage source. In this case the voltage source produces a repeating pulse, as shown below:




```
vin 1 0 pulse (0 1 5 4ns 3ns 2ns 5ns 17ns)
```

The general format for a pulse independent source is:

`V`*name Node1 Node2 pulse (Initial-Value Pulsed-Value Delay-Time Rise-Time Fall-Time Pulse-Width Period)*

The first control statement, `.tran 200ps 34ns`, specifies that a transient response simulation is to be run, i.e. we wish to know how the circuit behaves as a function of time. The total length of the simulation is to be 34 ns and the outputs are to be obtained every 500ps. The second control statement, `.print tran v(1) v(2) i(vin)`, specifies that the output is to be in the format of a printed table (`.print`); that transient waveforms (`tran`) are to be tabulated as a function of time; and that we wish to know the values for the voltage at the input (node 1 `v(1)`, the output (node 2 `v(2)`) and the current through the voltage source `Vin`

at the time steps specified in the `.tran` statement. Note that all control statements start with a period '.'. We are now ready to run the simulator.

2.2 Running Spice and Viewing the Output

The details of how to run SPICE vary from system to system. On a computer running the Unix operating system, SPICE can be run with a command line like the following:

```
spice test.cir test.out
```

This will cause SPICE to read in the file and produce the output listing file 'test.out'. Parts of the test.out are shown in Figure 2.2. The first part of the file is a header. Then the input file is listed. Just like a human circuit analyzer, SPICE has to first calculate the initial DC conditions before running the transient analysis. The results of this calculation are shown next. Finally, the transient response is tabulated.

In this output, '`D`' and '`E`' both indicate scientific notation, e.g. `4.500D-09` means 4.5×10^{-9}.

There are other ways to view the output. For example if a `.plot` control statement is used instead of a `.print` statement, the voltages and currents are plotted using character 'graphics', an example of which is given in Figure 2.3.

Using appropriate graphical packages, the output can also be plotted as smooth curves. For example, gnuplot was used to create the graph shown in Figure 2.4.

2.3 Error Messages

There are many ways to produce errors in SPICE. The most common error produced by novice users is to 'connect the circuit' up incorrectly. For example, in the SPICE input file discussed above, if we connect one of the nodes of the capacitor to node 1 instead of node 2, viz.

```
r1 1 2 1k
c1 0 1 5f
vin 1 0 pulse (0 1.5 4ns 3ns 5ns 2ns 17ns)
```

then we have not specified our circuit as drawn. In this case, we also leave one terminal of the resistor unconnected to anything else and SPICE detects the error and reports it in the output file:

```
0*ERROR*:  LESS THAN 2 CONNECTIONS AT NODE      2
```

However, life is rarely so simple. In a complex circuit it is always easy to get one node number wrong on one element but leave all of the nodes connected to two or more elements. In this case SPICE might detect no errors. If the output looks 'wrong' for any reason, the first thing to do is to draw your circuit by looking at the SPICE file as written and check that against your intended circuit.

Another important thing to remember about error messages is that SPICE is not very good at drawing attention to them. SPICE output files tend to be long and are cryptic looking. Error and Warning messages can be found almost anywhere within them. Read the entire file. Errors are discussed further in Chapter 3 and Appendix A.

```
1*******// ********  SPICE 2G.6    3/15/83 ********20:02:14******
0 SIMPLE RC NETWORK
0****     INPUT LISTING              TEMPERATURE =   27.000 DEG C
0******************************************************************
```

Header, including title from circuit file

Contents of circuit file (the "input") listed here

```
0****           INITIAL TRNSIENT SOUTION
0*************************************************************
  NODE    VOLTAGE    NODE    VOLTAGE
 (  1)    -0.0000   (  s)    0.0000

    VOLTAGE SOURCE CURRENTS
    NAME    CURRENT
    VIN     0.00D+00

0****           TRANSIENT ANALYSIS
0*************************************************************
    TIME      V(1)         V(2)         I(VIN)
X
 0.000E+00    0.000E+00    0.000E+00    0.000E+00
 5.000E-10    0.000E+00    0.000E+00    0.000E+00

 .......

 4.500E-09    2.500E+01    2.476E-01   -2.375E-06

 .......

 3.400E-08    0.000E+00    7.545E-05    7.545E-08
Y
0
    JOB CONCLUDED
```

Results of Initial Solution (DC Analysis) listed here

Output transient response tabulated as specified.

Figure 2.2: Portions of the SPICE output file.

```
0LEGEND:

 *: V(1)
 +: V(2)
 =: I(VIN)
X
     TIME      V(1)

 ...

(=)-----------------  -4.000D-06               -2.000D-06
                          - - - - - - - - - - - - - - - - - - - - - - -
 0.000D+00    0.000D+00   *                       +
 5.000D-10    0.000D+00   *                       +
 1.000D-09    0.000D+00   *                       +
 1.500D-09    0.000D+00   *                       +

 ....

  4.500D-09    2.500D-01   .            *      =    .     +
```

(Only part of the file is shown here: the legend, the 'y-axis' scale for `I(VIN)` (the 'y-axis' is left to right across the page) and part of the plot with the 'time' axis going down the page.)

Figure 2.3: Example of output produced by the `.plot` control statement.

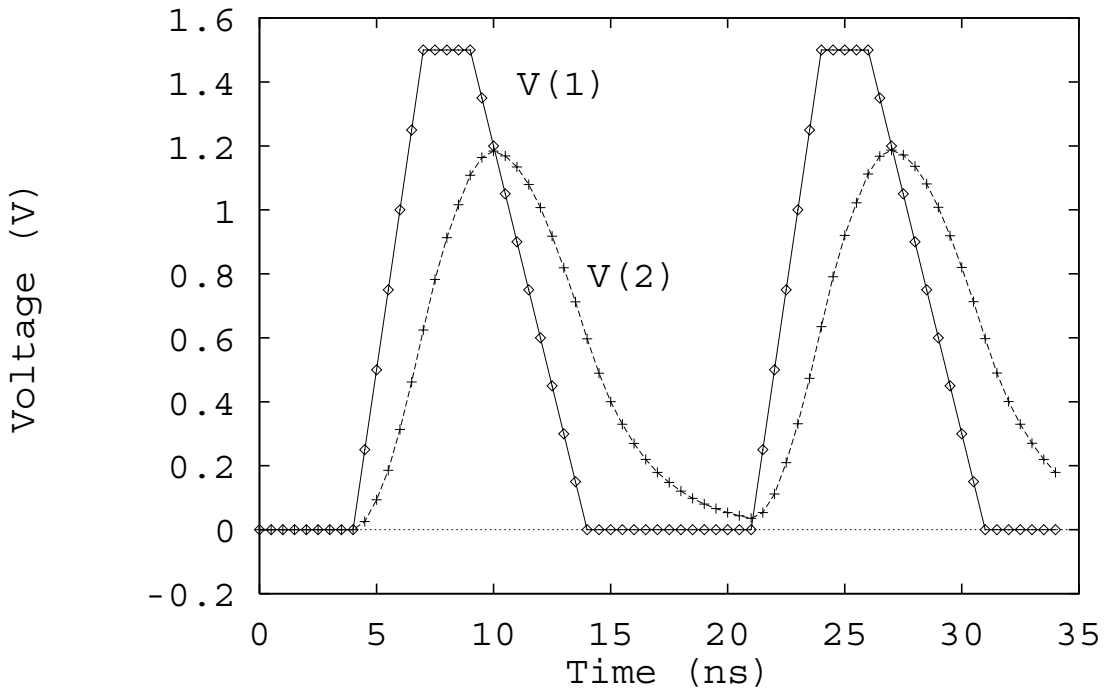


Figure 2.4: Results plotted graphically.

Chapter 3

Carrying On

In this chapter, we introduce the different element and control statement lines that SPICE allows you to use, starting with the different types of circuit elements. In particular we discuss inductors, active elements (diodes and transistors), and transmission lines. We then cover the different types of analyses you can do with SPICE, starting with the most common, the transient analysis, which is introduced in Chapter 2. The main role of the control statements is to specify these analyses.

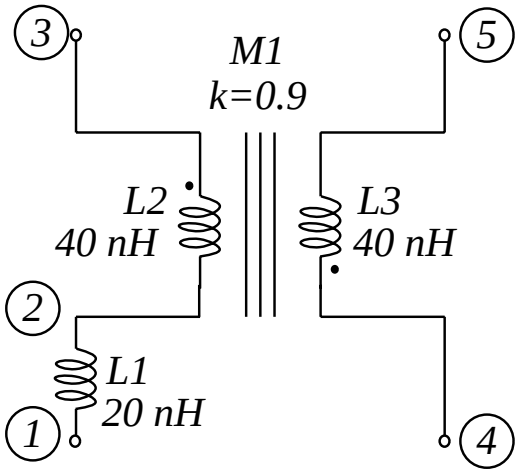
Each implementation of SPICE differs in the range of elements and control statements used. In this chapter, we describe those elements and control statements found in all versions of SPICE, i.e. those found in SPICE2G6.

3.1 Elements

Resistors and capacitors are described briefly in Chapter 2. We now look at the other elements, both passive and active. The most common form of the element is described only. For example, we do not describe how a temperature dependency could be specified. That sort of detail is found in the reference section (chapters 6 and 7).

3.1.1 Inductors and Mutual Inductors

An example showing an inductor and a mutual inductor is shown below:




```
L1 1 2 20nH
```

```
L2 3 2 40nH
L3 4 5 40nH
K  L1 L2 0.90
```

Note how the 'dot' is placed on the first node of each inductor.

3.1.2 Active Devices

Unlike passive devices, such as resistors, active, or semiconductor, devices can not be specified by one or a few parameter values. To save typing effort a separate `.model` line is created for every semiconductor device type that might appear in the circuit. In this part of the tutorial, we do not describe the meaning of the parameters in the model line. That can be found in the Reference section (chapters 6 and 7). Instead, we give examples showing how semiconductor devices are inserted into a SPICE circuit. We only describe the bipolar junction diode, bipolar junction transistor, and the MOS field effect transistor. Spice can also be used to describe junction field effect transistors and Gallium Arsenide but the process is the same.

Diodes and Bipolar Transistors

The schematic and partial SPICE net-list for a simple TTL inverter is shown in Figure 3.1. Note the following features in this circuit description:

- The format for bipolar junction transistors is:
  `Q`*name NCollector NBase NEmitter* [*NSubstrate*] *ModelName* [*Area*] [`OFF`]
  `+` [`IC=`*Vbe,Vce*]
  specifying a three terminal device.

- The format for diodes is: `D`*name* n_1 n_2 *ModelName* [*Area*] [`OFF`] [`IC=`V_D]
  specifying a two terminal device.

- The use of '`+`'s in the model lines to 'join' different lines in the file into one line for SPICE.

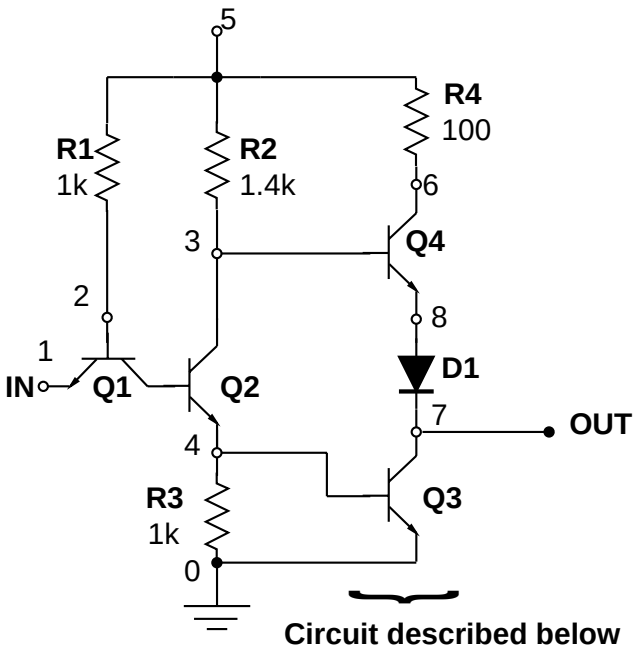
It is also possible to have an 'area' parameter in the diode and bipolar transistor element line. This area parameter is a scale factor, not an absolute measure. An area of '1.0' (which is the default) specifies that the model parameters are used unchanged. Specifying another area factor causes SPICE to change some of the model parameters to reflect a larger or smaller transistor. An area of '2.0' specifies a situation equivalent to two transistors operating in parallel.

MOSFETs

The MOSFET element line looks quite different than the element line for a bipolar transistor. There are two major differences. First, the MOSFET is a four terminal device. Three of the terminals (source, drain, gate) have analogous functions to the three terminals in a bipolar transistor. The current passes between the drain to the source (analogous to the emitter and collector) and is controlled by the voltage on the gate (analogous to the base). The fourth terminal is the 'substrate' referring to the bulk silicon in which the transistor sits. For correct functioning, the substrate must be connected to the ground or Vcc node, for n-channel and p-channel transistors respectively.

The second major difference is that the physical dimensions of the transistor are specified in the model line. Specifically, the channel length and width, and source and drain perimeters and areas are specified. These are the actual dimensions, as they appear on the chip. An example is shown in Figure 3.2. This example matches the usual general model format: `M`*name NDrain NGate NSsource NBulk ModelName* [`L=`*Length*] [`W=`*Width*]

`+` [`AD=`*DrainDiffusionArea*] [`AS=`*SourceDiffusionArea*]
`+` [`PD=`*DrainPerimeter*] [`PS=`*SourcePerimeter*]
`+` [`OFF`] [`IC=`V_{DS}, V_{GS}, V_{BS}]

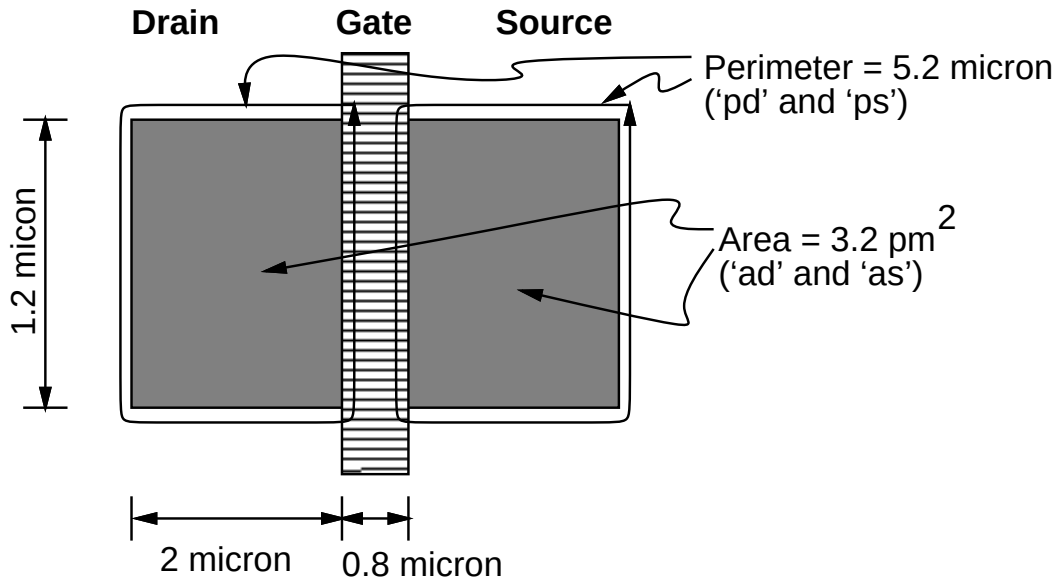



```
Q3 7 4 0 NQ1A
D1 8 7 DPN
Q4 6 3 8 NQ1A
R4 5 6 100
*
* Model element for NPN Transistor type NQ1A (courtesy Signetics)
.MODEL  NQ1A  NPN  IS= 1.95E-17  BF= 7.03E+01  VAF= 1.80E+01  IKF= 1.80E-02
```

(Remainder of model deleted)

```
* Model for diode type DPN
.MODEL DPN D(IS= 8.17E-17  RS= 2.85E+01  N= 9.99E-01  CJO= 1.65E-13
+      VJ= 8.01E-01 M= 4.61E-01  EG= 7.99E-01  XTI= 4.00E+00)
```

Figure 3.1: TTL Circuit Description.




```
M1 0 1 2 0 nenh l=0.8u w=1.6u ad=3.2p as=3.2p pd=5.2u ad=5.2u
```

Figure 3.2: Example of a MOSFET element specification.

where `[]` indicates optional parameters. PSpice supports additional element parameters. (For the full general format please see the reference catalog.)

Note that though the *drain* and *source* have different physical meanings (the *source* is the source of the majority carrier – electrons for an n-channel [nmos] device and holes for a p-channel [pmos] device), no error is produced if they are interchanged in the Spice circuit description. For example, in figure 3.2, using `M1 2 1 0 0` , produces the same simulation results as using `M1 0 1 2 0`.

An example of a CMOS digital inverter circuit, together with its Spice model is given in Figure 3.3. Note the use of the `.option` line in this example to fix circuit-wide default values for `L`, `W`, `AD`, and `AS`.

3.1.3 Transmission Lines

Though a transmission line is a four terminal device, two of the terminals are normally set to a common reference node, an example of which is shown in Figure 3.4. This lossless transmission line model supports only a single mode of propagation. If the two 'reference' terminals (nodes `0` in this example) correspond to two electrically different nodes in the physical circuit then two modes are excited and two transmission lines are required in the corresponding Spice description.

If quick simulation times are important then it is necessary to limit the use of small transmission lines. In a transient simulation the minimum time step does not exceed half the propagation delay of the line. Smaller time steps result in longer simulation times. If this is a problem, remember that a transmission line can be safely replaced by the equivalent lumped inductor and capacitor if the length of the line is smaller than 1/10th of the shortest signal wavelength of interest.

3.1.4 Voltage and Current Sources

Independent Sources

Spice supplies a number of independent voltage and current source types. As many of the source's features only make sense in the context of the analysis to be used, only some of the source's features are discussed

here. In particular, we present those features that might be used in a transient analysis (see Chapter 2 and Section 3.2.1).

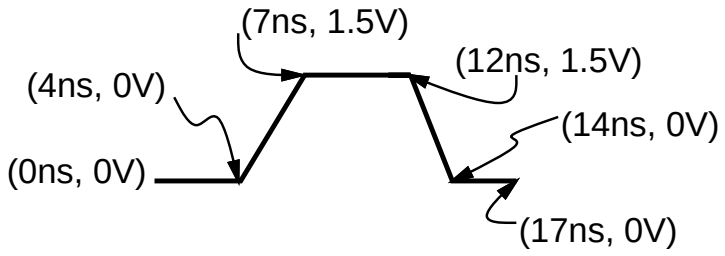
Voltage supplies are specified using DC independent sources, for example:

```
VCC 5 0 DC 5
```

for a 5 V DC power supply between nodes 5 and 0.

Any repeating non-sinusoidal waveform can be specified using the `pulse` waveform specification, an example of which was given in Chapter 2. `pulse` is often used to describe digital clocks, for example.

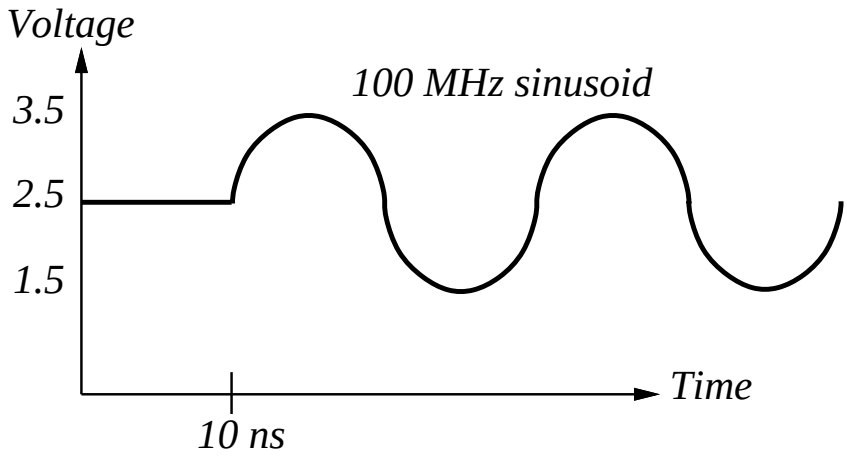
Non-repeating non-sinusoidal waveforms are specified using the piece-wise linear (`pwl`) waveform function. One period of the `pulse` example presented in Chapter 2 is shown below in the piece-wise linear format:




```
vin 1 0 pwl (0ns 0V 4ns 0V 7ns 1.5V 12ns 1.5V 14ns 0V 17ns 0V)
```

Sinusoidal and decaying sinusoidal waveforms are specified using the `SIN` function, for example:

```
Vin  4 0 sin(2.5 1 100meg 10ns)
```



Spice also allows you to specify exponential and single-frequency FM signals. Please see the reference catalog for details.

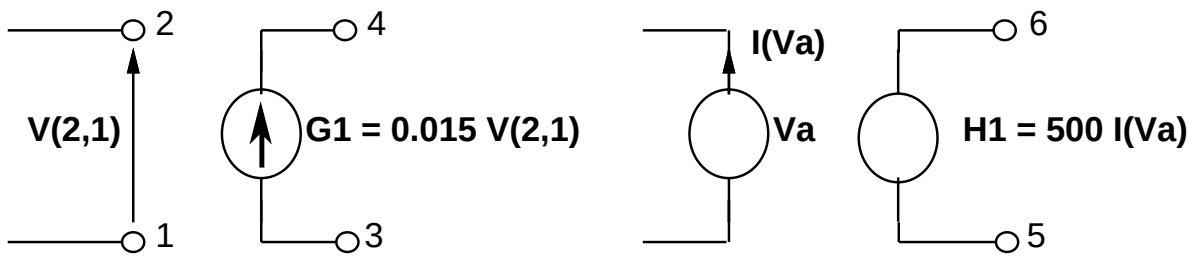
Dependent Sources

These are the most overlooked elements SPICE provides. Four different types of linear dependent sources can be specified in SPICE:

- Voltage-controlled voltage source and current-controlled current source:

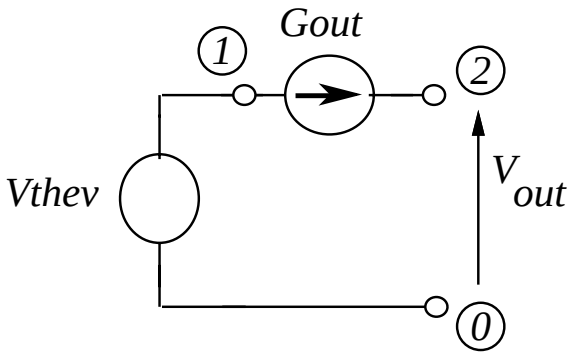

```
E1 4 3 2 1 3.3    A voltage gain of 3.3.
F1 6 5 Va 1.7     A current gain of 1.7
```

- Voltage-controlled current source and current-controlled voltage source:



`G1 4 3 2 1 15mmho` *A transconductance of* 15×10^{-3}*mho* (Ω^{-1}).
`H1 6 5 Va 0.5k` *A transresistance of 500 Ohms*

The above are linear sources. Non-linear sources can also be specified. For example, the following voltage-controlled current source actually specifies a non-linear resistance that could be used as part of a non-linear Thevenin equivalent circuit:




```
Gout 2 1 2 0 0 1m -0.6m
```

The format used in this example is:

G*xxx node1 node2 ref-node1 ref-node2 C0 C1 C2*

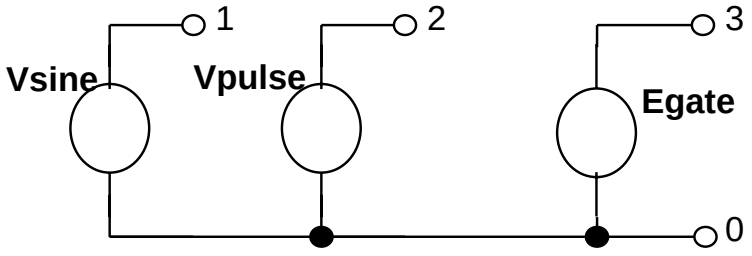
to produce a dependent source that obeys the equation:

$$I = C0 + C1(V(ref - node2) - V(ref - node1)) + C2(V(ref - node2) - V(ref - node1))^2.$$

In this case, the equation specifying the current is:

$$I = 0 + 1 \times 10^{-3} V_{out} - 0.6 \times 10^{-3} V_{out}^2$$

The above non-linear source is quadratic and dependent on only one other variable. The same format can be used to specify higher order polynomials. A source dependent on the voltages/currents on/in `ND` other nodes/branches can be specified by including a `poly(nd)` statement in the element line. For example, the following linear voltage-controlled voltage source specifies a gated sinusoidal source:

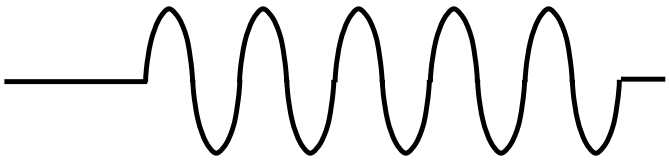



```
Vsine 1 0 sin (0 0.5 100k 5us)
Vpulse 2 0 pwl (0ns 0V 14us 0V 15us 1V 65us 1V 66us 0V)
Egate 3 0 poly(2)  1 0   2 0  0 0 0 1
```

i.e. This source specifies a voltage,

$$\texttt{Vgate} = 0 + 0 \times \texttt{Vsine} + 0 \times \texttt{Vpulse} + 1 \times \texttt{Vpulse} \times \texttt{Vpulse}$$

which has the following waveform:



In its general form, a polynomial of any complexity can be specified. e.g. The generalized voltage controlled voltage source,

```
EX <node> <node> poly(2) V1 V2  k0 k1 k2 k3 k4 k5 k6 k7
```

specifies a controlled voltage of the form

$$EX = k1 + k2 \times V1 + k3 \times V2 + k4 \times V1 \times V2 + k5V1^2 + k6V2^2 + k7V1^2V2^2$$

This could be extended to create polynomials as a function of 3, 4, etc. variables. However, as a practical matter, it is very difficult to read and understand non-linear polynomials with more than two inputs. It is easier to create two-input polynomials separately and combine them with another polynomial.

3.2 Analyses

3.2.1 Transient Analysis

In the transient analysis response is observed with one or more time-varying inputs. A simple example is given in Chapter 2.

The first step performed by SPICE in a transient analysis is to compute the initial DC or bias point condition. During this computation it is assumed that the voltage across capacitors is zero, the current through inductors is zero, and the value for dependent sources is zero. SPICE then conducts the transient simulation by calculating all of the voltages and currents at a set of points in time. In the rest of this section, we discuss a number of issues related to transient analyses, starting with a treatment of convergence.

DC Convergence

During both the DC analysis and the following transient analysis iterative numerical techniques are used to obtain a solution. The objective of these techniques is to iterate on the value of the node voltages and branch currents until successive iterations only bring very small changes in their values, i.e. SPICE *converges* on a solution in the DC analysis and at every time step.

Sometimes SPICE can not converge on a solution. If this occurs during the DC analysis it will report this problem in the output file with a '`convergence problem`' message like. Failure to converge in the DC analysis is usually due to an error in specifying node numbers, circuit values or model parameter values. These should be checked carefully before proceeding further. However, sometimes SPICE is having a genuine problem in converging and you might have to help it find a solution.

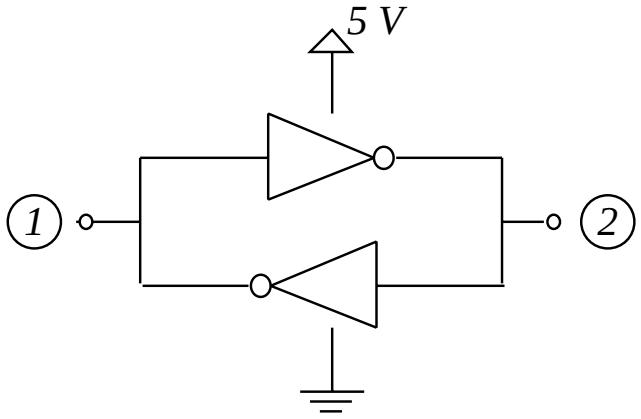
In many bistable circuits (e.g. flip-flops) and positive feedback circuits, SPICE will not converge in the DC analysis or will converge to an undesirable value (e.g. midway between logic-0 and logic-1 in a latch). One way to help SPICE converge to the correct value is to use the `off` option to turn off devices in the feedback path, e.g.,

```
M0 0 1 2 0 pd=5.2u ad=5.2u off
```

allowing SPICE to find a DC solution. Spice turns the devices back on during the transient analysis. Another approach is to use `nodeset` to provide 'hints' to SPICE or to specify initial conditions that force a solution.

Nodeset and Initial Conditions

The basic difference between using nodeset and specifying initial conditions is that the latter forces nodes to the specified voltage while nodeset only provides hints. The values specified by the nodeset line are only used during the first part of the DC solution procedure and then ignored in the later parts. Thus if they are incorrect, or inconsistent, convergence is not prevented. As an example of nodeset, ts use as follows in the the simple latch, will result in the output (node `2`) converging to 5 V (assuming a CMOS latch):




```
.nodeset V(1)=0V
```

When initial conditions are set, they are used through the entire DC solution right to the start of the transient analysis. For example in the circuit above, the use of `.IC V(1)=1V` will result in `V(1)` starting at 1 V in the transient analysis while the use of `.nodeset V(1)=1V` would result in `V(1)` starting at 0 V. An error or inconsistency in specifying initial conditions with `.IC` might prevent SPICE from converging.

A second way to specify initial conditions is to specify them in the element lines. For example, the statement,

```
C1 6 0 IC=3.1
```

initializes the voltage across capacitor `C1` to 3.1 V. For an inductor, the following statement will set the initial current flowing through it to 4.3 mA:

```
L3 4 5 IC=4.3m
```

If `IC=` statements are used then it is necessary to include a "Use IC=" (`UIC`) statement in the `.tran` statement, e.g.

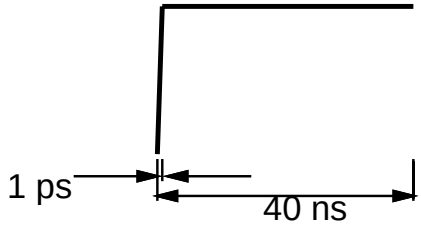
```
.tran 200ps 34ns UIC
```

Specifying `UIC` commands SPICE to skip the DC bias calculation, making it is necessary for the initial conditions to be completely specified through a combination of `IC=` and `.IC` statements. Be careful.

Simulation Time Step Size

Using a smaller time step increases both the results accuracy and computer run-time of the simulation. One thing to be very aware of is if short transmission lines or very fast edges are specified then the simulation time step will be very short. For example, trying to obtain a 'step response' with a waveform/statement

such as the following will greatly increase rise time (and also quite likely lead to convergence problems).




```
.Vin 4 0 PWL 0ns 0V 1ps 5V 40ns 5V
```

It is also possible to change the time step, and other step-related parameters, in the `.options` statement. Please see the reference catalog for details.

Transient Analysis Convergence Problems

SPICE might report a transient analysis convergence problem with a message like the following:

```
*ERROR*: Convergence problem in Transient Analysis at

         TIME =
```

etc.

Sometimes SPICE is not so hopeful and just 'dumps' you part way through the analysis, e.g. part way into a 40 ns analysis SPICE might suddenly stop the analysis at 34 ns and end with:

```
  3.400E-09     5.452E+00   6.602E+00   6.892E+00
Y
0          ***** JOB ABORTED
```

In this case, the problem was a too-short implicit time step caused by a very short (62.3 ps delay) transmission line:

```
Tline3 10 0 11 0 z0=60 td=62.3ps
```

Replacing the line with its equivalent lumped circuit,

```
Lline3 10 11 7.48nH
Cline3a 10 0 1.03pF
Cline3b 11 0 1.03pF
```

solved the problem.

If your transient analysis convergence problem is not being caused by a too short a time step, then it is most likely caused by an error in specifying a circuit node number or parameter value. Your circuit and `.model` lines should be checked carefully. Often looking at the circuit description as specified in the output listing is more useful than looking at the file you typed in, as the output listing is describing what SPICE 'sees'.

However, SPICE is a numerical program and can be quirky. For example, one simulation driven by the pulse

```
V2 4 0 Pulse(0V 5V 0n 1.2n 1.2n 20n 40n)
```

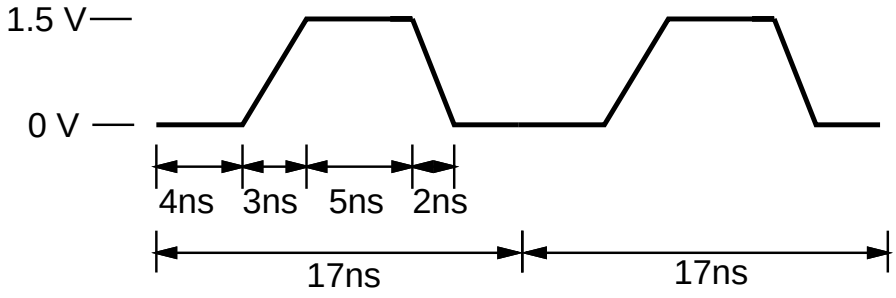
would abort about half way through the simulation. However, turning the pulse 'up-side-down' (interchanging `0V` and `5V`),

```
V2 4 0 Pulse(5V 0V 0n 1.2n 1.2n 20n 40n)
```

allowed the simulation to complete.

Spectral Analysis – Fourier Transform

The `.Four` control statement can be used to find the spectrum of any time-varying signal in a transient analysis. For example, in Chapter 2, we used the following time-domain signal as the input to an RC circuit:




```
vin 1 0 pulse (0 1.5 4ns 3ns 5ns 2ns 17ns)
```

The addition of the control statement,

```
.Four 58.82MegHz V(1) V(2)
```

to this file, specifies that the spectrum of the input (`V(1)`) and output (`V(2)`) voltage waveforms are also to be obtained. The frequency specified in this statement is the fundamental frequency of the waveform (1/17 ns = 58.82 MHz). As a result of this statement, the output file reports the magnitude and phase of the first nine harmonics for each signal. In this case, the output for V(1) is:

```
0****     FOURIER ANALYSIS                 TEMPERATURE =   27.000 DEG C

0******************************************************************************

 FOURIER COMPONENTS OF TRANSIENT RESPONSE V(1)
0DC COMPONENT =   5.293D-01
0HARMONIC   FREQUENCY    FOURIER    NORMALIZED    PHASE     NORMALIZED
    NO         (HZ)     COMPONENT    COMPONENT    (DEG)    PHASE (DEG)

     1      5.882D+07   7.750D-01     1.000000   -91.352       0.000

     2      1.176D+08   2.561D-01     0.330483    99.249     190.601

     3      1.765D+08   7.623D-02     0.098364    16.321     107.673

     4      2.353D+08   2.980D-02     0.038451  -123.300     -31.947

     5      2.941D+08   2.502D-02     0.032282  -167.139     -75.786

     6      3.529D+08   6.971D-03     0.008994   -46.439      44.913

     7      4.118D+08   9.869D-03     0.012734    81.719     173.071

     8      4.706D+08   1.722D-02     0.022225    -9.729      81.623

     9      5.294D+08   1.346D-02     0.017363  -169.978     -78.625
```

A similar table is obtained for V(2).

3.2.2 DC Analyses

Spice enables you to conduct the following DC analyses:

- DC solution for a particular input voltage/current condition (`.OP`).
- DC solutions over a range of input conditions (`.DC`).
- Small signal DC transfer functions, including gain, input and output resistance (`.TF`).
- Sensitivity of the DC value of an output to some set of parametric variations (`.SENS`).

These are discussed in turn.

The insertion of a line with just

```
.OP
```

on it asks SPICE to determine the DC bias point of the circuit with inductors shorted and capacitors opened, just the same as the DC analysis conducted before a transient analysis. It might be used in situations where you wish to know the DC bias point but the analysis you are doing does not request it (e.g. such as when determining a frequency response).

A command line beginning with `.DC` instructs SPICE to sweep the specified voltage source over the specified range, reporting the DC bias point for each combination of input conditions. If more than one source is specified in the `.DC` statement, then the first source will be swept over its entire range for every value of the second source. An example is given in Figure 3.5 in which two analyses alternatives are presented at the bottom. The left hand alternative instructs SPICE to plot the transfer characteristics of the CMOS inverter, the right hand example instructs SPICE to plot the output V-I characteristics for when Vin is 5 V. Both examples specify that the voltage sweep is to be from 0 to 5 V in 0.1 V increments. The resulting output V-I characteristic (obtained using the statements on the right hand side of the Figure 3.5) is plotted in Figure 3.6.

Now, if you wish to find the small signal output resistance at say `Vout` $= X$ V,

$$r_{out} = \left.\frac{\partial v}{\partial i}\right|_{V_{out}=X\ V}$$

then one way to obtain this would be to measure the slope of the plot shown in Figure 3.6 at `Vout` = 0.5 V. However, SPICE provides an easier way to get this result as a transfer function `.TF`:

```
.TF I(vout) Vout
```

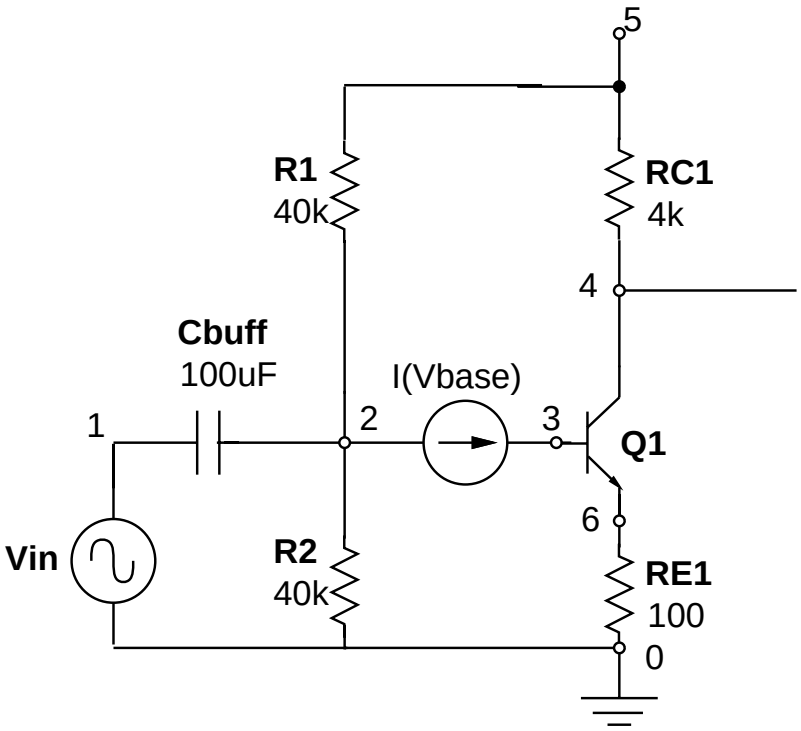
There is no need for a `.print` statement with `.TF`. Running this produces the output:

```
OUTPUT RESISTANCE AT I(VOUT)              =  2.247D+03
```

A DC sweep can also be done by specifying a slow moving input and a conducting a transient analysis. Sometimes this is necessary, for example in circuits with hysteresis, such as a Schmitt Trigger.

Sometimes we wish to know the sensitivities of various output parameters with respect to variations in circuit parameters. For example, we might wish to know whether to specify resistors to +/-10% or +/-1% in order to guarantee a certain bias point in a transistor amplifier. This is done with the `.sens` statement. The following example determines the sensitivity of the bias point of an amplifier to variations in resistance

values:



Extract from input file:

```
Common-Emitter Amplifier

R1 2 5 40k
R2 2 0 40k
* Measure the Base current
Vbase 3 2
Q1 4 3 6 NQ1A
RC1 5 4 4k
RE1 6 0 100

* AC source with unity magnitude and AC buffered
Vin 1 0 AC
Cbuff 1 2 100u

Vcc 5 0 5V

* Find the bias point
.OP

*Find the sensitivity of the bias voltage at the collector
.sens V(4)
```

Extracts from output file reporting DC bias point and the sensitivity analysis:

```
  NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE     NODE   VOLTAGE
NODE   VOLTAGE     NODE   VOLTAGE
```

```
 (  1)    0.0000    (  2)    1.1432    (  3)    1.1432    (  4)    0.3220    (
  5)    5.0000    (  6)    0.1237

0DC SENSITIVITIES OF OUTPUT V(4)
0        ELEMENT         ELEMENT       ELEMENT       NORMALIZED
          NAME            VALUE      SENSITIVITY    SENSITIVITY
                                    (VOLTS/UNIT) (VOLTS/PERCENT)

          R1           4.000D+04      1.114D-06      4.457D-04
          R2           4.000D+04     -3.303D-07     -1.321D-04
          RC1          4.000D+03     -7.010D-05     -2.804D-03
          RE1          1.000D+02      1.192D-03      1.192D-03
```

In this case, if the value for `RE1` changed by 100% the collector voltage would only change by only 119 mV.

3.2.3 Small Signal AC Analysis

Analog circuits are often analyzed in terms of their frequency response to steady-state, sinusoidal, small-voltage, input signals. With small voltage swing signals, all of the circuit elements can be treated as being linear around some bias point. Three types of AC analysis can be done:

1. Obtain circuit response(s) as a function of frequency using the `.AC` analysis.

2. Conduct a noise analysis as a function of frequency using a `.NOISE` element together with a `.AC` element.

3. Analyze the circuit for harmonic distortion using the `.DISTO` element together with the `.AC` element.

In this section, we discuss the first two types of analysis only. The distortion analysis capability provided in SPICE2G6is somewhat limited and so is not presented.

Consider the frequency response of the LC filter described, with its SPICE file, in Figure 3.7. There are several features in this file that differentiate it from a file specifying a transient analysis. First the signal source `Vin` is specified as an `AC` source, not a source in the time domain. Here it specifies a sinusoid with a magnitude of 1 Volt. The `.AC` control statement specifies that we wish the frequency range to be swept over a frequency range of 100 Hz to 10 kHz in decade (`dec`) increments with `20` points per decade. i.e. The output contains a total of 40 frequency points, 20 between 100 Hz and 1 kHz and 20 between 1 KHz and 10 Hz. The `.print` statement specifies that this is an AC analysis and specifies that the magnitude of the voltage (`VM`) of node 3 with respect to node 0 be printed at each frequency point. Examples of other results that can also be obtained include:

Control statement Example	**Meaning**
`.print AC VR V(2,3)`	*Real part of the voltage across the inductor*
`.print AC VI V(2,3)`	*Imaginary part of the voltage across the inductor*
`.print AC VP I(Vin)`	*Phase of current through the voltage source*
`.print AC VDB(3)`	*Voltage in dB,* $10 \times log_{10}(magnitude)$

The results obtained by running the SPICE file specified in Figure 3.7 are shown in Figure 3.8. Note again that a DC analysis is carried out before the AC analysis so as to obtain the bias point (this is not shown).

In Section 3.2.2, we show how to obtain the (non-linear) output impedance as a function of the output voltage. For small voltage swing signals, all impedances are linear, so we are interested in input and output impedance as a function of frequency. For example, we could plot the output impedance of the LCR circuit above using the following SPICE file:

```
RLC filter
*
* 'Short' input so that it does not form
* part of the output impedance
Vin 1 0 AC 0V
*
R1 1 2 15Ohm
L1 2 3 50mH
C1 3 0 1.5uF
*
.AC dec 20 100Hz 10kHz
*
* Measure output impedance with a current source
Iout 0 3 AC 1
*
* Measure Zout:
.print AC VM(3)
.end
```

The output impedance is plotted in Figure 3.9.

Spice is also capable of conducting a noise analysis as part of the AC analysis. This analysis is often useful as an aid to the design of analog circuits. For full details please see the `.NOISE` and `.PRINT` control statement descriptions in the reference catalog (Part III).

3.2.4 Monte Carlo Analysis

The Monte Carlo analysis is a statistical analysis of the circuit causing the circuit to be analyzed many times with a random change of model parameters (parameters in a `.MODEL` statement). It is available in the PSpice version only.

The form on the Monte Carlo analysis is

PSpice*Form*

`.MC` *NumberOfRuns AnalysisType OutputSpecification OutputFunction* [`LIST`]
`+` [`OUTPUT(` *OutputSampleType* `)`] [`RANGE(`*LowValue*`,` *HighValue*`)`]
`+` [`SEED=`*SeedValue*]

Monte Carlo analysis repeates `DC` analysis as specified by the `.DC` statement, `AC` small-signal analysis as specified by the `.AC` statement, or transient analysis as specified by the `.TRAN` statement. In the `.MC` statement the way in which the results of the multiple runs are interested is controlled by the *OutputSpecification*] and *OutputFunction* parameters.

A typical use of Monte Carlo analysis is to predict yield of a circuit by examining the effect of process variations such as length and width of transistors. As well the effect of temperature on circuit performance can be investigated.

The initial run uses the nominal parameter values given in the NETLIST. Subsequent runs statistically vary model parameters indicated as having either lot or device tolerances. These tolerances are specified in a `.MODEL` statement.

3.2.5 Transfer Function Specification

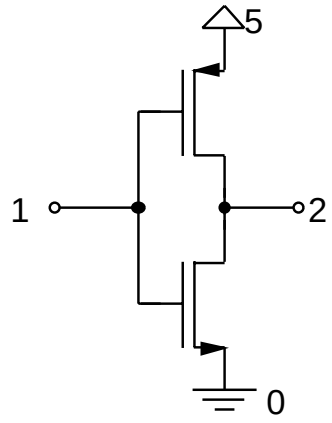
The transfer function specifies a small-signal `DC` analysis from which a small-signal transfer function and input and output resistances are computed. The transfer function computed is the ratio of the `DC` value of the output quantity to the input quantity. In the above examples the following transfer functions are

computed:

EXAMPLE	Transfer Function
.TF V(10) VINPUT	$\frac{\texttt{V(10)}}{\texttt{VINPUT}}$
.TF V(10,2) ISOURCE	$\frac{\texttt{V(10,2)}}{\texttt{ISOURCE}}$
.TF I(VLOAD) ISOURCE	$\frac{\texttt{I(VLOAD)}}{\texttt{ISOURCE}}$

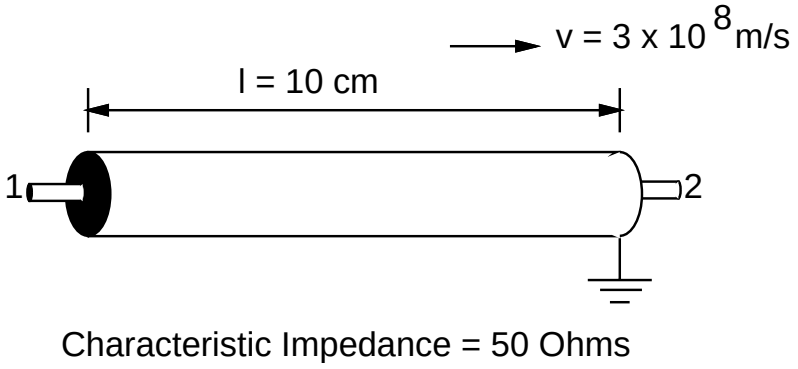
3.2.6 Parameteric Analysis

3.2.7 Sensistivity and Worst Case Analysis



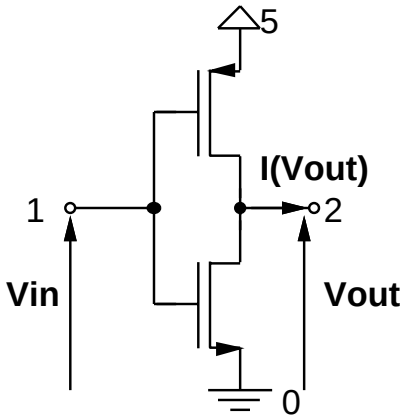

```
CMOS Inverter Example
*
M0 0 1 2 0 nenh l=0.8u w=1.6u ad=3.2p as=3.2p pd=5.2u ad=5.2u
M1 5 1 2 5 penh l=0.8u w=1.6u ad=3.2p as=3.2p pd=5.2u ad=5.2u
Vcc 5 0 DC 5V
*
* following option line fixes transistor length and width, and
* drain/source area defaults
*
.options defl=0.8u defw=1.6u defad=3.2p defas=3.2p
*
*
.model nenh nmos
+    Level=2               Ld=4.000e-8        Tox=1.750000e-08
+    Nsub=1.506725e+17     Vto=0.59073        Kp=6.124495e-05

(Remainder of model omitted.)

.model penh pmos
+    Level=2               Ld=4.000000e-08    Tox=1.750000e-08

(Remainder of model omitted.)
```

Figure 3.3: A CMOS inverter.




```
T1 1 0 2 0 Z0=50 TD=333ps
```

Figure 3.4: Example of a transmission line.




```
MOS Inverter
*
M0 0 1 2 0 nenh l=0.8u w=1.6u ad=3.2p as=3.2p pd=5.2u ad=5.2u
M1 2 1 5 5 penh l=0.8u w=1.6u ad=3.2p as=3.2p pd=5.2u ad=5.2u
Vcc 5 0 DC 5V
```

```
Vin 1 0                 Vin 1 0 5V
                        Vout 2 0
.DC Vin 0 5 0.1         .DC Vout 0 5 0.1
.print DC V(2)          .print DC I(Vout)
```

Figure 3.5: DC response example. Two alternative analyses are presented at the bottom and are described in the text.

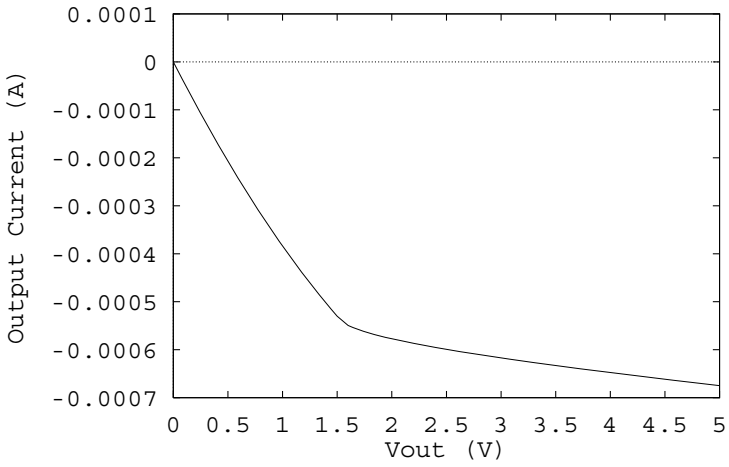
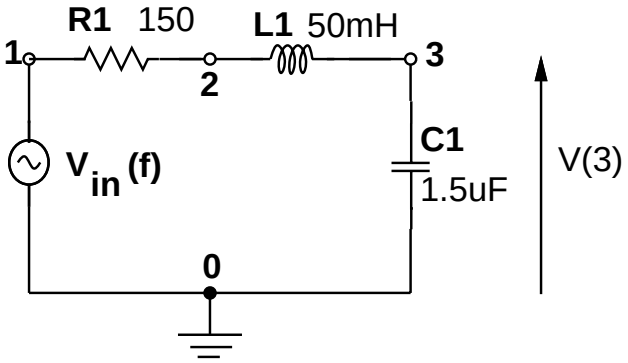


Figure 3.6: Output V-I characteristics for CMOS inverter.




```
RLC filter
*
Vin 1 0 AC 1V              AC signal source
*
R1 1 2 15Ohm
L1 2 3 50mH
C1 3 0 1.5uF
*
.AC dec 20 100Hz 10kHz     AC analysis specification
*
.print AC VM(3)            print results of AC analysis
```

Figure 3.7: LC filter and the SPICE file specifying a frequency response analysis. (Comments in *emphasis* are not part of the file.)

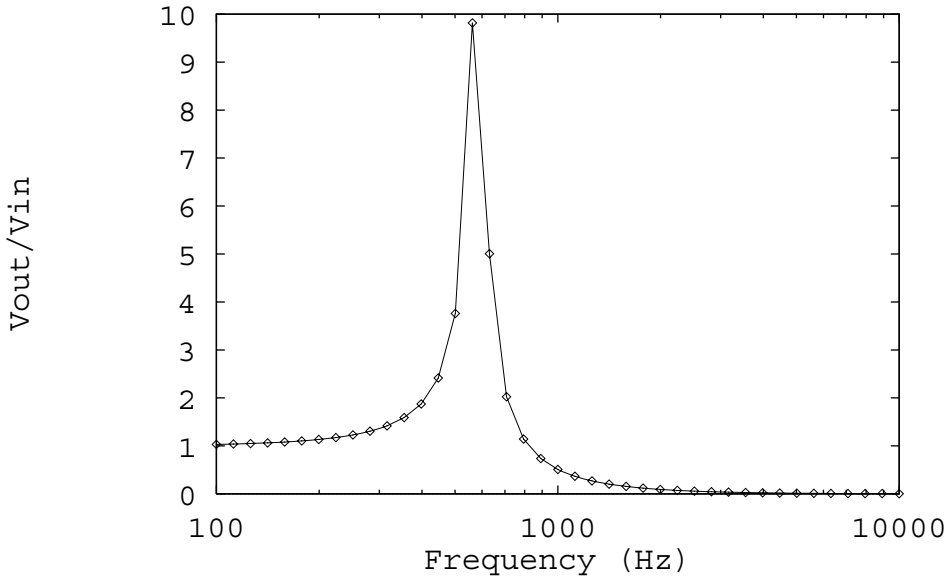


Figure 3.8: Frequency response of the circuit shown in Figure 3.7.

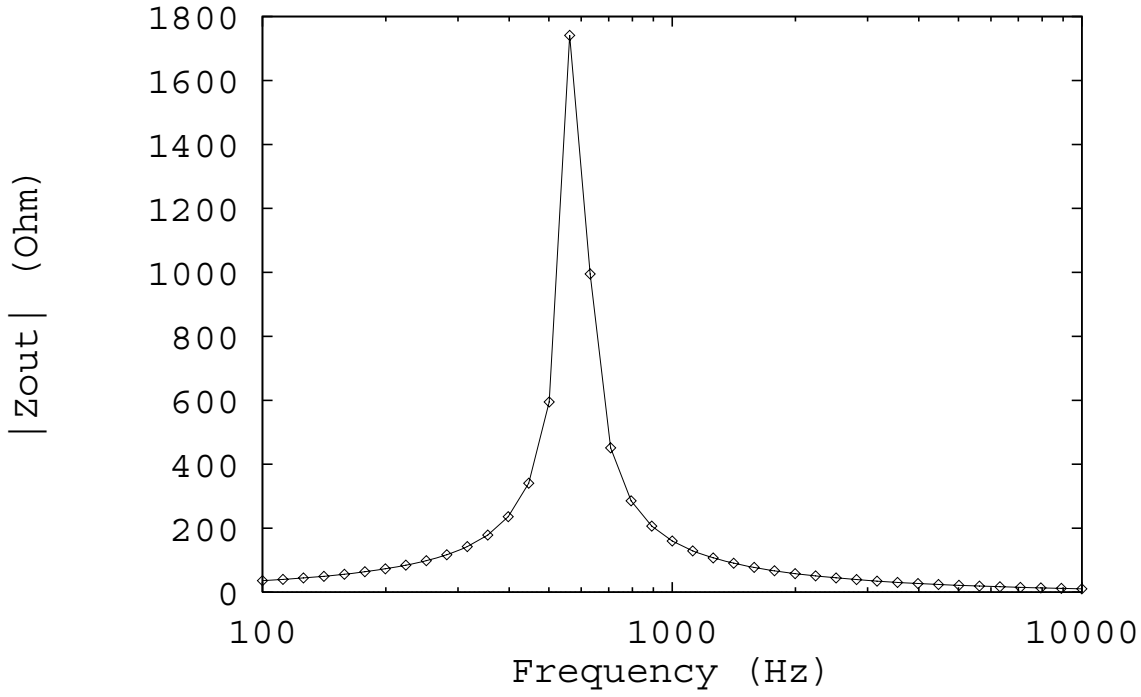


Figure 3.9: Output impedance vs. frequency for LC circuit shown in in Figure 3.7.

Chapter 4

How Spice Works

4.1 Introduction

SPICE has three basic analyses: `DC` analysis — initiated by the `.DC` statement; `AC` analysis — initiated by the `.AC` statement; and transient analysis — initiated by the `.TRAN` statement. `AC` analysis involves incorporating the relations relating terminal voltages and device currents into a matrix equation This matrix equation is called the network equation and embodies the topology of the network and the constitutive relations (that is, device equation) describing individual devices. It derives from a mathematical statement of Kirchoff's current law with device currents replaced by node voltages with the substitution achieved by using the constitutive relations of individual devices. Extensions to this basic form of the network equation (called the modal formulation of the network equation) is required for elements that can not be mathematicly modeled as a current or currents in terms of voltages. `AC` analysis is the appropriate place to begin a technical exposition of the analysis algorithms in SPICE as it illustrates the method of development of network equation is used by all other analyses.

Transient analysis requires, as well, that a time-stepping numerical integration algorithm be incorporated into the network equations. A straight forward approach to transient analysis, which was used prior to the development of the "SPICE" approach, is to derive the state equations in state space (that is in differential form using the s operator) and apply the time discretization of a numerical integration method to the complete network equation. However, in SPICE the time discretization is incorporated in the constitutive relations before the network equation is developed. The time discretized network equation must be solved iteratively and the Newton iteration procedure is applied to the device equations along with the time-discretization step. This has proved to be a particularly robust approach and is the main reason SPICE is so widely accepted.

`DC` analysis is a special case of transient analysis without the time discretization step and so is discussed after transient analysis.

4.2 AC Small Signal Analysis

`AC` small signal analysis is inititated by the `.AC` statement (see page 53). The aim in `AC` analysis is to determine the `AC` voltage at every node in the circuit which is now linear because of the small-signal approximation. First of all a matrix equation is developed that relates node voltages to external current sources. Through Thevenin's theorem external voltage sources are converted to external current sources and node voltages are easily related to voltages across elements. This formulation of the network equation is called the nodal admittance formulation. Unfortunately there are several elements, such as the current controlled voltage source element, which do not have an admittance description as required in this approach. A method of circumventing this problem is discussed in section **??** and is a direct extension of the method discussed below.

Consider the general network $\mathcal{N}$ in Figure 4.1(a) with N internal nodes in addition to the reference node.

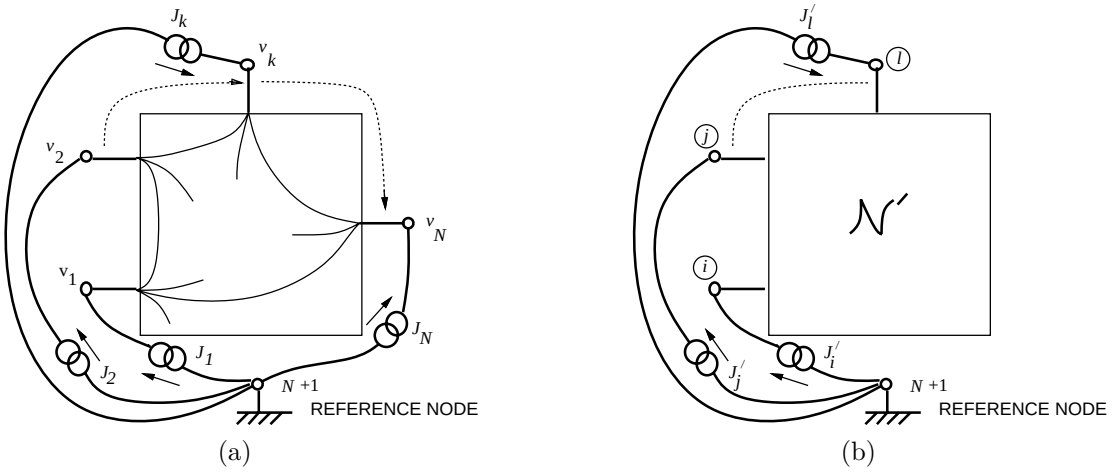


Figure 4.1: Definition of networks: (a) $\mathcal{N}$; and (b) $\mathcal{N}'$.

In general the reference node is not part of the circuit and the development of the network equations is simplified by treating all of the nodes (except the external reference node) in the same way.

All of the nodes of the network have external current sources J with J_n being the external current source between the n th node and the reference node. In practice most of the external current sources, the J's, are zero and those that are not are due to independent sources. The network equations used in SPICE relate the node voltages, the v's, and the J's. In matrix form

$$\mathbf{Y}\mathbf{v}_n = \mathbf{J}_n \tag{4.1}$$

where $\mathbf{v}_n$ is the vector of node voltages, and $\mathbf{J}_n$ is the vector of external currents. Solution of equation (**??**) enablEs the node voltages to be evaluated given the external current sources.

In SPICE the grounded reference node, node "`0`", is part of the network and so there is a linear dependence of the rows of $\mathbf{Y}$ ($|\mathbf{Y}| = 0$). Hence $\mathbf{Y}$ is called the indefinite nodal admittance matrix. One row and one column can be deleted from $\mathbf{Y}$ without the lossing any information to yield what is called the definite nodal admittance matrix. For now consider that the reference node is an arbitrary node. The correction required because the reference node is actually part of the circuit is discussed in section **??**.

4.3 DC Analysis

The analysis of nonlinear resistive circuits, or equivalently the analysis of circuits at `DC` is an important first step in `AC`and transient analysis. In both cases nonlinear resistive analysis determines the initial starting point for further analysis incorporating energy storage elements such as capacitors and inductors.

`DC` analysis is SPICE is identical to transient analysis discussed in the previous section except that the contributions of capacitors and inductors are ignored. `DC` analysis has better convergence properties ithan transient analysis since energy storage elements, and thus resonant responses, are eliminated. As well the analysis is numericly efficient since only a steady-state response is required and calculated.

4.4 Discussion

SPICE supports several analyses other than those discussed above. Essentially these are extensions of the `AC`, `DC` and transient analyses.

The transfer function analysis, initiated by the `.TF` statement, calculates a transfer function as the ratio of the `DC` value of an output quantity to the `DC` value of an input quantity over a range of values of the input quantity. A `DC` analysis at each value of the input quantity is performed.

With the `.DISTO` statement a distortion analysis is performed by dteremining the steady-state harmonic and intermodulation products for small input signal. Evaluation of the small-signal distortion is based on a third-order multi-dimensional Volterra series expansion of nonlinearities around their operating point. The method for calculating of the distortion products parallels `AC` analysis algorithm.

The sensitivity analysis calculates the `DC` small-signal sensitivities of each output quantity with respect to every circuit parameter. It is initiated by the `.SENS` statement. The transfer function computed is the sensitivity (or partial derivative) of the `DC` value of the output quantity with respect to the each and every circuit parameter.

If the `.NOISE` statement is included in the input file the noise generated by active devices and resistors is evaluated. All active devices and some passive devices have noise models consisting of uncorrelated noise current sources. These noise current sources are used as the external current sources in `AC`analysis. One noise source at a time is considered and the response at the output terminals and sources specified are calculated. The contributions from each source are added in a root-mean-squared sense as they are uncorrelated. The noise analysis utilizes the network equation formulated and solved in `AC` analysis.

A Monte Carlo analysis is performed when the `.MC` statement is specified. In the Monte Carlo analysis either a `DC`, an `AC`, or a transient analysis is performed multiple times.

An operating point analysis initiated by the `.OP` statement is just a single `DC`analysis.

The Fourier analysis performed when the `.FOUR` statement is used is not really a separate analysis at all. Really it is a way of examiing the results of a transient analysis by taking the Fourier transform of a voltage or current response.

4.5 To Explore Further

The essential aspects of SPICE are the models of devices and the algorithms for formulating and solving the network equations. The derivation of the `AC` model of devices is straightforward requiring the y parameters of the device. These are obtained from the analytic derivatives of the device equations calculated at the operating point of the circuit as determined from a `DC` analysis. For transient and `DC` analysis the associated discrete circuit model must be calculated from the device equations. The device equations of semiconductor devices are calculated from knowledge of the device physics. Derivation of the models used for semiconductor devices in SPICE2G6 and SPICE3 are described in [2] (*Semiconductor Device Modeling with SPICE* edited by P. Antognetti and G. Massobrio). Without fail the models in SPICE2G6 are available in all commercial versions of SPICE. SPICE3 and most commercial versions of SPICE provide additional or enhanced models. Derivations of more advanced models are described in [8] (*Semiconductor Device Modeling for VLSI* by K. Lee, M. Shur, T. Fjeldly and T. Ytterdal).

In SPICE the network equations are stored in sparse matrices to conserve memory usage. This also results in much faster solution of the network equations than if regular matrices were used. Greater detail than that provided in this chapter of the numerical algorithms used in SPICE can be found in [1] (*Fundamentals of Computer-Aided Circuit Simulation,"* by W. J. McCalla).

Chapter 5

Input File

5.1 Introduction

The operation of SPICE is controlled by statements which are embedded in an input file which includes as well descriptions of elements and their topology. The description of the elements and their topology is also known as a netlist. The output or results of a SPICE run are logged in an output file and in more modern versions of SPICE, in a data file for subsequent interactive plotting and analysis. With PSPICE the program for subsequent analysis is probe and with SPICE3 the comparable tool is NUTMEG.

5.2 Circuit Model

The model used by SPICE to represent circuits is as shown in Figure 5.1. SPICE supports a hierarchical description of a circuit with subcircuits. A large number of parameters are required for many elements, especially for active devices, and for these the model concept is introduced where most of the parameters of active elements can be defined separately from invocation of the element. This permits a single model description to be used by many elements. A model is specific to a particular element type but not all elements have models. Mostly models are used to set the parameters describing a semiconductor fabrication process and so are common to many elements.

When the input file is read subcircuits definitions and models are stored internally separately from the main circuit. Subcircuit calls are expanded if the subcricuits referenced are already defined and stored internally. If a referenced subcircuit is not defined then the subcircuit call is flagged as not being expanded and only when the input file has been completely read (up to the `.END` statement) is an attempt made to resolve incomplete expansions. In PSPICE library files (described in the `.LIB` statement discussion on page 68) are checked. The first time a library file is to be searched, an internal table of which subcircuits and models are available and where they can be found in the library files is constructed. Models are treated in a similar way, a model is used if it has been defined otherwise resolved references to models are expanded once the input file has been completely scanned (up to the `.END` statement). With PSPICE evaluation of expressions in subcircuits and models is only performed when on fully expanded subcircuits. Note that expressions and libraries are not supported in SPICE2G6or in SPICE3.

5.3 Input Lines

The input file of SPICE is essentially unstructured. It must begin with a `TITLE` line and should end in a `.END` statement although this is automaticlly assumed if the end of the input file is read. The string on the `TITLE` line is used as the banner in the output log file appearing at the top of each page. The `.END` statement marks the end of one circuit with the effect that several circuits can be specified in the file (at least for

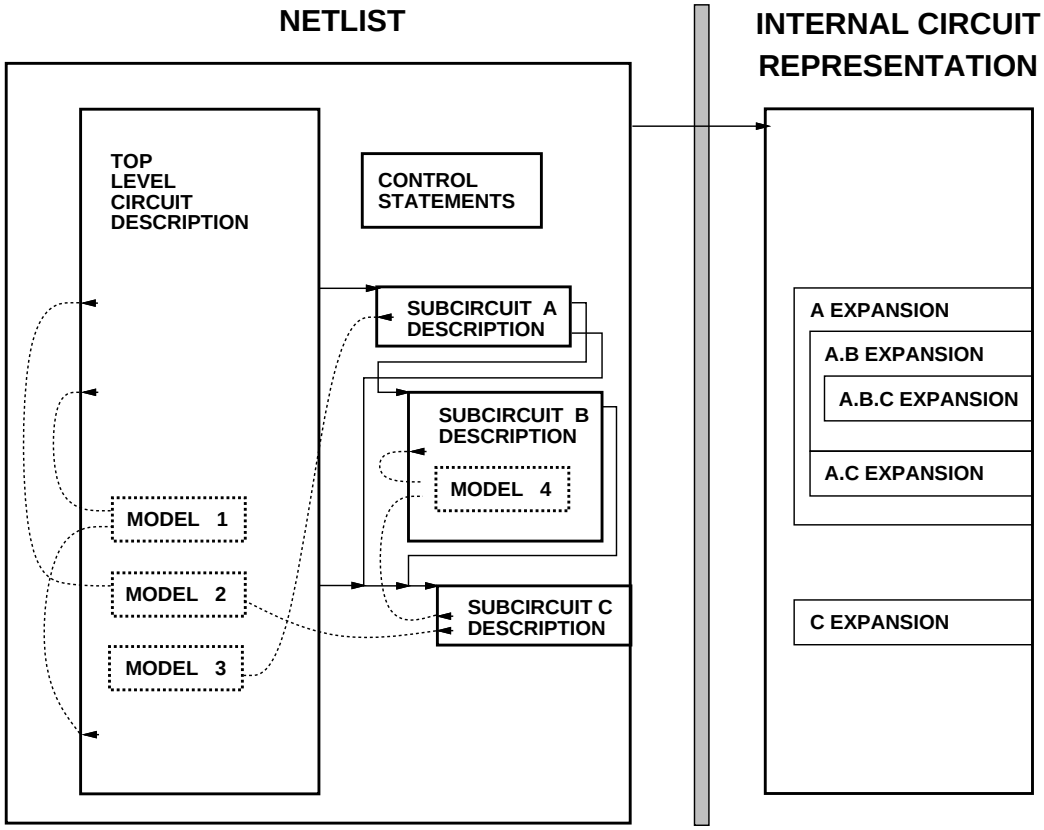


Figure 5.1: The SPICE circuit representation.

SPICE3 and PSPICE). In between the TITLE line and the .END statement can be any mix of statements — which control the operation of the simulator and the analysis to be performed; optional comment lines — for documenting the input file; and element lines — which specify the circuit elements. The input file can in fact contain no statements and the simulator will then perform an operating point .OP analysis. SPICE does not distinguish between upper and lower case characters.

Except for the TITLE line which is the first line of the input file, the type of input lines is distinguished by the first character on the line: statements begin with a period "."; element lines begin with an alphabetic character "A–Z" — with the letter identifying the element type (e.g. R for a resistor); and comment lines begin with an asterisk "*". In older terminology, based on the original use of punched cards, statements are referred to as "*dot cards*" or "*statement cards*"; element lines are referred to as "*element cards*" or "*device cards*"; and comment lines as "*comment cards.*"

PSPICE allows for in line comments indicated by a semicolon ";". The semicolon and everything following it on the same line are ignored except for purpose of echoing the input file in the output log file.

5.3.1 Analysis Statements

An analysis statement identifies the type of analysis to be performed. Any combination of analyses may be specified. Reporting of the results of an analysis is controlled by the .PRINT, .PLOT and, in with PSPICEby the .PROBE control statements. If no analysis statement is included in the input file then an operating point analysis (.OP) is performed by default. A brief description of the analysis options and the page on which a complete description can be found as follows.

Obtains the small-signal circuit response as a function of frequency. The .AC analysis is one of several small signal AC analyses.

DC solutions over a range of input conditions (.DC).

Analyze the circuit for harmonic and intermodulation distortion. This analysis is available in SPICE2G6 and SPICE3 but is not available in PSPICE as it has proved to be unreliable. In SPICE2G6 the distortion analysis must be performed in conjunction with a .AC analysis.

The .FOUR control statement can be used to find the spectrum of any time-varying signal in a (.TRAN) transient analysis. The .FOUR statement is unlike other analysis statements as it does not initiate a simulation but interprets the result of a simulation initiated by the .TRAN sattement.

The Monte Carlo analysis is a statistical analysis of the circuit causing the circuit to be analyzed many times with a random change of model parameters (parameters in a .MODEL statement). The analyses specified in the .DC, .AC or .TRAN statements can be simulated multiple times.

Conduct a small-signal noise analysis as a function of frequency. In PSPICE this statement must be used in conjunction with a .AC statement.

DC solution for a particular input voltage/current condition. This is the default analysis if no analysis is specified in the input file. This is the default analysis if not analysis type is specified in the input file.

In this analysis the poles and zeros of the small signal ACtransfer function of a two-port is evaluated.

5.3.2 Control Statements

5.3.3 Elements

The general form for elements is device name, followed by a list of nodes, followed by the numeric value of the element, followed in some cases by the name of a model, and then by other keywords:

Name Node1 Node2 ... NodeN NumericValue ModelName
\+ *keyword=NumericValue ... InitialConditions.*

For some elements initial conditions (*InitialConditions*) can also be specified which can be used to ensure that the desired initial state of astable circuits is obtained and also to aid in convergence. The first letter of the *Name* identifies the element. For example, if *Name* is `RTEST` then the element is a resistor.

The general form above is not the form for every element. The way in which Spice evolved resulted in the syntax for element lines not being fully consistent. Commercial extensions, as with PSpice, allowing alphabetic names for nodes rather than just an integer designation also result in syntaxical problems. With this extension it is not possible to use the fact that a field was alphabetic to distinguish between a node and a parameter name. However this change has necessitated no change to the the standard syntax as defined by Spice2g6. The problem appears in conjunction with other extensions which allow for an arbitrary number of nodes in some statements. The result is that the syntax can be slightly different than would be expected. For these reasons the description of the form of a particular element or statement must be consulted to ensure that the syntax is correct.

Passive Elements

The passive devices supported in Spice2g6, Spice3 and PSpice and where their descriptions can be found are as follows:

Active Elements

Form

`Q`*name NCollector NBase NEmitter* [*NSubstrate*] *ModelName* [*Area*] [`OFF`]
`+` [`IC=`*Vbe,Vce*]

Unlike passive devices active devices can not be specified by one or a few parameter values. Since many of the parameter values are the same for many devices it is convenient to specify them in a `.MODEL` statement that can be reused many times. All active elements require a `.MODEL` statement and most allow an optional substrate node to be used on the element line.

The active devices supported in SPICE2G6, SPICE3 and PSPICE and where their descriptions can be found are as follows:

5.3.4 Distributed Elements

The distributed devices and where they are described are as follows:

The convolution element enables a linear circuit described by a set of frequency dependent complex y parameters to be included in transient analysis.

5.3.5 Source Elements

The sources supported and where they can be found are as follows:

The control of the E, F, G and H elements can be control by a polynomial function of voltage or current.

5.3.6 Interface Elements

The interface elements supported and where they can be found are as follows:

Interfaces digital analog simulation by providing a means for a state transistion to control an analog response.

Determines the equivalent digital state of an analog signal.

Element enabling the scattering parameters of a circuit to be directly calculated. (Available in only a few versions of Spice.

Interfaces a circuit (or subcircuit) to a subcircuit.

5.4 Input Grammar

Each input line contains fields which are delimited (separated) by one of a number of characters. The most obvious delimiter is simply a space but other characters are also treated as "*white space*" characters. White space is defined as one or any combination of the following characters:

"blank" "tab" ( ) , =

While the above characters appear in the input file they are ignored except that they are treated as a field delimiter. The "(", ")", "=" and "," characters are often used in the input file and are included in specifiying the syntax for elements and statements but they serve only to add visual structure to the input.

Continuation lines begin with a plus "+" in the first column. For example

```
R1 1 2 1000
```

and

```
R1 1 2
+1000
```

are equivalent.

Comment lines begin with an asterisk "*" in the first position. In line comments are supported in PSpice and these begin with a semicolon ";" and must be contained wholly on one line. For example

```
* This just shows off a comment
R1 1 2 1000 ;This shows of an in line comment
```

In nearly every situation where a numeric value is required in the input an algebraic expression can be used instead. Everything between a "{" character and the matching "}" character is treated as an algebraic expression. The evaluation of algebraic expressions are discussed in the ALgebraic Expressions section on on page **??**.

5.4.1 Prefixes and Units

Almost-standard metric prefixes are used in SPICE, the prefix abbreviation, the full metric name, and the represented scale factors being as follows:

SPICE **Prefix**	**Metric Name**	**Scale Factor**
F	femto	10^{-15}
P	pico	10^{-12}
N	nano	10^{-9}
U	micro	10^{-6}
M	milli	10^{-3}
K	kilo	10^{+3}
MEG	mega	10^{+6}
G	giga	10^{+9}
T	tera	10^{+12}

As SPICE does not differentiate between upper and lower case, '`MEG`' (or '`meg`') is used for 'mega' instead of the standard metric upper case 'M'.

The value of an element is specified in terms of the conventionally accepted units, e.g. resistance in Ohms, capacitance in Farads, and inductance in Henries. If you wish you can spell it out more fully, e.g.

```
C1 0 2 5fF          or
C1 0 2 5fFarad      or even
C1 0 2 5fthingies
```

The last alternative is allowed as SPICE actually ignores whatever follows the '`f`' and assumes Farads.

5.5 Parameters

Parameters can be defined in two ways:

- In a parameter definition (`.PARAM`).
- As a subcircuit parameter in a `.SUBCKT` statement.

Parameters defined in a `.PARAM` statement can be used in subsequent statements and element lines by replacing a numeric value by an expression in which the parameter is used. The general form of the `.PARAM` statement is

```
.PARAM [ParameterName = NumericValue ... ]
+ [ParameterName = { Expression } ... ]
```

Here the *ParameterName* is the name of a parameter with the first character being alphabetic (a-zA-Z) and can be assigned a numeric value *NumericValue* which may be followed immediately by a spice scale factor. For example, SMALL=1.E-9, SMALL=+1N, SMALL=1NV and SMALL=1.E-9V are equivalent and all establish a parameter SMALL with a value of 10^{-9}. If *ParameterName* is the name of a previously defined parameter at the same level of subcircut expansion then the parameter value is changed. If the `.PARAM` statement is is in the top level circuit then the parameter value is global and is available any where in the netlist. If the `.PARAM` statement is in a subcircuit then the parameter value is local and can be used at the current subcircuit expansion level or lower in the subcircuit expansion hierarchy. The same idea applies to values of a parameter changed in a subcircuit. Value changes are local and are available in the current subcircuit and lower nested subcircuits. Libraries are searched for parameters not defined in the circuit NETLIST or in included files. A `.PARAM` statement does not have to be within a subcircuit in a library.

Instead of a numeric value an algebraic expression can be used to establish the value of the parameter. The expression is evaluated in the standard way for an algebraic expression replacing numeric values and is evaluated at the time of expansion rather than as the netlist is read. This ensures the correct hierarchical interpretation of the netlist. The treatment of expression is discussed in section 5.6 on page 46. Note that as always the expression must be enclosed in matching braces ({ ... }).

Parameters can be used nearly anywhere a numeric value is expected by including them in an expression evaluation even if the expression contains a single parameter. For example

```
.PARAM rbig=10MEG
R1 1 2 {RBIG}
```

establishes a resistance `R1` between nodes 1 and 2 with a value of 10^6 Ω.

Several predefined parameters are supported and the user must avoid defining these as unpredictable results may result. The predefined parameters are

Name	Value	Description
TEMP	not supported Reserved for future expansion	Analysis temperature.
VT	not supported Reserved for future expansion	Thermal voltage.

5.6 Expressions

In PSpice most places where a numeric value is normally used an expression (within braces { ... }) can be used instead. An expression can contain any supported mathematical operation, constant numeric values or expressions. Exceptions are

- Polynomial coefficients.
- The values of the transmission line device parameters `NL` and `F`.
- The values of the piece-wise linear characteristic in the `PWL` form of the independent voltage (`V`) and current (`I`) sources.
- The values of the resistor device parameter `TC`.
- As node numbers.

and

- Values of most statements (such as .TEMP, .AC, .TRAN etc.)

Specifically included are

- The values of all other device parameters.
- The values in `.IC` and `.NODESET` statements.
- The values in `.SUBCKT` statements.

and

- The values of all model parameters. `F`.

Operators that can be used in expressions are listed in Table 5.1.

Table 5.1: Expression operators.

Operator	Syntax	Description
PLUS	x+y	plus
MINUS	x-y	minus
UNARY_PLUS	+x	unary plus
UNARY_MINUS	-x	unary minus
MULTIPLY	x*y	multiply
DIVIDE	y/x	divide
POW	x^y or x**y	raise to a power, x^y
AND	x&y	AND
OR	x|y	OR
NOT	!x	NOT
XOR	x y	XOR (exclusive or)
SIN	sin(x)	sine, argument in radians
COS	cos(x)	cosine, argument in radians
TAN	tan(x)	tangent, argument in radians
ASIN	asin(x)	arcsine, argument in radians
ACOS	acos(x)	arccosine, argument in radians
ATAN	atan(x)	arctangent, argument in radians
SINH	sinh(x)	hyperbolic sine
COSH	cosh(x)	hyperbolic cosine
TANH	tanh(x)	hyperbolic tangent
EXP	exp(x)	exponentiation, e^x
ASINH	asinh(x)	arc-hyberbolic sine
ACOSH	acosh(x)	arc-hyberbolic cosine
ATANH	atanh(x)	arc-hyberbolic tangent
ABS	abs(x)	absolute, $\|x\|$
SQRT	sqrt(x)	square root, $\sqrt{x}$

5.6.1 Polynomials

Polynomial expressions can be used with the controlled source elements (E, F, G and H) to realize nonlinear controlled sources. The specification of the polynomial must be at the end of the input line and has two forms. The polynomial format for a voltage-controlled current source (the G element) or a voltage-controlled voltage source (the E element) is

`POLY(`N`)` `(`N_{C1+}`,`N_{C1-}`)` ... `(`N_{CN+}`,` N_{CN-}`)` C_0 C_1 C_2 C_3 ... where

`POLY` is the keyword indicating that a polynomial description follows.

N is the degree of the polynomial.

N_{C1+}, N_{C1-} The voltage at the node N_{C1+} with respect to the voltage at the node N_{C1-} is the controlling voltage V_1.

N_{CN+}, N_{CN-} The voltage at the node N_{CN+} with respect to the voltage at the node N_{CN-} is the controlling voltage V_N.

C_0 C_1 ... are the polynomial coefficients. Not all of the coefficients need be specified as the trailing coefficients that are not specified are treated as if they are zero.

Note that in spice parentheses, "(" and ")", and commas, ",", are treated as if they are spaces. The use of parentheses and commas serves only to make the netlist more easily read. **The exception to this is their use in expressions (see section 5.6).**

For voltage-controlled elements the output is calculated as

$$\begin{aligned}
\text{OUTPUT} \;=\; & C_0 \\
& +C_1V_1 + \ldots + C_NV_N \\
& +C_{N+1}V_1V_1 + C_{N+2}V_1V_2 + \ldots + C_{N+N}V_1V_N \\
& +C_{2N+1}V_2V_2 + C_{2N+2}V_2V_3 + \ldots + C_{2N+N-1}V_2V_N \\
& \vdots \\
& +C_{N!/(2(N-2)!)+2N}V_NV_N \\
& +C_{N!/(2(N-2)!)+2N+1}V_1V_1V_1 + C_{N!/(2(N-2)!)+2N+2}V_1V_1V_2 \\
& \qquad + \ldots + C_{N!/(2(N-2)!)+2N+N-1}V_1V_1V_N \\
& +C_{N!/(2(N-2)!)+3N}V_1V_2V_2 + \ldots + C_{N!/(2(N-2)!)+3N+N-2}V_1V_2V_N \\
& \vdots
\end{aligned}$$

A one dimensional polynomial (with only one pair of controlling nodes) is evaluated as

$$\text{OUTPUT} = C_0 + C_1V_1 + C_2V_1^2 + C_3V_1^3 + \ldots C_NV_1^N$$

An example of a voltage-controlled voltage source is

```
E1 2 3 POLY(2) (10,0) (12,2) 0.5 1 1 0.2 0.3 0.2
```

and of a voltage-controlled current source is

```
G1 2 3 POLY(4) (10,0) (12,2) (11,0) (13,0) 0.5 1 1 1 1 0.2 0.3 0.2
```

The format for a current-controlled current source (the `F` element) or a current-controlled voltage source (the `H` element) is

`POLY(`*N*`)` *VoltageSourceName*$_1$ ... *VoltageSourceName*$_N$ C_0 C_1 C_2 C_3 ... where

`POLY` is the keyword indicating that that a polynomial description follows.

N is the degree of the polynomial.

VoltageSourceName$_1$ is the name of the voltage source the current through which is control current I_1.

VoltageSourceName$_N$ is the name of the voltage source the current through which is control current I_N.

C_0 C_1 ... are the polynomial coefficients.

For these elements the output is calculated as

$$\begin{aligned}
\text{OUTPUT} \;=\; & C_0 \\
& + C_1 V_1 + \ldots + C_N V_N \\
& + C_{N+1} V_1 V_1 + C_{N+2} V_1 V_2 + \ldots + C_{N+N} V_1 V_N \\
& + C_{2N+1} V_2 V_2 + C_{2N+2} V_2 V_3 + \ldots + C_{2N+N-1} V_2 V_N \\
& \vdots \\
& + C_{N!/(2(N-2)!)+2N} V_N V_N \\
& + C_{N!/(2(N-2)!)+2N+1} V_1 V_1 V_1 + C_{N!/(2(N-2)!)+2N+2} V_1 V_1 V_2 \qquad (5.1) \\
& \quad + \ldots + C_{N!/(2(N-2)!)+2N+N-1} V_1 V_1 V_N \\
& + C_{N!/(2(N-2)!)+3N} V_1 V_2 V_2 + \ldots + C_{N!/(2(N-2)!)+3N+N-2} V_1 V_2 V_N \\
& \vdots \qquad (5.2)
\end{aligned}$$

An example of a current-controlled voltage source is:

H1 2 3 POLY(2) VIN V2 0.5 1 1 0.2 0.3 0.2

and of a current-controlled current source is:

F1 2 3 POLY(4) VIN V2 (11,0) (13,0) 0.5 1 1 1 1 0.2 0.3 0.2

5.6.2 Laplace Expressions

5.6.3 Chebyschev

5.7 Function Definition .FUNC PSpice Only

The `.FUNC` statement can be used to conveniently define commonly used expressions.

`.FUNC` FunctionName`(` [Argument1, Argument2, ... Argument10]]] `)` = Function-Declaration

FunctionName is the name of the function being defined. It must begin with an alphabetic character (A-Z).

Argument1 is a function argument. There can be from 0 to 10 arguments.

FunctionDeclaration can be any regular algebraic expression (see section **??** on page **??**) and can use previously defined functions and the Laplace variable *s*. The expression delimiters { and } need not be used. The *FunctionDeclaration* is automaticly enclosed within the expression delimiters { and }. The function declaration plus the two delimiters must be no more than 80 characters (one line) long.

The names of predefined functions msut be avoided. The predefined functions are listed in section **??** on page **??**.

Functions are treated as macros in the C programming language. when user defined functions are invoked a textual expansion is performed and the resultant expansion is evaluated as a regular expression. The FunctionDeclaration before and after expansion is enclosed within expression delimiters { and }. This defines how nested functions are treated.

It is faster to use predefined functions if available. Predefined functions also test the validity of the arguments and evaluate the correct asymptotic behavior.

5.8 Syntax Variations

Commercial versions have enhanced the syntax of Berkeley version of SPICE. In virtually all cases the syntax of SPICE2G6 and SPICE3is a subset of the syntax of commercial versions of SPICE. Here we list some exceptions.

- Units. Spice does not allow units immediately following a quantity. For example, the following is acceptable in all versions.
  ```
  VIN 1 0 DC 4
  VIN 1 0 DC 4UV
  ```
  For example, the following is not acceptable in all SPICE2G6 and SPICE3
  ```
  VIN 1 0 DC 4V
  ```
  but is acceptable in HSPICE, PSPICEand SOMEVERSIONSOFSPICE.

Chapter 6

Statement Catalog

Conventions

Square brackets "[. . .] "indicate an optional quantity.

Italics indicate a quantity that is replaced by a specific value.

.AC AC Analysis

The `.AC` statement initializes an `AC` small-signal analysis sweeping the frequency of the independent voltage and current sources.

Form

```
.AC DEC FrequenciesPerDecade FStart FStop
.AC OCT FrequenciesPerOctave FStart FStop
.AC LIN NumberPoints FStart FStop
```

`DEC` is the decade sweep keyword specifying that the frequency F_1 is to be swept logarithmically by decades.

FrequenciesPerDecade specifies the number of frequencies per decade.

`OCT` is the octave sweep keyword specifying that the frequency is to be swept logarithmically by octaves.

FrequenciesPerOctave specifies the number of frequencies per octave.

`LIN` is the linear sweep keyword specifying that the frequency is to be swept linearly.

NumberPoints specifies the total number of frequencies in a linear sweep.

FStart is the starting frequency of the frequency sweep.
(Units: Hz; Required; $FStart > 0$)

FStop is the stopping frequency of the frequency sweep.
(Units: Hz; Required; $FStop \geq FStart$)

Example

```
.AC DEC 10 1kHz 100Mhz
.AC DEC 10 1kHz 100Mhz
```

Note

1. A `DC` analysis is automatically performed prior to an `AC` small-signal analysis to find the operating point. Using the `DC` values of voltage and current at the operating point the linearized, small-signal models of nonlinear devices are determined.
2. In `AC` analysis voltage sources without `AC` specifications are shorted and current sources without `AC` specifications are opened.

COMMENT — Comment Card

Used to insert a comment in the circuit NETLIST.

Form

* *a comment string*

Note

1. A comment line begins with an asterisk "*" in the first position of the line. There can be no leading white space.
2. Comment lines can appear anywhere in the input file and are ignored except that they are echoed in the output log file.
3. In PSpiceComments can appear anywhere on following a semicolon ";". The remainder of the line (following the ";") is ignored except for purposes of echoing the input NETLIST in the log file.

.DC DC Analysis

In `DC` analysis the `DC` operating point of a circuit is determined for a range of values of up to two independent voltage or current sources.

Form

`.DC` *SourceName1 StartValue1 StopValue1 ValueIncrement1*
`+` [*SourceName2 StartValue2 StopValue2 ValueIncrement2*]

PSPICE*Form*

`.DC` [`LIN`] *SweepVariableName1 StartValue1 StopValue1 ValueIncrement1*
`+` [*SourceName2 StartValue2 StopValue2 ValueIncrement2*]

`.DC OCT` *SweepVariableName1 StartValue1 StopValue1 PointsPerOctave1*
`+` *SweepVariableName2 StartValue2 StopValue2 PointsPerOctave2*]

`.DC DEC` *SweepVariableName1 StartValue1 StopValue1 PointsPerDecade1*
`+` *SweepVariableName2 StartValue2 StopValue2 PointsPerDecade2*]

`.DC` *SweepVariableName1* `LIST` $Value_{1,1}$ [$Value_{1,2}$ ... $Value_{1,N}$]
`+` [*SweepVariableName2* `LIST` $Value_{2,1}$] $Value_{2,2}$ ... $Value_{2,N}$]]

SourceName1 is the name of the first independent voltage (`V` element) or current (`I` element) source the value of which will be swept.

StartValue1 is the starting value of the sweep of the first voltage or current source.

StopValue1 is the final value of the sweep of the first voltage or current source.

ValueIncrement1 is the increment by which the value of the first voltage or current source is incremented.

SourceName2 is the name of the first independent voltage (`V` element) or current (`I` element) source the value of which will be swept.

StartValue2 is the starting value of the sweep of the second voltage or current source.

StopValue2 is the final value of the sweep of the second voltage or current source.

ValueIncrement2 is the increment by which the value of the second voltage or current source is incremented.

`LIN` is the linear sweep keyword. This is the default sweep type.

`OCT` is the octave sweep keyword specifying that the sweep variable or variables is to be swept logarithmically by octaves.

`DEC` is the decade sweep keyword specifying that the sweep variable or variables is to be swept logarithmically by decades.

PointsPerOctave1 is the number of points per octave in a `OCT` sweep type for the first sweep.

PointsPerDecade1 is the number of points per decade in a `DEC` sweep type for the first sweep.

PointsPerOctave2 is the number of points per octave in a `OCT` sweep type for the second sweep.

PointsPerDecade2 is the number of points per decade in a `DEC` sweep type for the second sweep.

SweepVariableName1 is the name of the first sweep variable. The sweep variable can be:

1. the name of an independent voltage or current source. The *DCvalue* of the source is swept.
2. the name of a parameter of a specific model specified in the form *ModelName*(*ParameterKeyword*). For example to sweep the `IS` parameter of an `NPN` model of name `MYNPN` the *SweepVariableName* would be `MYNPN(IS)`.
3. the keyword `TEMP` which indicates that the analysis temperature in °C is swept. The model parameters are updated for each sweep value.

SweepVariableName2 is the name of the second sweep variable. The properties are as for *SweepVariableName1* as described above.

`LIST` indicates that the value of the *SweepVariable* will take the values, in order, in the following list rather than be swept.

$Value_{i,j}$ the jth value to be assigned to the ith sweep variable.

Example

```
.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
```

PSpice *Example*

```
.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
.DC LIN VCE 0 10 .25 IB 0 10U 1U
.DC DEC MYNPN(IS) 1.E-15 1.e-17 3
.DC TEMP -25 0 25 50 75 100 VIN 0.25 5.0 0.25
```

Note

1. The `.DC` statement initiates a `DC` operating point analysis.

2. In the `DC` analysis inductors are shorted and capacitors are open circuited.

3. A `DC` analysis over a range of source conditions can be used to produce the transfer characteristic of a circuit or current-voltage characteristics of a semiconductor device such as a transistor. An example of determining the transfer characteristic of an operational amplifier is given in section 8.5.1. An example of determining the current-voltage characteristics of a transistor is in section 8.5.1.

3 A `DC` analysis over a range of source conditions can also be used to provide biasing information in circuit design.

4 The specified independent voltage or current source or sources are stepped over a user-specified range and the `DC` output variables indicated by the `.PRINT` statement are stored for each source value.

5 When two sources are specified the first source is swept over its range for each value of the second source. For example, consider

```
.DC VDS 0 10 5 VGS 0 5 2.5
```

`VGS` is swept from 0 to 5V in 2.5V increments. For each value of `VGS` `VDS` is swept from 0 to 10V in 5V increments so that that 9 `DC` analyses performed are

RUN	`VDS`	`VGS`
1	0	0
2	5	0
3	10	0
4	0	2.5
5	5	2.5
6	10	2.5
7	0	5
8	5	5
9	10	5

6 When the sweep is completed the original values are restored.

7 The sweep can go in either direction. That is, *StartValue* can be less than or greater than *StopValue* but *ValueIncrement* must be positive.

.DISTO Small-Signal Distortion Analysis

In distortion analysis the steady-state harmonic and intermodulation products for small input signal are computed as a part of an `AC` analysis. One or two excitation frequencies, F_1 and F_2 may be specified. If only one excitation frequency, F_1, is specified the program evaluates the second and third harmonic distortions. If a second excitation frequency, F_2, is specified the three lowest order intermodulation distortion components are evaluated as well.

Form

`.DISTO` *ResistorName* [*OutputInterval* [F_2*OverF*$_1$ [F_1*ReferencePower*
`+` [F_2*ReferencePower*]]]]

SPICE3*Form*

`.DISTO DEC` *FrequenciesPerDecade* F_1*Start* F_1*Stop* [F_2*OverF*$_1$]
`.DISTO OCT` *FrequenciesPerOctave* F_1*Start* F_1*Stop* [F_2*OverF*$_1$]
`.DISTO LIN` *NumberPoints* F_1*Start* F_1*Stop* [F_2*OverF*1]

ResistorName the name of the output resistor. The power dissipated in this resistor is reported as the distortion measures. SPICE2G6 only.

OutputInterval is the optional output reporting interval at which distortion components produced by all nonlinear components is reported. By default, or if omitted no detailed output is produced. SPICE2G6 only.

F_1*ReferencePower* power level of F_1. SPICE2G6 only.
(Units: W; Optional; Default: 1.0E-3 (i.e. 1 mW or 1 dBm))

F_2*ReferencePower* power level of F_2. SPICE2G6 only.
(Units: W; Optional; Default: 1.0E-3 (i.e. 1 mW or 1 dBm))

`DEC` is the decade sweep keyword specifying that the frequency F_1 is to be swept logarithmically by decades.

FrequenciesPerDecade specifies the number of frequencies per decade.

`OCT` is the octave sweep keyword specifying that the frequency F_1 is to be swept logarithmically by octaves.

FrequenciesPerOctave specifies the number of frequencies per octave.

`LIN` is the linear sweep keyword specifying that the frequency F_1 is to be swept linearly.

NumberPoints specifies the total number of frequencies in a linear sweep.

F_1*Start* is the starting frequency of the F_1 sweep.

F_1*Stop* is the stopping frequency of the F_1 sweep.

F_2*OverF*$_1$ is the ratio of F_2 to F_1.
In SPICE3 , if F_2*OverF*$_1$ is omitted an harmonic analysis only is reported. Otherwise F_2 =F_2*OverF*$_1$ F_1*Start*
In SPICE2G6 , if F_2*OverF*$_1$ is omitted it defaults to 0.9. F_2 =F_2*OverF*$_1$ F_1

$F_2 OverF_1$ should be an irrational number between 0.0 and 1.0. If it is a rational number the signals at F_1 and F_2 are harmonically related and the spectral analysis may in error. Since an irrational number can not actually be specified care should be excercized in choosing $F_2 OverF_1$ so that F_1 and F_2 are not simple multiples of each other. For example, use 0.498 instead, if you want to set $F_2 OverF_1$ to 0.5. The rule to follow is to keep the denominator in the fractional representation of $F_2 OverF_1$ as large as possible with at least three digits for accurate results.

Example

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

Note

1. In SPICE2G6 the distortion analysis must be performed in conjunction with an `AC` analysis. The `.AC` statement (see page 53) specifies the sweep parameters for F_1.

2. In SPICE3 the distortion analysis is also performed in conjunction with the `AC` analysis specified by the `.AC` statement. The amplitudes and relative phases of the input distortion components are specified in the `.AC statement` as the arguments of the `DISTOF1` keyword for F_1 and of the `DISTOF2` keyword for F_2. This enables several sources to generate components at F_1 and/or F_2. If the DISTOF1 or DISTOF2 keywords are missing from a source then this source is assumed to have no input at F_1 or F_2 respectively.

   When a spectral analysis is performed (both F_1 and F_2 specified) the circuit is treated as having sinusoidal inputs at two different frequencies F_1 and F_2. F_1 is swept but F_2 is fixed at $F_2 OverF_1 FStart$. Each independent source in the circuit can have two inputs at F_1 and F_2 for distortion analysis.

Note

3 Distortion analysis is not valid if switches (if present) change state under the small excitations used for distortion calculations.

4 Evaluation of the small-signal distortion of a circuit is based on a multi-dimensional Volterra series analysis. The nonlinearities are expanded in a third order multi-dimensional Taylor series around the operating point determined from an operating point (`DC`) analysis. Using this analysis the distortion components are evaluated symbolically

5 In SPICE2G6 the following distortion components are evaluated.

HD2	-	the magnitude of the second harmonic at frequency $2F_1$ assuming that F_2 is not present.
HD3	-	the magnitude of the third harmonic at frequency $3F_1$ assuming that F_2 is not present.
SIM2	-	the magnitude of the sum frequency $F_1 + F_2$
DIM2	-	the magnitude of the difference frequency $F_1 - F_2$
DIM3	-	the magnitude of the third order intermodulation frequency $2F_1 - F_2$

1. In SPICE3 the complex values of the above distortion components are computed at all nodes in the circuit. The distortion components at any node can be reported using the `.PRINT` statement (discussed on page 92) or the `.PLOT` statement (discussed on page 88). The running variable in the output is the frequency F_1.

2. The quantities reported are the actual `AC` voltages and currents and must be normalized by the user to the sources at F_1 and F_2 to obtain true distortion measures.

.DISTRIBUTION Distribution Specification

The `.DISTRIBUTION` statement specifies the statistical tolerance distribution used in Monte Carlo analysis (see the `.MC` statement on page 69).

PSPICE*Form*

```
.DISTRIBUTION DistributionName (Deviation1, Probability1)
+ (Deviation2, Probability2)  [(Deviation3, Probability3) ... ]
```

DistributionName is the name the user assigns to the distribution defined by the succeeding values.

Deviation Relative deviation from nominal value.
($-1 \leq Deviation \leq 1$).

Probability Probability of preceding deviation.
($Probability \geq 0$)

Note

1. Parentheses, '(' and ')', and commas, ',', in the NETLIST are ignored and are generally used to make the NETLIST more readable.
2. The pairs of values (*Deviation*, *Probability*) define a piecewise linear probability distribution curve used in calculating the random numbers used in Monte Carlo analysis.
3. The *Deviation*s must be in ascending order:
$Deviation1 < Deviation2 < Deviation3 < \ldots$
4. 100 (*Deviation*, *Probability*) pairs can be specified.

.END End Statement

This statement indicates the end of the NETLIST of a circuit. While it is often inserted automatically by most SPICE simulators when the last statement of the NETLIST is read and it is not `.END` its usage is recommended as several circuits in one file are supported by some versions of SPICE including PSPICE.

Form

```
.END
```

.ENDS End Subcircuit Statement

The end subcircuit statement indicates the end of a subcircuit definition.

Form

`.ENDS` [*SubcircuitName*]

SubcircuitName Indicates which subcircuit is being terminated. If omitted, all subcircuits being defined are terminated. Its use is required when nested subcircuit definitions are being made. It is good practise to use the full form of `.ENDS`.

Example

```
.ENDS NAND_GATE
```

Note

1. See the definition of `.SUBCKT` on page 103.

.FOUR Fourier Analysis

The Fourier analysis statement initiates a Fourier analysis of the results of a transient analysis.

Form

`.FOUR` *Frequency OutputSpecification* [*OutputSpecification* ...]

Frequency specifies the fundamental frequency of the transient waveform to be analyzed. It is used to determine the period of the waveform.

OutputSpecification specifies the quantity to be reported as the result of the Fourier Analysis. It has the same format as the *OutputSpecification* in a `.PRINT` statement (see page 92).

Example

```
.FOUR 1M V(10,2) V(5) I(VLOAD)
```

Note

1. Fourier analysis uses the transient output waveform in the time interval from `TSTOP` $- T$ to `TSTOP`. T is the period of the *Frequency* parameter ($T = 1/Frequency$) specified in the `.FOUR` statement. `TSTOP` is the final transient analysis time specified in the `.TRAN` statement (which is described on page 109.

2. Unlike most other analyses, a `.PRINT`, `.PROBE` or `.PLOT` statement is not required for the data to be reported.

.FUNC Function Definition

In this statement commonly used expressions can be defined as more convenient functions.

PSPICE*Form*

`.FUNC` *FunctionName*`(` [*Argument1, Argument2, ... Argument10*]]] `)` = *Function-Declaration*

FunctionName is the name of the function being defined. It must begin with an alphabetic character (A-Z).

Argument1 is a function argument. There can be from 0 to 10 arguments.

FunctionDeclaration can be any regular algebraic expression (see section **??** on page **??**) and can use previously defined functions and the Laplace variable *s*. The expression delimiters { and } need not be used. The *FunctionDeclaration* is automatically enclosed within the expression delimiters { and }. The function declaration plus the two delimiters must be no more than 80 characters (one line) long.

Example

```
.FUNC
```

Note

1. The names of predefined functions must be avoided. The predefined functions are listed in section **??** on page **??**.
2. Functions are treated as macros in the C programming language. when user defined functions are invoked a textual expansion is performed and the resultant expansion is evaluated as a regular expression. The FunctionDeclaration before and after expansion is enclosed within expression delimiters { and }. This defines how nested functions are treated.
3. It is faster to use predefined functions if available. Predefined functions also test the validity of the arguments and evaluate the correct as asymptotic behavior.

.IC Initial Conditions

The `.IC` statement is used to set initial conditions for transient analysis. It has no effect on other types of analyses.

Form

`.IC V(`*NodeName1*`)=`*Voltage1* [*V(NodeName2*`) =`*Voltage2* ...]

`V` is the keyword specifying a node voltage

NodeName is the name of a node. Note that in SPICE2G6 and SPICE3 *NodeName* must be an integer.

Voltage is a numeric value.

Example

```
.IC V(11)=4.9 V(2)=2.5
```

Note

1. Initial conditions can be specified for the following elements: `B` (MESFET), `C` (capacitor), `D` (diode), `Q` (BJT), `M` (MOSFET) and `J` (JFET)
2. The `.IC` statement has two different effects depending on whether the `UIC` (use initial conditions) keyword is present on the `.TRAN` statement.
   (a) If the `UIC` keyword is specified in the `.TRAN` statement:
   The initial conditions specified in the `.IC` statement are used to establish the initial conditions. Initial conditions specified for individual elements using the `IC` parameter on the element line will always have precedence over those specified in a `.IC` statement.
   No `DC` analysis is performed prior to a transient analysis. Thus it is important to establish the initial conditions at all nodes using the `.IC` statement or using the `IC` element parameter.
   (b) If the `UIC` keyword is not specified in the `.TRAN` statement:
   A `DC` analysis is performed prior to a transient analysis. During the `DC` analysis the node voltages indicated in the `.IC` statement are held constant at the initial condition values. During transient analysis the nodes are not constrained to the initial condition values.

.INC Include Statement

Form

`.INC` *FileName*

Filename is the name of the file which is to be included.

Note

1. The contents of *Filename* are read as if it were part of the original file.
2. Libraries could be included using the `.INC` statement or by the `.LIB` statement discussed on page 68. The difference is that the `.INC` statement includes the contents of the library file (except for comments) in internal data structures. However a `.LIB` statement causes the library file to be scanned and an index constructed for model (`.MODEL`) and subcircuit (`.SUBCKT`) statements. The models subcircuits in the library file are included in internal data structures if they are referred to. Thus the use of `.LIB` statements is a much more efficient way of using library files leading to both a smaller program and faster library access, especially for large libraries. Library files can only contain models and subcircuits so that the type of files that can be incorporated using a `.LIB` statement is more restricted than the type of file that can be incorporated using a `.INC` statement.

.LIB Library Statement

The `.LIB` statement is an efficient way to include `.MODEL` statements and subcircuits.

Form

`.LIB` [*FileName*]

Filename is the name of the library file. (Optional; Default: NOM.LIB)

Note

1. The library file can only contain a restricted set of SPICE statements. It must contain only `.MODEL` statements, subcircuit definitions (between `.SUBCKT` and `.ENDS` statements), and `.LIB` statements.
2. The library file *Filename* is searched in the current directory and then in a list of directories specified by the environment variable `PSPICELIB` (for compatibility purposes the environment variable `PSPICELIB` is also supported).
3. If the DOS operating system is being used the library environment specification has the form

   `SET PSPICELIB =` *Directory1* [; *Directory2* ...]

   The environment variable may be set in the `AUTOEXEC.BAT` file in the root DOS directory or before PSPICE is evoked. For example:

   ```
   SET PSPICELIB = C:\SPICE\MY_LIB;D:\SPICE\TI_LIB
   ```

4. If the UNIX operating system is being used the library environment specification has the form

   `setenv PSPICELIB =` *Directory1* [; *Directory2* ...]

   For example:

   ```
   setenv PSPICELIB =  /SPICE/MY_LIB; /SPICE/TI_LIB
   ```

5. Libraries could be included using the `.INC` statement or by the `.LIB` statement. The difference is that the `.INC` statement includes the contents of the library file (except for comments) in internal data structures. However a `.LIB` statement causes the library file to be scanned and an index constructed of model (`.MODEL`) and subcircuit (`.SUBCKT`) statements. The models subcircuits in the library file are included in internal data structures only if they are referred to. Thus the use of `.LIB` statements is a much more efficient way of using library files leading to both a smaller program and faster library access, especially for large libraries.

.MC Monte Carlo Analysis

The Monte Carlo analysis is a statistical analysis of the circuit causing the circuit to be analyzed many times with a random change of model parameters (parameters in a `.MODEL` statement).

PSPICE*Form*

```
.MC NumberOfRuns AnalysisType OutputSpecification OutputFunction [LIST]
+ [OUTPUT( OutputSampleType )]
```

PSPICE*Form*

```
.MC NumberOfRuns AnalysisType OutputSpecification OutputFunction [LIST]
+ [OUTPUT( OutputSampleType )] [RANGE(LowValue, HighValue)]
+ [SEED=SeedValue]
```

NumberOfRuns is the total number of runs to do. This number includes the initial nominal run.

AnalysisType is the type of analysis to be performed in the Monte Carlo runs after the initial nominal run (using the nominal values of model parameters). All analyses specified in the `NETLIST` are performed in the nominal run. The *AnalysisType* must be one of the following:

`DC` is a keyword indicating that the `DC` analysis as specified by the `.DC` statement is repeated. The sweep variable used in analyzing the output *OutputSpecification* is the value of the independent voltage or current source specified in the `.DC` statement (as discussed on page 55).

`AC` is a keyword indicating that the `AC` small-signal analysis as specified by the `.AC` statement is repeated. The sweep variable used in analyzing the output *OutputSpecification* is frequency.

`TRAN` is a keyword indicating that the transient analysis as specified by the `.TRAN` statement is repeated. The sweep variable used in analyzing the output *OutputSpecification* is time.

OutputSpecification specifies the quantity to be reported as the result of the Monte Carlo Analysis. It has the same format as the *OutputSpecification* in a `.PRINT` statement (see page 92). The result is the value of the *OutputSpecification* with respect to a sweep for `DC` and `AC` analysis, and as a waveform for `TRAN analysis`.

OutputFunction indicates the function to be performed on the output indicated by *OutputSpecification* to reduce the sweep or waveform at each run to a single numeric value. The *OutputSpecification* must be one of the following keywords:

`YMAX` which produces the greatest deviation of the sweep or waveform from the nominal run.

`MAX` which results in the maximum value in each sweep or waveform.

`MIN` which results in the minimum value in each sweep or waveform.

`RISE_EDGE(`*Value*`)` which reports as the result the first run when the waveform crosses above the threshold *Value*. The algorithm used requires that one point in the waveform be below *Value* and the succeeding point be above *Value*.

`FALL_EDGE(`*Value*`)` which reports as the result the first run when the waveform crosses below the threshold *Value*. The algorithm used requires that one point in the waveform be above *Value* and the succeeding point be below *Value*.

`LIST` is an optional keyword that results in the model parameter values that are statistically varied being printed out prior to each run. If it is omitted then the statistically generated model parameter values are not produced prior to each run.

`OUTPUT` is an optional keyword indicating the type of output to be produced by runs after the initial nominal run. The output produced for each run sampled is determined by the `.PLOT`, `.PRINT` and `.PROBE` statements in the NETLIST. If this keyword is missing output is produced only for the nominal run.

OutputSampleType indicates the method by which runs are selected for output reporting. The output produced for each run selected is determined by the `.PLOT`, `.PRINT` and `.PROBE` statements in the NETLIST. *OutputSampleType* must be one of the following:

`ALL` indicates that the output is to be produced for all runs.

`FIRST` *Nruns* indicates that the output is to be produced only for the first *Nruns* runs.

`EVERY` *NthRun* indicates that the output is to be produced for *NthRun*th run.

`RUNS` *Run1* [*Run2 ...* [*Run25*]] indicates that the output is to be produced for the indicated runs. Up to 25 runs can be indicated.

`RANGE` is an optional range indicating the range of the sweep variable over which *OutputFunction* is to be performed. If this keyword is missing, output is produced but the range is not restricted. The range of the sweep variable to be considered is from *LowValue* to *HighValue* inclusive.

LowValue is the low end of the sweep variable to be considered in evaluating *OutputFunction*.

HighValue is the low end of the sweep variable to be considered in evaluating *OutputFunction*.

`SEED` is the keyword for the seed of the random number generator used in Monte Carlo Analysis

SeedValue is the value of the seed used in the random number generator used to select sample runs at random. (Optional; Default: 17,533; $1 \leq$ *SeedValue*$\leq 32,767$)

Note

1. A typical use of Monte Carlo analysis is to predict yield of a circuit by examining the effect of process variations such as length and width of transistors.

2. A `.TEMP` statement also can result in multiple circuit simulations as the temperature is varied. In conjunction with a `.MC` statement a Monte Carlo analysis is performed for each temperature before the temperature is updated in a temperature sweep.

3. Only model parameters (parameters in a `.MODEL` statement) are varied.

4. If the *AnalysisType* is `DC` only one independent voltage or current source can be specified in the `.DC` statement (discussed on page 55).

5. The random number generator is the subtractive method generator described by Knuth [28, p. 171].

6. The initial run uses the nominal parameter values given in the NETLIST. Subsequent runs statistically vary model parameters indicated as having either lot `LOT` or device `DEV` tolerances. These tolerances are specified in a `.MODEL` statement (see page 72).

Method

The initial Run of a Monte Carlo Analysis uses the nominal values of the model parameters possibly updated to their new values if the analysis temperature is different from the the nominal temperature in effect when the `.MODEL` statement was read. The nominal values of the model parameters as well as their tolerances and type of statistical distribution are specified in a `.MODEL` statement (see page 72). If $X(T)$ is the value of the model parameter at temperature T, r is a random number between -1 and $+1$ inclusive, and X_T is the tolerance of $X(T_{\mathrm{NOM}})$ is the nominal temperature, the value of X to be used in Monte Carlo analysis is

$$X'(T) = X(T)(1 + rX_T) \tag{6.1}$$

The random number r has a statistical distribution which has a type indicated in the `.MODEL` statement. The distribution type must be either one of the built-in distributions or specified by the user in a `.DISTRIBUTION` statement (see page 61). The tolerance is specified immediately following the distribution in a .MODEL statement. Specification of the distribution type and tolerance assignment are described on page 74.

.MODEL Model Statement

Model statements specify the parameters of elements that either are too numerous to put on the element line or are common to many elements.

Form

`.MODEL` *ModelName ModelType* `(`*Keyword*=*Value ...* `)`

PSpice*Form*

`.MODEL` *ModelName ModelType* `(`*Keyword*=*Value* [*ToleranceSpecification*] ... `)`

ModelName is the name of the model specified by the user.

ModelType is the model type which is specific to particular elements. The model types are given in the table below.

ToleranceSpecification is the specification of the statistical distribution and tolerance of a parameter. This is used in a Monte Carlo simulation to assign random variations to the model parameter. *ToleranceSpecification* is discussed on page 74.

Keyword is the name of the model parameter which is discussed on the pages referred to below.

Value is the numeric value of the model parameter.

Example

```
.MODEL MOSFET1 MOS (LEVEL=2 VTO=-0.76 GAMMA=0.6 CGSO=3.35E-10)
```

MODEL TYPE	SPICE VERSION—	ELEMENT NAME	DESCRIPTION	PAGE
PASSIVE DEVICE MODELS				
RES		R	resistor model	236, 237
CAP		C	capacitor model	150, 150
IND	PSpice	L	inductor model	187
SEMICONDUCTOR DEVICE MODELS				
D		D	diode model	152
NPN		Q	NPN bjt model	228
PNP		Q	PNP bjt model	228
LPNP	PSpice	Q	Lateral PNP bjt model	228
NJF		J	N-channel junction FET (JFET)	175
PJF		J	P-channel junction FET (JFET)	176
GASFET	PSpice	B	N-channel GaAs MESFET	264, 118
NMF	Spice3	B	N-channel GaAs MESFET	
PMF	Spice3	B	P-channel GaAs MESFET	
NMOS		M	N-channel MOSFET	191
PMOS		M	P-channel MOSFET	191
SWMODEL	Spice3			**??**
URC	Spice3	T	Lossy RC transmission line	243
MISCELLANEOUS DEVICE MODELS				
CORE	PSpice	K	nonlinear, transformer (magnetic core)	183
VSWITCH	PSpice	S	voltage switch	240
ISWITCH	PSpice	W	current switch	254
SW	Spice3			**??**
CSW	Spice3			**??**
DISTRIBUTED DEVICE MODELS				
URC			Uniform Distributed RC model	243

MODEL TYPE	SPICE VERSION	ELEMENT NAME	DESCRIPTION	PAGE
DIGITAL INTERFACE AND DIGITAL DEVICE MODELS				
DINPUT	PSpice	N	digital input model	222
DOUTPUT	PSpice	O	digital output model	224
UIO	PSpice	U	digital I/O model	
UGATE	PSpice	U	standard gate	
UTGATE	PSpice	U	tri-state gate	
UEFF	PSpice	U	edge-triggered flip-flop	
UGFF	PSpice	U	gated flip-flop	
UWDTH	PSpice	U	pulse width checker	
USUHD	PSpice	U	setup and hold checker	
UDLY	PSpice	U	digital delay line	
UADC	PSpice	U	multi-bit analog-to-digital converter	
UDAC	PSpice	U	multi-bit digital-to-analog converter	

Virtually all of the device model parameters have default values which generally result in typical operation.

Immediately following the specification of a model parameter a statistical distribution and tolerance can be assigned. These are used in conjunction with Monte Carlo analysis which is controlled by the `.MC` statement described on page 69 The Monte Carlo analysis is a statistical analysis of the circuit causing the circuit to be analyzed many times with a random change of certain model parameters). The form of the tolerance and distribution assignment is

`DEV` [`/DeviceTrackingIndex`] [/*DistributionType*] Tolerance [`%`]
`+ LOT` [`/LotTrackingIndex`] [/*DistributionType*] Tolerance [`%`]

`DEV` is the keyword for the device tolerance specification.

DeviceTrackingIndex specifies which random number to use. It must be one of 1, 2, ... 10. Two parameters with the same *DeviceTrackingIndex* are correlated. Conversely if two parameters have different *DeviceTrackingIndex* then they are uncorrelated.
LotTrackingIndex and *DeviceTrackingIndex* refer to different random numbers and so choosing the same *LotTrackingIndex* and *DeviceTrackingIndex* does not correlate the `LOT` and `DEV` distributions.
(Units: none; Optional; Default: 1)

DistributionType is the type of the statistical distribution used in generating random numbers. There are two predefined distribution types and a user specified distribution type can be used. The allowable types are:

`UNIFORM` specifies a uniform distribution of random numbers between -1 and +1.

`GAUSS` specifies a Gaussian distribution of random numbers between $\pm 4\sigma$. In this case *Tolerance* specifies σ.

DistributionName is the name of a user specified distribution specified in a `.DISTRIBUTION` statement (see page 61).

LotTolerance is the lot tolerance specification. It may be specified as an absolute quantity or as a percentage if followed by %. It is converted to a fraction of the nominal value. The magnitude of the fractional value must be ≤ 1.

`LOT` is the keyword for the lot tolerance specification.

LotTrackingIndex specifies which random number to use. It must be one of 1, 2, ... 10. Two parameters with the same *LotTrackingIndex* are correlated. Conversely two parameters have different *LotTrackingIndex*'s then they are uncorrelated.
LotTrackingIndex and *DeviceTrackingIndex* refer to different random numbers and so choosing the same *LotTrackingIndex* and *DeviceTrackingIndex* does not correlate the `LOT` and `DEV` distributions.
(Units: none; Optional; Default: 1)

DeviceTolerance is the device tolerance specification. It may be specified as an absolute quantity or as a percentage if followed by %. It is converted to a fraction of the nominal value. The magnitude of the fractional value must be ≤ 1.

Example

```
.MODEL
```

Note

1. Both lot (`LOT`) and device (`DEV`) tolerances can be specified separately. Lot and device tolerances combine for the purposes of determining the total tolerance of a model parameter. The sum of device and lot tolerances must be less than the nominal model parameter value (i.e. is less than 100%).

2. A total of 20 random number are generated internally. Of these 10 are for lot tolerancing and 10 are for device tolerancing. Parameters with the same lot tolerance index (*LotToleranceIndex*) use the same random number to generate statistical variations and so are fully correlated. Similarly parameters with the same device tolerance index (*DeviceToleranceIndex*) use the same random number. However the random number for a lot and that for a device are always uncorrelated. If $X(T)$ is the value of a model parameter at temperature T, $R_{\rm LOT}(\mathit{LotToleranceIndex})$ is random number for lot variations and $R_{\rm DEV}(\mathit{DeviceToleranceIndex})$ is random number for device variations, $X_{\rm LOT}$ is the fractional lot tolerance $X_{\rm DEV}$ is the fractional device tolerance, the value used in a Monte Carlo simulation is

$$X'(T) = X(T)\,(1 + R_{\rm LOT}(\mathit{LotTrackingIndex})X_{\rm LOT} + R_{\rm DEV}(\mathit{DeviceTrackingIndex})X_{\rm DEV}) \qquad (6.2)$$

.NODESET Node Voltage Initialization

Form

`.NODESET V(`*NodeName*`)=`*Voltage* [`V(`*NodeName*`)=`*Voltage* ...]

V is the keyword specifying a node voltage

NodeName is the name of a node. Note that in SPICE2G6 and SPICE3 *NodeName* must be an integer.

Example

```
.NODESET V(11)=4.9 V(2)=2.5
```

Note

1. This statement can be used if convergence problems are encountered in a `DC` analysis. For most circuits SPICE will be able to determine the `DC` voltage or initial transient solution.
2. The statement can be used with astable circuits such as multivibrators and flip flops to ensure that these circuits are initialized in a particular state.

.NOISE Small-Signal Noise Analysis

In noise analysis the noise generated by active devices and resistors is evaluated.

Form

`.NOISE` *OutputVoltageSpecification InputSourceName OutputInterval*

SPICE3*Form*

`.NOISE` *OutputVoltageSpecification InputSourceName* `DEC`
`+` *FrequenciesPerDecade FStart FStop* [*OutputInterval*]

`.NOISE` *OutputVoltageSpecification InputSourceName* `OCT`
`+` *FrequenciesPerOctave FStart FStop* [*OutputInterval*]

`.NOISE` *OutputVoltageSpecification InputSourceName* `LIN`
`+` *NumberPoints FStart FStop OutputInterval*

OutputVoltageSpecification is a specification of an output voltage which is to be the output of the noise analysis. It acts as a summing point for the noise contributions of the individual noise current generators. The noise voltage appearing at the output for each noise generator is summed in the RMS sense. Any voltage specification may be used including the voltage at a node, e.g. using V(5), or the voltage between nodes, e.g. using V(5,3). The noise is reported in units of $V/\sqrt{Hz}$.

InputSourceName is the name of the independent voltage (`V`) or current (`I`) source that is to be the input reference source to which equivalent input noise is referred. The input source does not produce noise itself. If the input source is an independent voltage source then the equivalent input noise is reported in units of $V/\sqrt{Hz}$. If the input source is an independent current source then the equivalent input noise is reported in units of $A/\sqrt{Hz}$.

OutputInterval is the optional output reporting interval at which the values of the noise current generators internal to the elements of the circuit are reported. The report is produced every *OutputInterval* th frequency. If zero or omitted no detailed output is produced.

`DEC` is the decade sweep keyword specifying that the noise analysis is to be evaluated at a number of frequency points. The frequency is swept logarithmically by decades.

FrequenciesPerDecade specifies the number of frequencies per decade.

`OCT` is the octave sweep keyword specifying that the noise analysis is to be evaluated at a number of frequency points. The frequency is swept logarithmically by octaves.

FrequenciesPerOctave specifies the number of frequencies per octave.

`LIN` is the linear frequency sweep keyword. At each frequency a noise analysis is performed.

NumberPoints specifies the total number of frequencies in a linear sweep.

FStart is the starting frequency of the sweep.

FStop is the stopping frequency of the sweep.

Example

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

Example

```
.NOISE V(5) VIN
.NOISE V(5,3) I1
```

Note

1. In noise analysis the noise generated by active devices and resistors is evaluated as a noise spectral density. The densities are integrated over the the range of frequencies to obtain a gross noise measure for the circuit (for the specified frequency range). The finer the frequency spacing the more accurate will be the noise analysis. The noise contributions of individual noise generators are summed at the node or branch specified by *OutputVoltageSpecification*. The noise at this output port is reported as well as the equivalent input noise (the output noise referred to the input) at the input source identified by *InputSourceName*.

2. Two types of output are produced by noise analysis:

   (a) noise spectral density versus frequency, and

   (b) total integrated noise over the specified frequency range.

3. In SPICE3 the `AC` frequency sweep for noise analysis must be specified in the `.NOISE` statement. In PSPICE the frequency sweep specified in the `.AC statement` is used.

4. The noise table is produced while analysis is being performed. Noise reporting is produced using the `.PRINT` and `.PLOT` statements.

5. Noise is generated by resistors and by semiconductor devices. Resistors generate thermal noise while the noise model of semiconductor devices includes thermal noise, shot noise and flicker noise. Noise models of individual elements are discussed in the element catalog beginning on page 115.

Two-Port Noise and Gain Calculations

This is supported in a few versions of Spice.

For all circuits the output `ONOISE` and effective input noise `INOISE` are calculated. Also the voltage gain `GAIN` is calculated as the output voltage devided by the voltage across the source. Extended gain and noise parameters are available if the circuit is defined as a two port. The two-port parameters that are calculated are defined in terms of the signal and noise powers shown in Fig. 6.1 The actual noise and signal powers that are delivered to the circuit are P_i and N_i. Also the actual noise and signal powers that are delivered to the load resistance are P_o an N_o. The available noise and signal powers are the powers that would be delivered to the circuit with ideal lossless matching networks. That is, when the input and output impedances equal the source (R_S) and load (R_L) resistances — so that $P_i = P_{Ai}$ and $P_o = P_{Ao}$; and $N_I = N_A$ and $N_O = N_A$.

Two-port noise analysis yields the following quantities which can be specified in the `.PRINT` and `.PLOT` statements:

`ONOISE`	-	V_{NO}	RMS output noise voltage in $\mathrm{V}/\sqrt{\mathrm{Hz}}$
`INOISE`	-	V_{NO}	RMS equivalent input noise voltage in $\mathrm{V}/\sqrt{\mathrm{Hz}}$
`GAIN`	-	G	voltage gain
`GT`	-	G_T	transducer gain
`NF`	-	NF'	spot noise factor
`SNR`	-	SNR_i	output voltage signal-to-noise ratio
`TNOISE`	-	T_{NOISE}	output noise temperature in celsius.

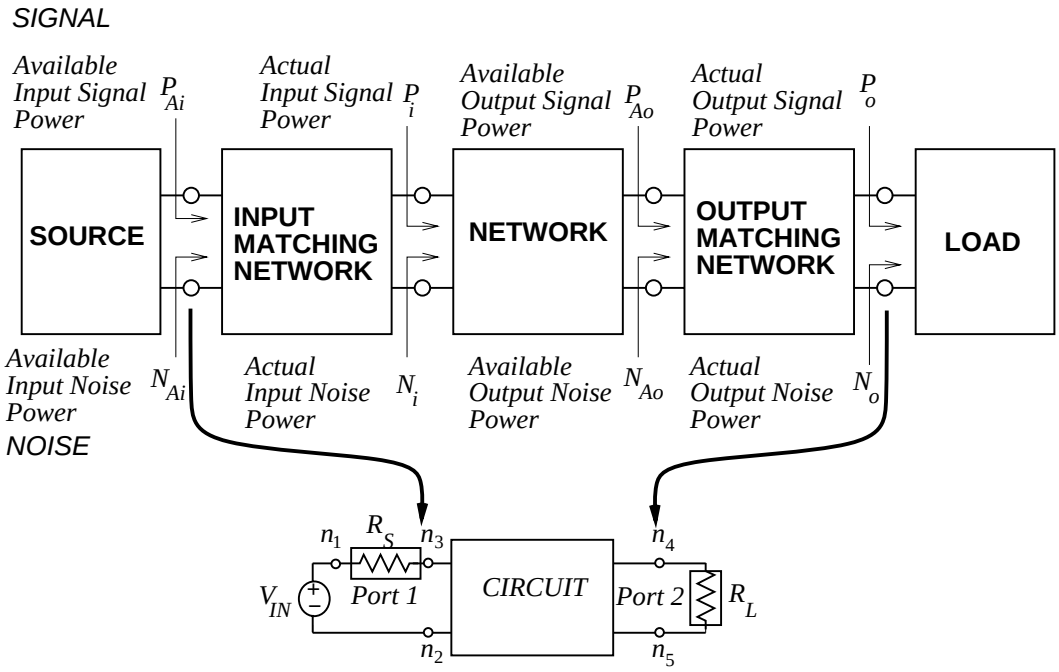


Figure 6.1: Signal and noise definitions for a two-port.

All of the quantities can be output in dB with the exception of $T_{\rm NOISE}$. When `DB(NF)` is used the spot noise figure is obtained.

The transducer gain, G_T, for the two port is defined as

$$G_T = P_o/P_{Ai} \tag{6.3}$$

The most common measure of the noise performance of a two port is the noise factor or noise figure. The spot noise factor NF' of a linear two-port network is defined as the ratio of the noise power delivered by the network to the load impedance to the fraction of the noise power due to the input termination alone. This noise is calculated with the input termination, R_S, at the standard temperature $T_0 = 290$ K. Note that this differs from the default analysis temperature of spice which is 300 K or 16.85 c. The noise power delivered to the output is the total noise power indicated by `ONOISE` less the noise power contributed to the output by R_L since it is not part of the two-port. These subtractions must be done using squared voltage quantities because the noises are uncorrelated. The noise power at the output due to R_S is the voltage gain squared multiplied by the square of the noise voltage in series with R_S. The noise factor calculated by SomeVersionsOfSpice is the spot noise factor as the noise powers are not averaged over frequency. The spot noise factor is

$$NF' = \frac{{}_0V_{NO}^2 - V_{NO,RL}^2}{{}_0V_{NO,RS}^2} \tag{6.4}$$

Here the leading zero subscript indicates that the noise is calculated with R_S at T_0. ${}_0V_{NO}$ is the output noise voltage with R_S at T_0, $V_{NO,RL}$ is the component due to the noise generated by R_L and ${}_0V_{NO,RS}$ is the component due to the noise generated by R_S at T_0.

The noise temperature in Kelvin is

$$T_{NOISE} = T_K(NF' - 1) \tag{6.5}$$

where T_K is the analysis temperature in Kelvin.

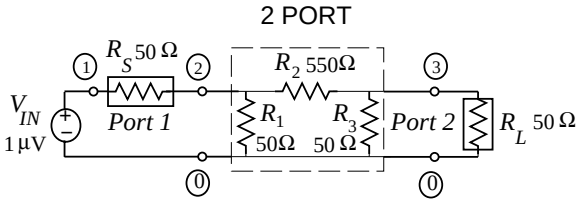


Figure 6.2: Circuit used as an example for specifying noise analysis.

The output voltage signal-to-noise ratio is the ratio of the signal voltage to the noise voltage:

$$\mathrm{SNR}_o = \frac{V_O''}{V_{NO}''\sqrt{2}} \tag{6.6}$$

where the $\sqrt{2}$ is required since noise voltages are specified in RMS terms but the signal voltages are specified as a peak voltage. V_O is the signal voltage at the output taking into account the signal-to-noise ratio, SNR_i, at the input of the circuit. SNR_i can either be specified on the voltage source line or, if not, calculated usnig the thermal voltage of R_S.

Three parameters affect the results of the noise analysis. These are the source resistance R_S, the load resistance, R_L and the input signal to noise ratio, SNR_i. The values used for the parameters are as follows:

R_S = The resistance of port 1 (`ZL`, or if Port 1 is not defined,
= The `RS` resistance specified on the source line, or if not specified.
= 50 Ω.

R_L = The resistance of port 2 (`ZL`, or if Port 2 is not defined.
= The `RL` resistance specified on the source line, or if not specified,
= 50 Ω.

SNR_i = The `SNR` specified on the source line, or if not specified.
= it is calculated as the signal voltage specified on the input line devided by the thermal noise voltage of R_S with appropriate correction for the difference between RMS and peak quantities.

Example of Two-Port Noise and Gain Analysis

The netlist for performing a two-port noise and gain analysis of the circuit in Fig. 6.2 is as follows.

```
Gain and noise analysis of resistive attenuator
vin 1 0 AC 1u RS=50 SNR=100
RS 1 2 50
R1 2 0 55
R2 2 3 500
R3 3 0 55
RL 3 0 50
*The following sets the analysis temperature to the standard temperature
.TEMP 16.85
.AC DEC 1 1MEG 2G
.NOISE V(3,0) VIN 1
.PRINT NOISE nf db(nf) gt db(gt) gain snr inoise onoise
.END
```

The example below performs the same analysis using ports and also prints the scattering parameters of the circuit.

```
Gain and noise analysis of resistive attenuator using ports.
vin 1 0 AC 1u SNR=100
PIN PNR=1 ZL=50
R1 2 0 55
R2 2 3 500
R3 3 0 55
POUT PNR=2 ZL=50
*The following sets the analysis temperature to the standard temperature
.TEMP 16.85
.AC DEC 1 1MEG 2G
.NOISE V(3) VIN 1
.PRINT NOISE nf db(nf) gt db(gt) gain snr inoise onoise
.PRINT AS S(1,1) S(1,2), S(2,1), S(2,2)
.END
```

.OP Operating Point Analysis

In the operating point analysis a DC analysis is performed to determine the DC voltages and currents without performing any sweeps.

Form

```
.OP
```

Note

1. The operating point analysis is performed by default prior to AC small-signal (.AC) and transient (.TRAN) analyses to determine the operating point about which the circuit is linearized for AC analysis and as the initial starting point for transient analysis.
2. The (.OP) analysis is performed if no other analyses are specified.
3. The operating point analysis is performed with inductors shorted and capacitors opened.
4. Following DC analysis the nonlinear devices are linearized to determine their AC small-signal models.
5. A .TRAN analysis performs its own DC analysis to determine the initial conditions (or bias point) for transient analysis ignoring the bias point determined in a .OP analysis.

.OPTIONS Option Specification

The options specification provides the user control over the program and a way of setting defaults for certain elements and analyses.

Form

`.OPTIONS` [*Keyword*] ... [*Keyword=Value*] ...

Multiple keywords can be included in a single `.OPTIONS` statement and in any order.

Keywords:

`ACCT` Sets reporting of accounting and statistics. This option is a flag and does not have a value.
(Default: not set)

`ABSTOL=`*Value* Sets the absolute current error tolerance.
(Default: 1 pA (10^{-12}))

`BYPASS` The bypass option.
SPICE3 only.

`CHGTOL=`*Value* Resets the charge tolerance of the program.
(Units: C; Default: 10 fC 10^{-14})

`CPTIME=`*Value* Sets the maximum CPU-time.
(Units: s; Default: ∞)
SPICE2G6 only.

`DEFAD=`*Value* Sets the default value of the MOS drain diffusion area (`AD`) used in the M element (see page 189).
(Units: m^2 Default: 0)

`DEFAS=`*Value* Sets the default value of the MOS source diffusion area (`AS`) used in the M element (see page 189).
(Units: m^2 Default: 0)

`DEFL=`*Value* Sets the value of the MOS channel length (`L`) used in the M element (see page 189).
(Units: m; Default: 100μm (1E-4))

`DEFW=`*Value* Resets the value for MOS channel width (`W`) used in the M element (see page 189).
(Units: m; Default: 100 μm (1E-4))

`EXPAND` Reports in output logfile the devices and nodes created in subcircuit expansions. This option is a flag and does not have a value.
(Default: no expansion)
PSPICE only.

`GMIN=`*Value* Resets the value of the minimum conductance G_{MIN}. The usage of G_{MIN} is controlled by the code implementing individual elements. Generally is the minimum conductance between nodes. It is used to aid convergence.
(Units: S; Default: 10^{-12} (1E-12))

ITL1=*IntegerValue* Sets the limit on the number of DC iterations.
(Default: 100)

ITL2=*IntegerValue* Sets the DC transfer curve iteration limit.
(Default: 50)

ITL3=*IntegerValue* Sets the minimum number of iterations used in transient analysis.
(Default: 4)
SPICE2G6 only.

ITL4=*IntegerValue* Sets the maximum of transient iterations at each time point.
(Default: 10)
SPICE2G6 only.

ITL5=*IntegerValue* Sets the transient analysis total iteration limit. If ITL5=0 this test is omitted. Used in transient analysis (.TRAN).
PSPICE only.
(Default: 5000)

LIBRARY Reports in output logfile the statements and devices extracted from a library file This option is a flag and does not have a value.
(Default: no report)
PSPICE only.

LIMPTS= *Value* Sets the maximum number of points that can be printed or plotted in a DC , AC or transient analysis.
(Default: 201)
SPICE2G6 only.

LIMTIM= *Value* Sets the maximum CPU time for generating plots. Used only if the program was terminated because the time specified by the CPTIME option was exceeded.
(Units: s; Default: 10)
SPICE2G6 only.

LIST Sets summary reporting of circuit elements in input NETLIST.
(Default: not set)

LVLCOD=*IntegerValue* Sets an internal option of the program when running on CDC computers. If LVLCOD=1 machine code for the matrix solution is generated.
(Default: 2)
SPICE2G6 only.

LVLTIM=*IntegerValue* If LVLTIM=1 then iteration time step control is used. If LVLTIM=2 then the time step indicated by the truncation error is used. If METHOD=GEAR and MAXORD¿2 then LVLTIM is set to 2 by SPICE.
(Default: 2)
SPICE2G6 only.

MAXORD=*IntegerValue* Sets the maximum order of the integration method if METHOD=GEAR
(Default: 2; 2≤MAXORD≤6)
SPICE2G6 only.

`METHOD`=*String* Sets the numerical integration method to be used. If `METHOD`=`GEAR` then Gear's method is used. If `METHOD`=`TRAPEZOIDAL` then the trapezoidal method is used.
(Default: `TRAPEZOIDAL`)
SPICE2G6 only.

`NODE` Sets reporting of the node table.
(Default: not set)

`NOECHO` Suppresses listing of input file in output log file.
(Default: input lines listed.)
PSPICE only.

`[NOFREQ ]` Number of frequency points to be used in simulation of distributed circuit. Must be a power of 2.
(Default: 1024)

`NOMOD` Un-sets reporting of model parameters.
(Default: set)

`NOPAGE` Un-sets page breaks in the output log file. Useful if the log file is to printed by a program which automatically paginates the output. This option is a flag and does not have a value.
(Default: set)

`NUMDGT`=*IntegerValue* Sets the number of significant digits used in printing values in the output log file.
(Default: 4; $0 < \mathit{IntegerValue} \leq 8$)
SPICE2G6 and PSPICE only.

`OPTS` Sets reporting of the option values.
(Default: not set)

`PIVREL`=*Value* Sets the minimum acceptable pivot value used in partial pivoting in the solution of the network equations (such as solving for the nodal voltages $\mathbf{v}_n$ in (4.1) on page 34). `PIVREL` is the minimum acceptable ratio of an acceptable pivot value to the largest column entry.
(Default: 0.001)

`PIVTOL`=*Value* Sets the minimum value of a matrix element for it to be used as a pivot.
(Default: 10^{-13})

`RELTOL`=*Value* Sets the relative error tolerance of voltages and currents.
(Default: 0.001)

`TNOM`=*Value* Sets the nominal temperature. This is assumed to be the temperature at which the model parameters were measured. In some cases it is overwritten by a temperature parameter in the `.MODEL` statement.
(Units: °C; Default: 27°C (300K))

`TRTOL`=*Value* Sets the factor by which the approximated truncation error evaluated in transient analysis is scaled.
(Default: 7)

`VNTOL`=*Value* Sets the absolute voltage error tolerance.
(Units: V; Default: 1μV (1E-6))

.PARAM — PSpice Only

Parameter Definition

This statement defines parameters that can be used in subsequent statements and element lines.

PSpice*Form*

```
.PARAM [ParameterName = NumericValue ... ] [ParameterName = { Expression } ... ]
```

ParameterName Name of a parameter with first character being alphabetic (a-z). If this is the name of a previously defined parameter at the same level of subcircuit expansion then the parameter value is changed. If the `.PARAM` statement is is in the top level circuit then the parameter value is global. If the `.PARAM` statement is is in a subcircuit then the parameter value is local and can be used at the current subcircuit expansion level or lower in the subcircuit expansion hierarchy.

NumericValue is a numeric value which can be an integer or floating point number followed by optional scale factor and/or unit. (e.g. 1.E-9, 1N, 1NV and 1.E-9V are equivalent.)

Expression is a standard expression as described in the Algebraic Expressions section on page **??**. Note that the expression must be enclosed in braces ({ ... }).

Example

```
.PARAM VDD = 10V, VSS = 0 VREF = 2.5
.PARAM VREF =  VDD/2 .PARAM LENGTH = 1.10*L
```

Note

1. Predefined parameters are supported and the user must avoid using these. Predefined parameters:

Name	Value	Description
TEMP	not supported Reserved for future expansion	Analysis temperature.
VT	not supported Reserved for future expansion	Thermal voltage.

2. In PSpice , in most places where a numeric value is required an expression (within braces { ... }) can be used instead. An expression can contain any support mathematical operation, constant numeric values or expressions. Exceptions are

   - Polynomial coefficients.
   - The values of the transmission line device parameters `NL` and `F`.
   - The values of the piece-wise linear characteristic in the `PWL` form of the independent voltage (`V`) and current (`I`) sources.

   and

   - The values of the resistor device parameter `TC`.
   - As node numbers.
   - Values of most statements (such as .TEMP, .AC, .TRAN etc.)

   Specifically included are

   - The values of all other device parameters.
   - The values in `.IC` and `.NODESET` statements.
   - The values in `.SUBCKT` statements.

   and

   - The values of all model parameters. `F`.

3. Hierarchical usage of `.PARAM` statements in subcircuits is supported. The parameters defined in a `.PARAM` statement are available in the subcircuit in which they are defined or in lower nested subcircuits. Thus parameters defined in a subcircuit are not available higher in the hierarchy. The same concept applies to values of a parameter changed in a subcircuit. Value changes are local and are available in the current subcircuit and lower nested subcircuits.

4. Libraries are searched for parameters not defined in the circuit NETLIST or in included files. A `.PARAM` statement does not have to be within a subcircuit in a library.

.PLOT Plot Specification

The plot specification controls the information that is plotted as the result of various analyses.

Form

`.PLOT TRAN` *OutputSpecification* [*PlotLimits*]
`+` [*OutputSpecification* [*PlotLimits*] ...]

`.PLOT AC` *OutputSpecification* [*PlotLimits*]
`+` [*OutputSpecification* [*PlotLimits*] ...]

`.PLOT DC` *OutputSpecification* [*PlotLimits*]
`+` [*OutputSpecification* [*PlotLimits*] ...]

`.PLOT NOISE` *NoiseOutputSpecification* [`(`*DistortionReportType*`)`] [*PlotLimits*]
`+` [*NoiseOutputSpecification* [`(`*DistortionReportType*`)`] [*PlotLimits*]

`.PLOT DISTO` *DistortionOutputSpecification* [`(`*DistortionReportType*`)`] [*PlotLimits*
]
`+` [*DistortionOutputSpecification* [`(`*DistortionReportType*`)`] [*PlotLimits*] ...]
[`(`*DistortionReportType*`)`] [*PlotLimits*]

PSpice*Form*

`.PLOT TRAN` *OutputSpecification* [*PlotLimits*]
`+` [*OutputSpecification* [*PlotLimits*] ...]

`.PLOT AC` *OutputSpecification* [*PlotLimits*]
`+` [*OutputSpecification* [*PlotLimits*] ...]

`.PLOT DC` *OutputSpecification* [*PlotLimits*]
`+` [*OutputSpecification* [*PlotLimits*] ...]

`.PLOT NOISE` *NoiseOutputSpecification* [`(`*DistortionReportType*`)`] [*PlotLimits*]
`+` [*NoiseOutputSpecification* [`(`*DistortionReportType*`)`] [*PlotLimits*]

`TRAN` is the keyword specifying that this `.PLOT` statement controls the reporting of results of a transient analysis initiated by the `.TRAN` statement.

`AC` is the keyword specifying that this `.PLOT` statement controls the reporting of results of a small-signal `AC` analysis initiated by the `.AC` statement.

`DC` is the keyword specifying that this `.PLOT` statement controls the reporting of results of a `DC` analysis initiated by the `.DC` statement.

`NOISE` is the keyword specifying that this `.PLOT` statement controls the reporting of results of a noise analysis initiated by the `.NOISE` statement.

`DISTO` is the keyword specifying that this `.PLOT` statement controls the reporting of results of a small-signal `AC` distortion analysis initiated by the `.DISTO` statement.

OutputSpecification specifies the voltage or current to be plotted against the sweep variable. The sweep variable is dependent on the type of analysis.
Many forms of `OutputSpecification` are supported by PSpice . Below is a description of the basic forms that are supported both by Spice2g6 and PSpice . A comprehensive description of `OutputSpecification` supported by PSpice is given in the section on output specification on page 94.

Voltages may be specified as an absolute voltage at a node: `V(`*NodeName*`)` or the voltage at one node with respect to that at another node, e.g. `V(Node1Name,Node2Name)`.
For the reporting of the results of an `AC` analysis the following outputs can be specified by replacing the `V` as follows:

`VR`	-	real part
`VI`	-	imaginary part
`VM`	-	magnitude
`VP`	-	phase
`VDB`	-	$10\log(10\,magnitude)$

In `AC` analysis the default is `VM` for magnitude.

Currents are specified by referencing the name of the voltage source through which the current is measured, e.g. `I(V`*VoltageSourceName*`)`.
For the reporting of the results of an `AC` analysis the following outputs can be specified by replacing the `I` as follows:

`IR`	-	real part
`II`	-	imaginary part
`IM`	-	magnitude
`IP`	-	phase
`IDB`	-	$10\log(10\,magnitude)$

In `AC` analysis the default is `IM` for magnitude.

PlotLimits are optional and can be placed after any output specification. *PlotLimits* has the form `(`*LowerLimit*`,`*UpperLimit*`)` . All quantities will be plotted using the same *PlotLimits*. The default is to automatically scale the plot and perhaps use different scales for each of the quantities to be plotted.

NoiseOutputSpecification specifies the noise measure to be reported. The two options are `ONOISE` which reports the output noise and `INOISE` which reports the equivalent input noise. See the `.NOISE` statement on page 77 for a detailed explanation.

It must be one of the following:

`ONOISE`	-	magnitude of the output noise
`DB(ONOISE)`	-	output noise in dB
`INOISE`	-	magnitude of the equivalent input noise
`DB(INOISE)`	-	equivalent input noise in dB
`GAIN`	-	voltage gain
`DB(GAIN)`	-	voltage gain in dB (= 20 log(`GAIN`)
`GT`	-	transducer gain
`DB(GT)`	-	transducer gain in dB (= 10 log(`GT`)
`NF`	-	spot noise factor
`DB(NF)`	-	spot noise figure (= 10 log(`NF`)
`SNR`	-	output signal-to-noise ratio
`DB(SNR)`	-	output signal-to-noise ratio in dB (= 20 log(`SNR`)
`TNOISE`	-	output noise temperature.

SParameterOutputSpecification specifies the S-parameter output variables that are to be printed. Each variable must have one of the following forms:

`S(i,j)`	-	Magnitude of S_{ij}
`SR(i,j)`	-	Real part of S_{ij}
`SI(i,j)`	-	Imaginary part of S_{ij}
`SP(i,j)`	-	Phase of S_{ij} in degrees
`SDB(i,j)`	-	Magnitude of S_{ij} in dB (= 20 log(`S(i,j)`))
`SG(i,j)`	-	Group delay of S_{ij}

The port numbers are i, j which are specified using the `PNR` keywor when the port ('P') element is specified.

DistortionOutputSpecification specifies the distortion component to be reported in a tabular format of up to 8 columns plus an initial column with the sweep variable. The *DistortionOutputSpecification* is one of the following:

`HD2`	-	the second harmonic distortion
`HD3`	-	the second harmonic distortion
`SIM2`	-	the sum frequency intermodulation component
`DIM2`	-	the difference frequency intermodulation component
`DIM3`	-	the third order intermodulation component

See the `.DISTO` statement on page 58 for a description of these distortion components.

DistortionReportType specifies the format for reporting the distortion components. It must be one of the following:

R	-	real part
I	-	imaginary part
M	-	magnitude
P	-	phase
DB	-	$10\log(10\,magnitude)$

The default is `M` for magnitude.

Example

```
.PLOT TRAN V(10) V(5,3) I(VIN)
.PLOT AC VM(10) VR(5,3) IP(VLOAD)
.PLOT DC V(10) V(5,3) I(VIN)
.PLOT NOISE ONOISE INOISE DB(ONOISE) DB(INOISE)
.PLOT NOISE GAIN DB(GT) DB(NF) SNR TNOISE
.PLOT AS SDB(1,1) SP(1,1) SDB(1,2) SP(1,2)
.PLOT DISTO HD2 HD3 SIM2(DB)
```

Note

1. There can be any number of `.PLOT` statements.
2. All of the output quantities specified on a single `.PLOT` statement will be plotted on the same graph using ASCII characters. An overlap will be indicated by the letter `X`. The plot produced by the `.PLOT` statement is a line printer plot. While plotting is primitive it can be plotted on any printer and is incorporated in the output log file.
3. The plot output of the results of an `AC` analysis always have a logarithmic vertical scale.
4. The current through any element can be found by inserting independent voltage sources in series with the elements. This is generally what is required in SPICE2G6and SPICE3. However PSPICE supports direct specification of the voltage and currents of most elements. See the section on page 94.
5. More elaborate plotting is available with SPICE3 using the NUTMEG plotting program described on the NUTMEG chapter beginning on page **??**; and with PSPICE using the `.PROBE` statement described on page 98.

.PRINT Print Specification

The print specification controls the information that is reported as the result of various analyses.

Form

```
.PRINT TRAN  OutputSpecification  [OutputSpecification ... ]

.PRINT AC  OutputSpecification  [OutputSpecification ... ]

.PRINT DC  OutputSpecification  [OutputSpecification ... ]

.PRINT DISTO DistortionOutputSpecification ( DistortionReportType )
+ [DistortionOutputSpecification ( DistortionReportType ) ... ]
```

`TRAN` is the keyword specifying that this `.PRINT` statement controls the reporting of results of a transient analysis initiated by the `.TRAN` statement.

`AC` is the keyword specifying that this `.PRINT` statement controls the reporting of results of a small-signal `AC` analysis initiated by the `.AC` statement.

`DC` is the keyword specifying that this `.PRINT` statement controls the reporting of results of a `DC` analysis initiated by the `.DC` statement.

`NOISE` is the keyword specifying that this `.PRINT` statement controls the reporting of results of a noise analysis initiated by the `.NOISE` statement.

`DISTO` is the keyword specifying that this `.PRINT` statement controls the reporting of results of a small-signal `AC` distortion analysis initiated by the `.DISTO` statement.

OutputSpecification specifies the voltage or current to be reported in a tabular format of up to 8 columns plus an initial column with the sweep variable.
Many forms of `OutputSpecification` are supported by PSpice . Below is a description of the basic forms that are supported both by Spice2g6 and PSpice . A comprehensive description of `OutputSpecification` supported by PSpice is given in the section on output specification on page 94.

Voltages may be specified as an absolute voltage at a node: `V(`*NodeName*`)` or the voltage at one node with respect to that at another node, e.g. `V(Node1Name,Node2Name)`.
For the reporting of the results of an `AC` analysis the following outputs can be specified by replacing the `V` as follows:

`VR`	-	real part
`VI`	-	imaginary part
`VM`	-	magnitude
`VP`	-	phase
`VDB`	-	$10\log(10\,magnitude)$

In `AC` analysis the default is `VM` for magnitude.

Currents are specified by referencing the name of the voltage source through which the current is measured, e.g. `I(V`*VoltageSourceName*`)`.

For the reporting of the results of an `AC` analysis the following outputs can be specified by replacing the `I` as follows:

`IR`	-	real part
`II`	-	imaginary part
`IM`	-	magnitude
`IP`	-	phase
`IDB`	-	$10\log(10\,magnitude)$

In `AC` analysis the default is `IM` for magnitude.

NoiseOutputSpecification specifies the noise measure to be reported. The two options are `ONOISE` which reports the output noise and `INOISE` which reports the equivalent input noise. See the `.NOISE` statement on page 77 for a detailed explanation.
It must be one of the following:

`ONOISE`	-	RMS output noise voltage
`DB(ONOISE)`	-	output noise voltage in dB (= 20 log(`ONOISE`)
`INOISE`	-	RMS equivalent input noise voltage
`DB(INOISE)`	-	equivalent input noise voltage in dB (= 20 log(`INOISE`)
`GAIN`	-	voltage gain
`DB(GAIN)`	-	voltage gain in dB (= 20 log(`GAIN`)
`GT`	-	transducer gain
`DB(GT)`	-	transducer gain in dB (= 10 log(`GT`)
`NF`	-	spot noise factor
`DB(NF)`	-	spot noise figure (= 10 log(`NF`)
`SNR`	-	output signal-to-noise ratio
`DB(SNR)`	-	output signal-to-noise ratio in dB (= 20 log(`SNR`)
`TNOISE`	-	output noise temperature.

SParameterOutputSpecification specifies the S-parameter output variables that are to be printed. Each variable must have one of the following forms:

`S(i,j)`	-	Magnitude of S_{ij}
`SR(i,j)`	-	Real part of S_{ij}
`SI(i,j)`	-	Imaginary part of S_{ij}
`SP(i,j)`	-	Phase of S_{ij} in degrees
`SDB(i,j)`	-	Magnitude of S_{ij} in dB (= 20 log(`S(i,j)`))
`SG(i,j)`	-	Group delay of S_{ij}

The port numbers are i, j which are specified using the `PNR` keyword when the port element is specified.

DistortionOutputSpecification specifies the distortion component to be reported in a tabular format of up to 8 columns plus an initial column with the sweep variable. The *DistortionOutputSpecification* is one of the following:

`HD2`	-	the second harmonic distortion
`HD3`	-	the second harmonic distortion
`SIM2`	-	the sum frequency intermodulation component
`DIM2`	-	the difference frequency intermodulation component
`DIM3`	-	the third order intermodulation component

See the .DISTO statement on page 58 for a description of these distortion components.

DistortionReportType specifies the format for reporting the distortion components. It must be one of the following:

R	-	real part
I	-	imaginary part
M	-	magnitude
P	-	phase
DB	-	$10\log(10\,magnitude)$

The default is M for magnitude.

Example

```
.PRINT TRAN V(10) V(5,3) I(VIN)
.PRINT AC VM(10) VR(5,3) IP(VLOAD)
.PRINT DC V(10) V(5,3) I(VIN)
.PRINT NOISE ONOISE INOISE DB(ONOISE) DB(INOISE)
.PRINT NOISE GAIN DB(GT) DB(NF) SNR TNOISE
.PRINT AS SDB(1,1) SP(1,1) SDB(1,2) SP(1,2)
.PRINT DISTO HD2 HD3 SIM2(DB)
```

Note

1. There can be any number of .PRINT statements.
2. The number of significant digits of the results reported is NUMDGT which is set in a .OPTIONS statement (see page 85).
3. The current through any element can be found by inserting independent voltage sources in series with the elements. This is generally what is required in SPICE2G6and SPICE3. However PSPICE supports direct specification of the voltage and currents of most elements. See the section on page 94.

Output Specification for PSPICE

PSPICE supports a relatively large variety of output specifications compared to that available with SPICE2G6 and SPICE3. The output specifications described in the following can be used .PRINT and .PLOT statements. The various forms of output specifications enable the current and voltages of virtually all devices to be examined¿

DC and TRAN Reporting

The output specifications available for the DC sweep and transient analyses are

I(*DeviceName*) Current through a two terminal device (such as a resistor R element) or the output of a controlled voltage or current source. e.g. I(R22) is the current flowing through resistor R22 from node N_1 to N_2 of R22.

I TerminalName(*DeviceName*) Current flowing into terminal named *TerminalName* (such as B for gate) from the device named *DeviceName* (such as Q12). e.g. IB(Q12)

I PortName(*TransmissionLineName*) Current at port named *PortName* (either A or B) of the transmission line device named *TransmissionLineName*

V(*NodeName*) Voltage at a node of name *NodeName*.

V(n_1, n_2) Voltage at node n_1 with respect to the voltage at node n_2.

V(*DeviceName*) Voltage across a two terminal device (such as a resistor R element) or at the output of a controlled voltage or current source.

V TerminalName(*DeviceName*) Voltage at terminal named *TerminalName* (such as G for gate) of the device named *DeviceName* (such as M12). e.g. VG(M12)

V TerminalName1 TerminalName2(*DeviceName*) Voltage at terminal named *TerminalName1* (such as G for gate) th respect to the terminal name *TerminalName2* (such as S for source) of the device named *DeviceName* (such as M12). e.g. VGS(M12)

V PortName(*TransmissionLineName*) Voltage at port named *PortName* (either A or B) of the transmission line device named *TransmissionLineName* (such as T5). e.g. VA(M5)

Two Terminal Device Types Supported for DCand Transient Analysis Reporting

The single character identifier for the following elements as well as the rest of the device name can be used as the *DeviceName* in the I(*DeviceName*) and I(*DeviceName*) output specifications.

Element Type	Description
C	capacitor
D	diode
E	voltage-controlled voltage source
F	current-controlled current source
G	voltage-controlled current source
H	current-controlled voltage source
I	independent current source
L	inductor
R	resistor
V	independent voltage source

Multi-Terminal Device Types Supported for DCand Transient Analysis Reporting

The single character identifier for the following elements as well as the rest of the device name can be used as the *DeviceName* in the I TerminalName(*DeviceName*), V TerminalName(*DeviceName*) and V TerminalName1 TerminalName2(*DeviceName*) output specifications.

Element Type	Description
B	GaAs MESFET Terminals: D — drain G — gate S — source
J	JFET Terminals: D — drain G — gate S — source
M	MOSFET Terminals: B — bulk or substrate D — drain G — gate S — source
Q	BJT Terminals C — collector B — base E — emitter S — source

AC Reporting

The output specifications available for reporting the results of an AC frequency sweep analysis includes all of the specification formats discussed above for DC and transient analysis together with a number of possible suffixes:

- DB - $10\log(10\,magnitude)$
- M - magnitude
- P - phase
- R - real part
- I - imaginary part
- G - group delay = $\partial\phi/\partial f$
  where ϕ is the phase of the quantity being reported and f is the analysis frequency.

In AC analysis the default suffix is M for magnitude.

Two-Terminal Device Types Supported for AC Reporting

The single character identifier for the following elements as well as the rest of the device name can be used as the *DeviceName* in the I(*DeviceName*) and I(*DeviceName*) output specifications.

Element Type	Description
C	capacitor
D	diode
I	independent current source
L	inductor
R	resistor
V	independent voltage source

Multi-Terminal Device Types Supported for DCand Transient Analysis Reporting

The single character identifier for the following elements as well as the rest of the device name can be used as the *DeviceName* in the `I`TerminalName`(`*DeviceName*`)`, `V`TerminalName`(`*DeviceName*`)` and `V`TerminalName1 TerminalName2`(`*DeviceName*`)` output specifications.

Element Type	Description
`B`	GaAs MESFET Terminals: D — drain G — gate S — source
`J`	JFET Terminals: D — drain G — gate S — source
`M`	MOSFET Terminals: B — bulk or substrate D — drain G — gate S — source
`Q`	BJT Terminals C — collector B — base E — emitter S — source

.PROBE — Data Output Specification

There is a big problem here — Probe is a trademark of Microsim corporation

The `.PROBE` statement saves the node voltages and device currents in a file for subsequent probing.

PSPICE*Form*

`.PROBE` [`/CSDF`] [*OutputSpecification* ...]

`/CSDF` is a keyword resulting in the output probing file being written in ASCII format. By default the probing file is output in the more efficient binary format. However, only the ASCII formated is fully portable between computers and operating systems.

OutputSpecification specifies a node voltage or device current to be included in the probe data file. The output specifications supported are those supported for the `.PRINT` and `.PLOT` statements. A comprehensive description of the `OutputSpecification` supported is given in the section on output specification on page 94.
If an *OutputSpecification* is not given then all node voltages and device currents are stored in the probing file.

Example

```
.PROBE V(10) V(5,3) I(VIN)
.PROBE VM(10) VR(5,3) IP(VLOAD)
.PROBE/CSDF V(10) IG(VIN)
.PROBE
```

The first example will output data from `.DC`, `.AC` and `.TRAN` analyses.
The second example will output data only from a `.AC` analysis.
The third example will output the node voltage at node 10 for `.DC`, `.AC` and `.TRAN` analyses but output the group delay of the current in the independent voltage source `VIN` only from an `.AC` analysis. As well the data will be output in ASCII format.
The fourth example results in all node voltages and device currents being stored.

Note

1. The probing data is stored in the file `PROBE.DAT` .
2. The results of `DC`, `AC` and transient analyses are saved. An *OutputSpecification* which is unique to a particular type of analysis is ignored when the the results of analyses are being selected to storage. An example is `VI(10)` which selects the imaginary part of the voltage at node 10 and so only applies for storage of the results for an `AC` small-signal analysis.
3. The results stored in the probe file can subsequently be viewed using the PROBEprogram described in chapter **??**.

.PZ Pole-Zero Analysis

In pole zero analysis the poles and zeros of the small signal ACtransfer function of a two-port is evaluated.

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
```

Node1 is the positive input node.

Node2 is the negative input node.

Node3 is the positive output node.

Node4 is the negative output node.

CUR is the keyword to evaluate the transfer function (output voltage)/(input current)

VOL is the keyword to evaluate the transfer function (output voltage)/(input voltage)

POL is the keyword to evaluate the poles of the transfer function only.

ZERO is the keyword to evaluate the zeroes of the transfer function only.

PZ is the keyword to evaluate the poles and zeroes of the transfer function.

Example

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

Note

1. The pole-zero analysis works with resistors R, capacitors C, inductors L, linear controlled sources E,F,G and H; independent voltage and current sources V and I, bipolar junction transistors, Q; MOSFETs, M; JFETs J; and diodes D. In particular distributed devices such as Transmission lines are not supported as these do not have a pole-zero description.

2. In interactive mode, the command syntax is the same except that the first field is PZ instead of .PZ. To print the results, one should use the command "PRINT ALL".

3. The program first computes the DC operating point and then determines the linearized, small-signal models for all the nonlinear devices in the circuit. This circuit is then used to find the poles and zeros.

.SAVEBIAS — Save Bias Conditions

.SENS Sensitivity Analysis

The sensitivity analysis calculates the `DC` small-signal sensitivities of each output quantity with respect to every circuit parameter.

Form

`.SENS` *OutputSpecification* [*OutputSpecification* ...]

OutputSpecification is the specification of the small-signal output quantity. It has the same format as the *OutputSpecification* in a `.PRINT` statement (see page 92).

Example

```
.SENS V(10) V(10,2) I(VLOAD)
```

Note

1. The `.SENS` statement initiates a small-signal `AC` analysis.
2. The transfer function computed is the sensitivity (or partial derivative) of the `DC` value of the output quantity with respect to the each and every circuit parameter. For example, if a resistor is specified with value x and a capacitor is specified with value c then, for the above examples the following sensitivities are computed:

$$\frac{\partial \mathtt{V(10)}}{\partial x} \qquad \frac{\partial \mathtt{V(10,2)}}{\partial x} \qquad \frac{\partial \mathtt{VLOAD}}{\partial x} \qquad \frac{\partial \mathtt{V(10)}}{\partial c} \qquad \frac{\partial \mathtt{V(10,2)}}{\partial c} \qquad \frac{\partial \mathtt{VLOAD}}{\partial c}$$

.STEP Parameteric Analysis

.SUBCKT Subcircuit Statement

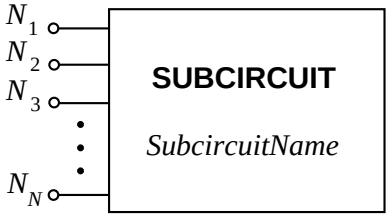


Figure 6.3: Subcircuit.

Form

`.SUBCKT` *SubcircuitName* N_1 [N_2 N_3 ... N_N]

PSPICE*Form*

`.SUBCKT` *SubcircuitName* N_1 [N_2 N_3 ... N_N]
`+` [`PARAMS:` [*Keyword* = { *Expression* } ...] [*Keyword* = *Value* ...]]

PSPICE*Form*

`.SUBCKT` *SubcircuitName* N_1 [N_2 N_3 ... N_N]
`+` [*PARAMS:* [*keyword* = { *Expression* } ...] [*Keyword* = *Value* ...]]

SubcircuitName is the name of the subcircuit.

N_1 is the first node of the subcircuit.

N_N is the Nth node of the subcircuit.

`PARAMS:` indicates that parameters are to be passed to the subcircuit.

Keyword: is a keyword which may be replaced by a value specified on a subcircuit call (`X`) element. (See page 257).

Value: is a numeric value.

Expression: is an algebraic expression which evaluates to a numeric value. (See section **??** on page **??** for allowable expressions).

Example

```
.SUBCKT MULTI 2 4 17 3 1
```

Note

1. The global ground node, node 0 (or in PSpice node `GND`) must not be one of the subcircuit nodes.

2. Subcircuits are incorporated by using the "`X`" element. The number of nodes of the "`X`" element must correspond to the number of nodes in the definition of the subcircuit (i.e. is in the `.SUBCKT` statement). See page 257 for a description of the `X` element.

3. The last line in a subcircuit definition is the .ENDS line (see page 63).

4. The only restriction on the statements within a subcircuit is that control lines such as `.AC`, `.DC` or `.OPTIONS` are not allowed. However, element lines, model statements and other subcircuit definitions and subcircuit calls are allowed.

5. Device models or subcircuit definitions included as part of a subcircuit definition are local only. Subcircuits and models that are to be "known" by all elements and subcircuits must be defined at the top level of the circuit hierarchy.

6. The nodes of elements in the subcircuit definition are local except for those that also appear on the `.SUBCKT` statement and for the ground (`0`) node. The local nodes are given the unique name
   SubcircuitName1:[*SubcircuitName2*: . . .]*LocalNodeName* .
   and local devices are given the unique name
   DeviceType:*SubcircuitName1*:[*SubcircuitName2* . . .]*DeviceName* .

.TEMP Temperature Specification

The `.TEMP` statement specifies the temperatures at which the circuit is to be simulated.

Form

`.TEMP` T_1 [T_2 ... T_N]

T_1 is the first temperature at which the circuit is to be simulated.

T_N is the Nth temperature at which the circuit is to be simulated.

Note

1. The circuit is first simulated at temperature T_1 and then resimulated at temperature T_2 and so on.
2. If the `.TEMP` statement is missing then T_1 is assumed to be $T_{\rm NOM}$ specified in a `.OPTIONS` statement.
3. Model parameters are specified at $T_{\rm NOM}$ and prior to the simulation at a new temperature temperature dependent device parameters are reevaluated.

.TEXT — Text Parameter Definition

.TF Transfer Function Specification

The transfer function specifies a small-signal `DC` analysis from which a small-signal transfer function and input and output resistances are computed..

Form

`.TF` *OutputSpecification InputSourceName*

OutputSpecification is the specification of the small-signal output quantity. It has the same format as the *OutputSpecification* in a `.PRINT` statement (see page 92).

InputSourceName specifies the the name of the small-signal input independent voltage (`V`) or current (`I`) source.

Example

```
.TF V(10) VINPUT
.TF V(10,2) ISOURCE
.TF I(VLOAD) ISOURCE
```

Note

1. The `.TF` statement initiates a small-signal `DC` analysis from which a small-signal transfer function and input and output resistances are computed.
2. The transfer function computed is the ratio of the `DC` value of the output quantity to the input quantity. In the above examples the following transfer functions are computed:

EXAMPLE	Transfer Function
.TF V(10) VINPUT	$\frac{\texttt{V(10)}}{\texttt{VINPUT}}$
.TF V(10,2) ISOURCE	$\frac{\texttt{V(10,2)}}{\texttt{ISOURCE}}$
.TF I(VLOAD) ISOURCE	$\frac{\texttt{I(VLOAD)}}{\texttt{ISOURCE}}$

TITLE — Title Line

The `TITLE` line must be the first line of the input file. The string on this line included as the banner in the output log file appearing at the top of each page.

.TRAN Transient Analysis

In transient analysis the current and voltages in a circuit are computed as a function of time.
General form:

Form

`.TRAN` *TSTEP TSTOP* [*TSTART* [*TMAX*]] [`UIC`]

TSTEP is the time increment for reporting transient simulation results.
(Units: s)

TSTOP is the final analysis time.
(Units: s)

TSTART is the start time for reporting the transient results. Transient analysis always begins at time 0. Before the time `TSTART` no results are recorded.
(Units: s; Optional; Default: 0)

TMAX is the maximum step size used in incrementing the time during transient analysis.
(Units: s; Optional; Default: the smaller of `TSTEP` and (`TSTOP-TSTART`)/50)

`UIC` is the optional keyword to use initial conditions specified on the element line, by a `.IC` statement or a `.NODESET` statement. Normally the operating point is determined (using a `DC` analysis) before a transient analysis is initiated. If the `UIC` keyword is present the initial `DC` analysis is omitted and instead the initial conditions specified by the `IC` parameter supported by certain elements, by the `.IC` statement, or by the `.NODESET` statement are used. In addition PSpice supports using an operating point solution that was previously saved using the `.SAVEBIAS` statement.

Example

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

Note

1. If the UIC keyword is not present a DC analysis is automatically performed prior to a transient (.TRAN) analysis to find the operating point of the circuit. All sources which are not time dependent (for example, power supplies) are set to their DC value. The transient time interval is specified on a .TRAN control line. The operating point solution is used as the initial conditions for a transient analysis. Individual operating point solutions are overridden by initial transient conditions specified on the .TRAN statement or by the initial conditions specified for specific elements by the IC keyword.

2. Normally the operating point is determined (using a DC analysis) before a transient analysis is initiated. If the UIC keyword is present the initial DC analysis is omitted and instead the initial conditions specified by the IC parameter supported by certain elements, by the .IC statement (see 66), or by the .NODESET statement (see 76) are used. In addition PSpice supports using an operating point solution that was previously saved using the .SAVEBIAS statement described on page 100.

3. If the UIC keyword is present the initial conditions specified in the .IC statement (described on page 66) are used to establish the initial conditions. Initial conditions specified for individual elements using the IC parameter on the element line will always have precedence over those specified in a .IC statement. No DC analysis is performed prior to a transient analysis. Thus it is important to establish the initial conditions at all nodes using the .IC statement or using the IC element parameter.

.WATCH Watch Analysis Statement

.WCASE Sensitivity and Worst Case Analysis

The sensitivity and worst case analysis is a statistical analysis of a circuit causing the circuit to analyzed many times with a random change of model parameters (parameters in a `.MODEL` statement).

Form

`.WCASE` *NumberOfRuns AnalysisType OutputSpecification OutputFunction* [`LIST`]
`+` [`OUTPUT(`*OutputSampleType* `)`] [`RANGE(`*LowValue*`,` *HighValue*`)`]

NumberOfRuns is the total number of runs to do. This number includes the initial nominal run.

AnalysisType is the type of analysis to be performed in Monte Carlo runs after the initial nominal run. All analyses specified in the `NETLIST` are performed in the nominal run. The *AnalysisType* must be one of the following:

to be filled in

`DC` is a keyword indicating that the `DC` analysis as specified by the `.DC` statement is repeated. The sweep variable used in analyzing the output *OutputSpecification* is the value of the independent voltage or current source specified in the `.DC` statement which is discussed on page 55.

`AC` is a keyword indicating that the `AC` small-signal analysis as specified by the `.AC` statement is repeated. The sweep variable used in analyzing the output *OutputSpecification* is frequency.

`TRAN` is a keyword indicating that the transient analysis as specified by the `.TRAN` statement is repeated. The sweep variable used in analyzing the output *OutputSpecification* is time.

OutputSpecification specifies the quantity to be reported as the result of the Monte Carlo Analysis. It has the same format as the *OutputSpecification* in a `.PRINT` statement (see page 92). The result is the value of the *OutputSpecification* with respect to a sweep for `DC` and `AC` analysis, and as a waveform for `TRAN analysis`.

OutputFunction indicates the function to be performed on the output indicated by *OutputSpecification* to reduce the sweep or waveform at each run to a single numeric value. The *OutputFunction* must be one of the following keywords:

`YMAX` which produces the greatest deviation of the sweep or waveform from the nominal run.

`MAX` which results in the maximum value in each sweep or waveform.

`MIN` which results in the minimum value in each sweep or waveform.

`RISE_EDGE(`*Value*`)` which reports as the result the first run when the waveform crosses above the threshold *Value*. The algorithm used requires that one point in the waveform be below *Value* and the succeeding point be above *Value*.

`FALL_EDGE(`*Value*) which reports as the result the first run when the waveform crosses below the threshold *Value*. The algorithm used requires that one point in the waveform be above *Value* and the succeeding point be below *Value*.

`RANGE` is an optional range indicating the range of the sweep variable over which *OutputFunction* is to be performed. If this keyword is missing output is produced the range is not restricted. The range of the sweep variable to be considered is from *LowValue* to *HighValue* inclusive.

LowValue is the low end of the sweep variable to be considered in evaluating *OutputFunction*.

HighValue is the low end of the sweep variable to be considered in evaluating *OutputFunction*.

`LIST` is an optional keyword that results in the model parameter values that are statistically varied being printed out prior to each run. If it is omitted then the the statistically generated model parameter values are not produced prior to each run.

`OUTPUT` is an optional keyword indicating the type of output to be produced by runs after the initial nominal run. The output produced for each run sampled is determined by the `.PLOT`, `.PRINT` and `.PROBE` statements in the NETLIST. If this keyword is missing output is produced only for the nominal run.

SEED is the keyword for the seed of the random number generator used in Monte Carlo Analysis

SeedValue is the value of the seed used in the random number generator used to select sample runs at random. (Optional; Default: 17,533; $1 \leq$ *SeedValue*$\leq 32,767$)

Note

1. If the *AnalysisType* is `DC` only one independent voltage or current source can be specified in the `.DC` statement (discussed on page 55).
2. The random number generator is the subtractive method generator described by Knuth [28, p. 171].
3. The initial run uses the nominal parameter values given in the NETLIST. Subsequent runs statistically vary model parameters indicated as having either lot `LOT` or device `DEV` tolerances. These tolerances are specified in a `.MODEL` statement (see page 72).

.WIDTH Width Specification

Form

`.WIDTH OUT=`*ColumnWidth*

`OUT` is the keyword for column width of the output file.

ColumnWidth is the column width of the output file. It must be either 80 or 132. If there is no `.WIDTH` statement the ColumnWidth defaults to 80.

Example

```
.WIDTH OUT=80
.WIDTH OUT=132
```

Chapter 7

Element Catalog

A Convolution

Form

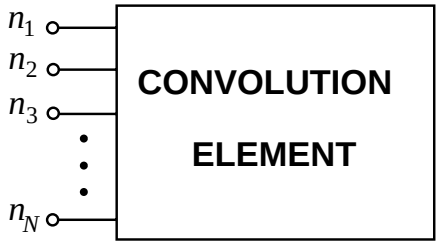


Figure 7.1: A — convolution element.

A*name* n_1 n_2 [n_3 ... n_N] *ModelName*

n_1 is the first node (required),

n_2 is the second node (required),

n_3 is the third node (optional),

n_N is the Nth node (optional),

ModelName is the model name which defines this convolution element.

Example

```
ANET 1 2 3 4 NETWORK1
```

Model Type

CONV

CONV Model — Convolution Model

Form

.MODEL *ModelName* CONV([[*keyword* = *value*] ...]

Example

```
.MODEL CONV1 CONV( FILE = "coupledline.y" NFREQS = 500
+ NPORTS = 4 ZM = 50 THRESHOLD = 0.01)
```

B GaAs MESFET

(versions: PSpice)

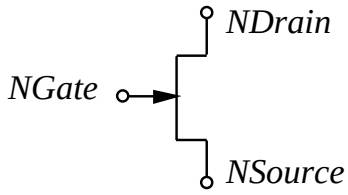


Figure 7.2: B — GASFET element.

PSpice*Form*

Bname *NDrain NGate NSource ModelName* [*Area*]

NDrain is the drain node

NGate is the gate node

NSource is the source node

ModelName is the model name

Area is the area factor in dimensionless units
(Units: none; Optional; Default: 1; Symbol: *Area*)

Example

```
B1 1 2 3 GAAS12
B1 1 2 3 GAAS12 0.5
```

Model Type

GASFET

GASFET Model GaAs MESFET Model

Form

.MODEL *ModelName* GASFET([[*keyword* = *value*] ...])

Example

```
.MODEL GAAS12 GASFET( LEVEL=1 )
```

PSpice provides three MESFET device models some microwave versions provide six. The parameter LEVEL specifies the model to be used:

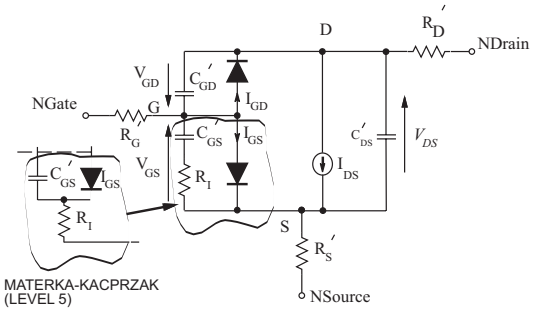


Figure 7.3: Schematic of the GASFET model. V_{GS}, V_{DS}, and V_{GD} are intrinsic gate-source, drain-source and gate-drain voltages between the internal gate, drain, and source terminals designated G, D, and S respectively. R_I is not used in PSpice.

LEVEL = 1 → Curtice Quadratic model
This was the first widely accepted model for a GaAs MESFET and described in [11]. It uses rather simple empirical fits to measured data. See page 120. (VERSIONS: PSPICE)

LEVEL = 2 → Raytheon model
This model is also known as the Statz model and model was developed at Raytheon for the modeling of GaAs MESFETs used in digital circuits. It is also based on empirical fits to measured data [23]. See page 120. (VERSIONS: PSPICE)

LEVEL = 3 → TOM or TriQuint model
The name of this model derives from *TriQuint's Own Model* [24]. See page 120. (VERSIONS: PSPICE)

LEVEL = 4 → Curtice-Ettenberg Cubic model
This is a refinement on the LEVEL 1 model [12]. It also uses simple empirical fits to measured data. See page 120.

LEVEL = 5 → Materka-Kacprzak model [13]
A distinguishing characteristic is that the drain-source current is analytic and so it has better convergence characteristics than the other models. See page 138.

LEVEL = 6 → Angelov model
Another empirical GASFET model with analytic characteristics. See page 143.

LEVEL = -1 → TOM-2 model
An improved model from TriQuint. Another empirical GASFET model with analytic characteristics. See page 136.

LEVEL 1, 2, 3 and 4 GASFET models

Many parameters of the LEVEL 1, 2, 3 and 4 `GASFET` models are the same and so these models will be considered together. The parameter keywords are given in table **??**. It is assumed that the model parameters were determined or measured at the nominal temperature $T_{\rm NOM}$ (default 27°C) specified in the most recent `.OPTIONS` statement preceding the `.MODEL` statement.

Table 7.2: MESFET model parameters.

Name	Description	Units	Default
`A0`	drain saturation current for $V_{GS} = 0$ (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_0)	A	0.1
`A1`	coefficient of V_1 (primary transconductance parameter) (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_1)	A/V	0.05
`A2`	coefficient of V_1^2 (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_1)	A/V^2	0
`A3`	coefficient of V_1^3 (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_1)	A/V^3	0
`AA0`	temperature cofficient of `A0` (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_{A0})	A	0
`AA1`	temperature cofficient of `A1` (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_{A1})	A	0
`AA2`	temperature cofficient of `A2` (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_{A2})	A	0
`AA3`	temperature cofficient of `A3` (`LEVEL=4`) (VERSION: SOMEVERSIONSOFSPICE) (A_{A3})	A	0
`AALF`	linear temperature coefficient of `ALPHA` (`LEVEL=1,2,3,-1`) (VERSION: SOMEVERSIONSOFSPICE) (A_α)	$°C^{-1}$	0
`AB`	linear temperature coefficient of `B` (`LEVEL=2,3`) (VERSION: SOMEVERSIONSOFSPICE) (A_B)	$°C^{-1}$	0
`ABET`	linear temperature coefficient of `BETA` (VERSION: SOMEVERSIONSOFSPICE) (A_β)	%/°C	0
`ACGS`	linear temperature coefficient of `CGS` (`LEVEL=1,3,4,-1`) (VERSION: SOMEVERSIONSOFSPICE) (A_{CGS})	$°C^{-1}$	0
`ACGD`	linear temperature coefficient of `CGD` (`LEVEL=1,3,4,-1`) (VERSION: SOMEVERSIONSOFSPICE) (A_{CGD})	$°C^{-1}$	0
`ADEL`	linear temperature coefficient of `DELTA` (`LEVEL=3,-1`) (VERSION: SOMEVERSIONSOFSPICE) (A_δ)	$°C^{-1}$	0
`AF`	flicker noise exponent (A_F)	-	1
`AGAM`	linear temperature coefficient of `GAMA` (`LEVEL=1,3,4`) (VERSION: SOMEVERSIONSOFSPICE) (A_λ)	%/°C	0
`ALAM`	linear temperature coefficient of `LAMBDA` (`LEVEL=1,2,3`) (VERSION: SOMEVERSIONSOFSPICE) (A_λ)	%/°C	0
`ALPHA`	saturation voltage parameter (`LEVEL=1,2,3,-1`) (α)	V^{-1}	2
`ALFA`	alternative keyword of `ALPHA` (`LEVEL=1,2,3,-1`) (VERSION: SOMEVERSIONSOFSPICE) (α)	V^{-1}	2

Continued on next page

Table 7.2: MESFET model parameters.

Name	Description	Units	Default
AQ	linear temperature coefficient of Q (LEVEL=3) (VERSION: SOMEVERSIONSOFSPICE) (A_Q)	$°C^{-1}$	0
AR1	linear temperature coefficient of R1 (LEVEL=1,4) (VERSION: SOMEVERSIONSOFSPICE) (A_{R1})	$°C^{-1}$	0
AR2	linear temperature coefficient of R2 (LEVEL=1,4) (VERSION: SOMEVERSIONSOFSPICE) (A_{R2})	$°C^{-1}$	0
ALPHATCE	exponential temperature coefficient of ALPHA (LEVEL=-1) ($T_{C,\alpha}$)	%/°C	0
ARD	alternative keyword for TRD1 (A_{RD})	$°C^{-1}$	0
ARF	linear temperature coefficient of RF (LEVEL=1,4) (A_{RF})	$°C^{-1}$	0
ARG	alternative keyword for TRG1 (A_{RG})	$°C^{-1}$	0
ARI	linear temperature coefficient of RI (A_{RI})	$°C^{-1}$	0
ARS	alternative keyword for TRS1 (A_{RS})	$°C^{-1}$	0
AT	linear temperature coefficient of TAU (A_τ)	$°C^{-1}$	0
AU	linear temperature coefficient of U (LEVEL=1,3) (A_U)	$°C^{-1}$	0
AVDS	linear temperature coefficient of VDS0 (LEVEL=4) (A_{VDS0})	$°C^{-1}$	0
AVBD	linear temperature coefficient of VBD (A_{VBD})	$°C^{-1}$	0
AVT0	(AVT-zero) linear temperature coefficient of VTO (A_{VT0})	$°C^{-1}$	0
B	doping tail extending parameter (B) (VERSION: PSPICE) (LEVEL=2) (VERSION: SOMEVERSIONSOFSPICE) (LEVEL=2,3)	V^{-1}	0.3
BETA	transconductance coefficient (LEVEL=1,2,3) (β)	A/V^2	0.1
BETA	transconductance coefficient (LEVEL=4) (VERSION: SOMEVERSIONSOFSPICE) (β)	$1/V^2$	0
BETATCE	exponential temperature coefficient of BETA ($T_{C,\beta}$)	%/°C	0
BRD	quadratic temperature coefficient of RD (VERSION: SOMEVERSIONSOFSPICE) (B_{RD})	$°C^{-2}$	0
BRG	quadratic temperature coefficient of RG (VERSION: SOMEVERSIONSOFSPICE) (B_{RG})	$°C^{-2}$	0
BRI	quadratic temperature coefficient of RI (VERSION: SOMEVERSIONSOFSPICE) (B_{RI})	$°C^{-2}$	0
BRS	quadratic temperature coefficient of RS (VERSION: SOMEVERSIONSOFSPICE) (B_{RD})	$°C^{-2}$	0
BVT0	quadratic temperature coefficient of VT0 (VERSION: SOMEVERSIONSOFSPICE) (B_{VT0})	$°C^{-2}$	0
CDS	drain-source capacitance (C_{DS})	F	0
CGD	zero-bias gate-drain p-n capacitance (C_{GD})	F	0
CGD0	alternative keyword for CGD (VERSION: SOMEVERSIONSOFSPICE) (C_{GD})	F	0
CGS	zero-bias gate-source p-n capacitance (C_{GS})	F	0
CGS0	alternative keyword for CGS (VERSION: SOMEVERSIONSOFSPICE) (C_{GS})	F	0
DELTA	output feedback parameter (LEVEL=3) (δ)	$(AV)^{-1}$	0
DELTA	output feedback parameter (LEVEL=-1) (δ)	$(AV)^{-1}$	0.2

Continued on next page

Table 7.2: MESFET model parameters.

Name	Description	Units	Default
DELT	alternative keyword for DELTA (LEVEL=3) (VERSION: SOMEVERSIONSOFSPICE) (δ)	$(AV)^{-1}$	0
DELT	alternative keyword for DELTA (LEVEL=-1) (VERSION: SOMEVERSIONSOFSPICE) (δ)	$(AV)^{-1}$	0
DLVL	breakdown model flag not used DLVL = 1 use original model = 2 use enhanced model	-	1
E	drain current power law coefficient (VERSION: SOMEVERSIONSOFSPICE) (LEVEL=1) (E)	-	2
EG	bandgap voltage (barrier height) at 0 K ($E_G(0)$) Schottky Barrier Diode: 0.69 Silicon: 1.16 Gallium Arsenide: 1.52 Germanium: 0.67 (LEVEL=1,2,3,4)	eV	1.52
EG	bandgap voltage (barrier height) at 0 K ($E_G(0)$) Schottky Barrier Diode: 0.69 Silicon: 1.16 Gallium Arsenide: 1.52 Germanium: 0.67 (LEVEL=-1)	eV	1.11
FC	forward-bias depletion capacitance factor (F_C)	-	0.5
GAMMA	Static feedback parameter also known as voltage slope parameter of pinch-off voltage (VERSION: PSPICE) (LEVEL=3) (VERSION: SOMEVERSIONSOFSPICE) (LEVEL=1,3) (γ)	-	0
GAMA	alternative keyword for GAMMA (VERSION: SOMEVERSIONSOFSPICE) (LEVEL 1,3,-1) (γ)	-	0
GAMA	Slope of drain characteristic in the linear region (LEVEL=4) (VERSION: SOMEVERSIONSOFSPICE) (γ')	-	1.5
GAMMATCE	exponential temperature coefficient of GAMMA ($T_{C,\gamma}$)	%/°C	0
GAP1	First bandgap correction factor (F_{GAP1}) Silicon: 0.000473 Old Value for Silicon: 0.000702 Gallium Arsenide: 0.000541 Germanium: 0.000456 (VERSION: HSPICE; SOMEVERSIONSOFSPICE)	eV/°C	0.000541
GAP2	Second bandgap correction factor (F_{GAP2}) Silicon: 0.000636 Old Value for Silicon: 0.001108 Gallium Arsenide: 0.000204 Germanium: 0.000210 (VERSION: HSPICE; SOMEVERSIONSOFSPICE)	°C	0.000204
GMAX	enhanced breakdown model parameter not used	S	0
IS	gate p-n saturation current (I_S)	A	1E-14
K1	enhanced breakdown model parameter not used	V^{-1}	0

Continued on next page

Table 7.2: MESFET model parameters.

Name	Description	Units	Default
K2	enhanced breakdown model parameter not used	V	0
K3	enhanced breakdown model parameter not used	V^2	0
KF	flicker noise coefficient (K_F)	-	0
LAMBDA	channel-length modulation (LEVEL=1,2,3,-1) (λ)	V^{-1}	0
LAMB	alternative keyword for LAMBDA (LEVEL=1,2,3) (VERSION: SOMEVERSIONSOFSPICE) (λ)	V^{-1}	0
LEVEL	model index 1 → Curtice quadratic model; 2 → Raytheon model; 3 → TOM (Triquint) model; 4 → Curtice cubic model; 5 → Materka-Kacprzak model; 6 → Angelov model; LEVELs 4,5,6 SOMEVERSIONSOFSPICE only	-	1
M	gate p-n grading coefficient (M)	-	0.5
MGS	gate-source p-n grading coefficient (VERSION: SOMEVERSIONSOFSPICE) (M_{GS})	-	M
MGD	gate-drain p-n grading coefficient (VERSION: SOMEVERSIONSOFSPICE) (M_{GS})	-	M
N	gate p-n emission coefficient (n)	-	1
NG	constant part of threshold ideality factor LEVEL -1 SOMEVERSIONSOFSPICE only (N_G)	-	1
ND	part of threshold ideality factor that depends on V_{DS} LEVEL -1 SOMEVERSIONSOFSPICE only (N_D)	V^{-1}	0
NPLT	(NOT USED) (VERSION: SOMEVERSIONSOFSPICE) (V_{GMN})	s	0
Q	power-law parameter (LEVEL=3,-1) (Q)	-	2
R1	breakdown gate-drain resistance (LEVEL=1,4) (VERSION: SOMEVERSIONSOFSPICE) NOT USED (R_1)	Ω	∞
R2	breakdown dependency on channel current (LEVEL=1,4) (VERSION: SOMEVERSIONSOFSPICE) (LEVEL=1,4) NOT USED (R_2)	Ω	0
RD	drain resistance (R_D)	Ω	0
RF	forward-biased gate-source resistance (LEVEL=1,4) (VERSION: SOMEVERSIONSOFSPICE) (R_F)	Ω	∞
RG	gate resistance (R_G)	Ω	0
RI	channel resistance (NOT USED) (VERSION: SOMEVERSIONSOFSPICE) (R_I)	Ω	0
RS	source resistance (R_S)	Ω	0
T	alternative keyword for TAU (VERSION: SOMEVERSIONSOFSPICE) (τ)	s	0
TBET	alternative keyword for BETATCE (VERSION: SOMEVERSIONSOFSPICE) ($T_{C,\beta}$)	s	0
TJ	junction temperature (NOT USED) (VERSION: SOMEVERSIONSOFSPICE) (T_J)	s	0
TM	(NOT USED) (VERSION: SOMEVERSIONSOFSPICE) (T_M)	s	0
TME	(NOT USED) (VERSION: SOMEVERSIONSOFSPICE) (T_{ME})	s	0

Continued on next page

Table 7.2: MESFET model parameters.

Name	Description	Units	Default
TNOM	nominal temperature (NOT USED) (VERSION: SOMEVERSIONSOFSPICE) ($T_{\rm NOM}$)	s	0
TAU	conduction current delay time (τ)	s	0
TRG1	linear temperature coefficient of RG (A_{RG})	$^\circ$C^{-1}	0
TRD1	linear temperature coefficient of RD (A_{RD})	$^\circ$C^{-1}	0
TRS1	linear temperature coefficient of RS (A_{RS})	$^\circ$C^{-1}	0
U	critical field parameter for mobility degradation (LEVEL=1,2,3) (VERSION: SOMEVERSIONSOFSPICE) (U)	V/m	0
VBI	gate p-n potential ($V_{\rm BI}$)	V	1
VBR	enhanced breakdown model parameter not used	V^2	∞
VBD	breakdown voltage (VERSION: SOMEVERSIONSOFSPICE) ($A_{VBD)}$)	V	∞
VDELTA	capacitance transistion voltage (LEVEL=2,3) (V_Δ)	V	0.2
VDSO	V_{DS} at which BETA was measured (LEVEL=4) (V_Δ)	V	4
VGMN	(NOT USED) (V_{GMN})	s	0
VGMX	(NOT USED) (V_{GMN})	s	0
VDMX	(NOT USED) (V_{GMN})	s	0
VMAX	capacitance limiting voltage (LEVEL=2,3) ($V_{\rm MAX}$)	V	0.5
VTO	(VT-oh) pinch-off voltage (V_{T0})	V	-2.5
VTO	(VT-0) alternative keyword for VTO (VERSION: SOMEVERSIONSOFSPICE) ($A_{VDS0)}$)	V	-2.5
VTOTC	linear temperature coefficient of VTO ($T_{C,VT0}$)	V/$^\circ$C	0
XTI	temperature exponent of IS (X_{TI})	-	0
VBITC	linear temperature coefficient of VBI ($T_{C,BI}$)	-	0

Some versions of SPICE use incorrect default values for some of the parameters. One example is the default value for EG. The accepted value has changed with time. It is always a good idea not to rely on default values other than 0.

The physical constants used in the model evaluation are

k	Boltzmann's constant	$1.3806226\,10^{-23}$ J/K
q	electronic charge	$1.6021918\,10^{-19}$ C

Standard Calculations

Absolute temperatures (in kelvins, K) are used. The thermal voltage $V_{\rm TH} = kT/q$ and the band gap energy at the nominal temperature is

$$E_G(T_{\rm NOM}) = E_G(0) - F_{\rm GAP1} 4T_{\rm NOM}^2/\left(T_{\rm NOM} + F_{\rm GAP2}\right). \tag{7.1}$$

Here $E_G(0)$ is the parameter EG — the band gap energy at 0 K. $F_{\rm GAP1}$ and $F_{\rm GAP2}$ are not parameters in PSPICE. PSPICE documentation indicates that PSPICE uses $F_{\rm GAP1} = 0.000702$ and $F_{\rm GAP2} = 1108$.

Temperature Dependence

Temperature effects are incorporated as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K).

$$\alpha(T) = \alpha(T_{\rm NOM})\left(1.01^{(T_{C,\alpha}(T - T_{\rm NOM}))} + A_\alpha(T - T_{\rm NOM})\right) \tag{7.2}$$

$$\beta(T) = \beta(T_{\rm NOM})\left(1.01^{(T_{C,\beta}(T - T_{\rm NOM}))} + A_\beta(T - T_{\rm NOM})\right) \tag{7.3}$$

$$I_S(T) = I_S(T_{\rm NOM})e^{\left(E_G(T)T/T_{\rm NOM} - E_G(T)\right)/(nV_{\rm TH})}\,(T/T_{\rm NOM})^{\left(X_{TI}/n\right)} \tag{7.4}$$

$$C'_{GS}(T) = \begin{cases} C_{GS}(T_{\rm NOM})\left\{1 + M_{GS}\left[0.0004(T - T_{\rm NOM}) + \left(1 - \dfrac{V_{BI}(T)}{V_{BI}(T_{\rm NOM})}\right)\right]\right\} \\ \qquad\qquad A_{CGS} \text{ not specified} \\ C_{GS}(T_{\rm NOM})(1 + A_{CGS}(T - T_{\rm NOM})) \qquad A_{CGS} \text{ specified} \end{cases}$$

$$C'_{GD}(T) = \begin{cases} C_{GD}(T_{\rm NOM})\left\{1 + M_{GD}\left[0.0004(T - T_{\rm NOM}) + \left(1 - \dfrac{V_{BI}(T)}{V_{BI}(T_{\rm NOM})}\right)\right]\right\} \\ \qquad\qquad A_{CGD} \text{ not specified} \\ C_{GD}(T_{\rm NOM})(1 + A_{CGD}(T - T_{\rm NOM})) \qquad A_{CGD} \text{ specified} \end{cases}$$

$$E_G(T) = E_G(0) - F_{\rm GAP1}4T^2/\left(T + F_{\rm GAP2}\right) \tag{7.5}$$

$$\lambda(T) = \lambda(T_{\rm NOM})(1 + A_\lambda(T - T_{\rm NOM})) \tag{7.6}$$

$$\alpha(T) = \begin{cases} \alpha(T_{\rm NOM})(1 + A_\alpha(T - T_{\rm NOM})) & \text{LEVEL = 1,2,3} \\ \alpha(T_{\rm NOM})(1.01^{T_{C,\alpha}(T - T^{\rm NOM})} + A_\alpha(T - T_{\rm NOM})) & \\ A_\alpha(T - T_{\rm NOM})) & \text{LEVEL = -1} \end{cases} \tag{7.7}$$

$$U(T) = U(T_{\rm NOM})(1 + A_U(T - T_{\rm NOM})) \tag{7.8}$$

$$A_0(T) = A_0(T_{\rm NOM})(1 + A_{A0}(T - T_{\rm NOM})) \tag{7.9}$$

$$A_1(T) = A_1(T_{\rm NOM})(1 + A_{A1}(T - T_{\rm NOM})) \tag{7.10}$$

$$A_2(T) = A_2(T_{\rm NOM})(1 + A_{A2}(T - T_{\rm NOM})) \tag{7.11}$$

$$A_3(T) = A_3(T_{\rm NOM})(1 + A_{A3}(T - T_{\rm NOM})) \tag{7.12}$$

$$\delta(T) = \delta(T_{\rm NOM})(1 + A_\delta(T - T_{\rm NOM})) \tag{7.13}$$
$$\gamma(T) = \gamma(T_{\rm NOM})(1 + A_\gamma(T - T_{\rm NOM})) \tag{7.14}$$
$$Q(T) = Q(T_{\rm NOM})(1 + A_Q(T - T_{\rm NOM})) \tag{7.15}$$
$$R_1(T) = R_1(T_{\rm NOM})(1 + A_{R1}(T - T_{\rm NOM})) \tag{7.16}$$
$$R_2(T) = R_2(T_{\rm NOM})(1 + A_{R2}(T - T_{\rm NOM})) \tag{7.17}$$
$$R_D(T) = R_D(T_{\rm NOM})\left(1 + A_{RD}(T - T_{\rm NOM}) + B_{RD}(T - T_{\rm NOM})^2\right) \tag{7.18}$$
$$R_F(T) = R_F(T_{\rm NOM})(1 + A_{RF}(T - T_{\rm NOM})) \tag{7.19}$$
$$R_G(T) = R_G(T_{\rm NOM})\left(1 + A_{RG}(T - T_{\rm NOM}) + B_{RG}(T - T_{\rm NOM})^2\right) \tag{7.20}$$
$$R_S(T) = R_S(T_{\rm NOM})\left(1 + A_{RS}(T - T_{\rm NOM}) + B_{RS}(T - T_{\rm NOM})^2\right) \tag{7.21}$$
$$R_I(T) = R_I(T_{\rm NOM})\left(1 + A_{RI}(T - T_{\rm NOM}) + B_I(T - T_{\rm NOM})^2\right) \tag{7.22}$$

$$\tau(T) = \tau(T_{\rm NOM})(1 + A_\tau(T - T_{\rm NOM})) \tag{7.23}$$
$$V_{BD}(T) = V_{BD}(T_{\rm NOM})(1 + A_{VBD}(T - T_{\rm NOM})) \tag{7.24}$$
$$V_{BI}(T) = \begin{cases} V_{BI}(T_{\rm NOM})T/T_{\rm NOM} - 3V_{\rm TH}\ln(T/T_{\rm NOM}) & \texttt{LEVEL} \neq -1 \\ V_{BI}(T_{\rm NOM})T/T_{\rm NOM} + T_{C,VBI}(T - T_{\rm NOM}) & \texttt{LEVEL} = -1 \end{cases}$$
$$+ E_G(T_{\rm NOM})T/T_{\rm NOM} - E_G(T) \tag{7.25}$$
$$V_{DS0}(T) = V_{DS0}(T_{\rm NOM})(1 + A_{VDS0}(T - T_{\rm NOM})) \tag{7.26}$$
$$V_{T0}(T) = V_{T0}\left(1 + A_{VT0}(T - T_{NOM}) + B_{VT0}(T - T_{NOM})^2\right) + T_{C,VT0}(T - T_{NOM}) \tag{7.27}$$
$$V_{VMAX}(T) = V_{VMAX}(T_{\rm NOM}) + T_{C,VMAX}(T - T_{\rm NOM})\texttt{LEVEL} = -1 \tag{7.28}$$

Parasitic Resistances

The resistive parasitics R'_S, and R'_D are calculated from the sheet resistivities `RS` ($= R_S$) and `RD` ($= R_D$), and the *Area* specified on the element line. `RG` ($= R_G$) is used as supplied.

$$R'_D = R_D/Area \tag{7.29}$$
$$R'_G = \begin{cases} R_G & \text{PSpice} \\ R_G/Area & \text{SomeVersionsOfSpice} \end{cases} \tag{7.30}$$
$$R'_S = R_S/Area \tag{7.31}$$

The parasitic resistance parameter dependencies are summarized in figure 7.4.

Leakage Currents

Current flows across the normally reverse biased gate-source and gate-drain junctions. The gate-source leakage current $I_{GS} = Area\, I_S e^{(V_{GS}/V_{\rm TH} - 1)}$ (7.32)
and the gate-drain leakage current $I_{GD} = Area\, I_S e^{(V_{GD}/V_{\rm TH} - 1)}$ (7.33)

The dependencies of the parameters describing the leakage current are summarized in figure 7.5.

KEYWORD PARAMETERS	
RD	R_D
RG	R_G
RS	R_S

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$R'_D = f(Area, R_D)$
$R'_G = f(Area, R_G)$
$R'_S = f(Area, R_S)$

Figure 7.4: MESFET parasitic resistance parameter relationships.

KEYWORD PARAMETERS	
IS	I_S

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$I_{GS} = f(I_S, Area)$
$I_{GD} = f(I_S, Area)$

Figure 7.5: GASFET leakage current parameter dependencies.

LEVEL 1 (Curtice Model)

LEVEL 1 (Curtice Model) I/V Characteristics

The LEVEL 1 current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point. Curtice [11] proposed two DC current models: a quadratic channel current model and a cubic channel current model. The quadratic channel model is implemented as the LEVEL 1 model.

Normal Mode: ($V_{DS} \geq 0$)

The regions of operation are defined as follows with $V_{GST} = V_{GS} - (V_{T0} - \gamma V_{DS})$ (7.34)

cutoff region:	$V_{GST}(t-\tau) \leq 0$
linear and saturation regions:	$V_{GST}(t-\tau) > 0$

Then

$$I_{DS} = \begin{cases} 0 & \text{cutoff region} \\ Area \dfrac{\beta\,(1+\lambda V_{DS})}{1+UV_{GST}(t-\tau)} V^E_{GST}(t-\tau)\tanh\left(\alpha V_{DS}\right) & \text{linear, saturation regions} \end{cases} \tag{7.35}$$

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the MESFET I/V characteristics are evaluated as in the normal mode (7.35) but with the drain and source subscripts exchanged, and V_{GD} is the controlling voltage instead of V_{GS}.

The relationships of the parameters describing the I/V characteristics for the LEVEL 1 model are summarized in figure 7.6.

LEVEL 1 (Curtice Model) Capacitances

KEYWORD PARAMETERS	
ALPHA	α
BETA	β
LAMBDA	λ
U	U
VTO	V_{T0}

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$I_{DS} = f(Area, \alpha, \beta, \lambda, U, V_{T0})$

Figure 7.6: LEVEL 1 (Curtice model) I/V dependencies.

The drain-source capacitance

$$C'_{DS} = Area\, C_{DS} \tag{7.36}$$

The gate-source capacitance

$$C'_{GS} = \begin{cases} Area\, C_{GS} \left(1 - \frac{V_{GS}}{V_{BI}}\right)^{-M_{GS}} & V_{GS} \le F_C V_{BI} \\ Area\, C_{GS}\, (1 - F_C)^{-(1+M_{GS})} \left[1 - F_C(1 + M_{GS}) + M_{GS}\frac{V_{GS}}{V_{BI}}\right] & V_{GS} > F_C V_{BI} \end{cases} \tag{7.37}$$

The gate-drain capacitance

$$C'_{GD} = \begin{cases} Area\, C_{GD} \left(1 - \frac{V_{GD}}{V_{BI}}\right)^{-M_{GD}} & V_{GD} \le F_C V_{BI} \\ Area\, C_{GD}\, (1 - F_C)^{-(1 + M_{GD})} \left[1 - F_C(1 + M_{GD}) + M_{GD}\frac{V_{GD}}{V_{BI}}\right] & V_{GS} > F_C V_{BI} \end{cases} \tag{7.38}$$

The LEVEL 1 capacitance parameter dependencies are summarized in figure 7.7.

KEYWORD PARAMETERS	
CGD	C_{GD}
CGS	C_{GS}
CDS	C_{DS}
FC	F_C
VBI	V_{BI}
MGS	M_{GS}
MGD	M_{GD}

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$C'_{DS} = f(Area, C_{DS})$
$C'_{GD} =$ $f(Area, C_{GD}, F_C, V_{BI}, M_{GD})$
$C'_{GS} =$ $f(Area, C_{GS}, F_C, V_{BI}, M_{GS})$

Figure 7.7: LEVEL 1 (Curtice model) capacitance dependencies.

LEVEL 2 (Raytheon Model)

LEVEL 2 (Raytheon Model) I/V Characteristics

The LEVEL 2 current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \ge 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \ge 0$)

The regions of operation are defined as follows with $V_{GST} = V_{GS} - V_{T0}$ (7.39)

cutoff region:	$V_{GST}(t-\tau) \leq 0$
linear region:	$V_{GST}(t-\tau) > 0$ and $V_{DS} \leq 3/\alpha$
saturation region:	$V_{GS}(t-\tau) > V_{T0}$ and $V_{DS} > 3/\alpha$

Then

$$I_{DS} = \begin{cases} 0 & \text{cutoff region} \\ Area \dfrac{\beta}{1+UV_{GST}} (1+\lambda V_{DS}) \dfrac{V_{GST}(t-\tau)^2}{1+BV_{GST}(t-\tau)} \text{Ktanh} & \text{linear and saturation regions} \end{cases} \quad (7.40)$$

where

$$\text{Ktanh} = \begin{cases} 1 - \left(1 - V_{DS}\frac{\alpha}{3}\right)^3 & \text{linear region} \\ 1 & \text{saturation regions} \end{cases} \quad (7.41)$$

is a Taylor series approximation to the tanh function of the `LEVEL` 1 model.

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the MESFET I/V characteristics are evaluated as in the normal mode (7.40) but with the drain and source subscripts exchanged, and V_{GD} is the controlling voltage instead of V_{GS}.

The relationships of the parameters describing the I/V characteristics of the `LEVEL` 2 model are summarized in figure 7.8.

KEYWORD PARAMETERS	
`ALPHA`	α
`B`	B
`BETA`	β
`LAMBDA`	λ
`U`	μ
`VTO`	V_{T0}

+

GEOMETRY PARAMETER
$Area$

→

DEVICE PARAMETERS
$I_{DS} = f(Area, \alpha, B, \beta, \lambda, UV_{T0})$

Figure 7.8: LEVEL 2 (Raytheon model) I/V dependencies.

LEVEL 2 (Raytheon Model) Capacitances

This is a symmetrical capacitance model. The drain-source capacitance $C'_{DS} = Area\, C_{DS}$ (7.42)
The gate-source capacitance

$$C'_{GS} = Area \left[C_{GS} F_1 F_2 \left(1 - \frac{V_{\rm NEW}}{V_{BI}} \right)^{-\frac{1}{2}} + C_{GD} F_3 \right] \tag{7.43}$$

The gate-drain capacitance

$$C'_{GD} = Area \left[C_{GS} F_1 F_3 \left(1 - \frac{V_{\rm NEW}}{V_{BI}} \right)^{-\frac{1}{2}} + C_{GD} F_2 \right] \tag{7.44}$$

where

$$F_1 = \frac{1}{2} \left\{ 1 + \frac{V_{\rm EFF} - V_{T0}}{\sqrt{(V_{\rm EFF} - V_{T0})^2 + V_\Delta^2}} \right\} \tag{7.45}$$

$$F_2 = \frac{1}{2} \left\{ 1 + \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.46}$$

$$F_3 = \frac{1}{2} \left\{ 1 - \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.47}$$

$$V_{\rm EFF} = \frac{1}{2} \left\{ V_{GS} + V_{GD} + \sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}} \right\} \tag{7.48}$$

$$V_{\rm NEW} = \begin{cases} A_1 & A_1 < V_{\rm MAX} \\ V_{\rm MAX} & A_1 \geq V_{\rm MAX} \end{cases} \tag{7.49}$$

$$A_1 = \frac{1}{2} \left[V_{\rm EFF} + V_{T0} + \sqrt{(V_{\rm EFF} - V_{T0})^2 + V_\Delta^2} \right] \tag{7.50}$$

The capacitance parameter dependencies are summarized in figure 7.9. The above capacitance model

KEYWORD PARAMETERS	
ALPHA	α
CGD	C_{GD}
CGS	C_{GS}
CDS	C_{DS}
VBI	V_{BI}
VTO	V_{T0}
VDELTA	V_Δ
VMAX	$V_{\rm MAX}$

+

GEOMETRY PARAMETER
$Area$

→

DEVICE PARAMETERS
$C'_{DS} = f(Area, C_{DS})$
$C'_{GD} = f(Area, C_{GD}, \alpha, B, F_C, V_{BI}, V_{T0})$
$C'_{GS} = f(Area, C_{GS}, \alpha, B, F_C, V_{BI}, V_{T0})$

Figure 7.9: LEVEL 2 (Raytheon model) capacitance dependencies.

does not satisfy drain-source charge conservation. Over a cycle, charge can be pumped from the drain to the source. In practice this is often not a problem when this capacitance model is used buyt the user must be ware. If it is a problem the user will see a periodic reponse can not be obtained even if the excotation is periodic. For a further discussion see References [26, 27].

The above capacitance model does not satisfy drani-source charge conservation. Over a cycle charge can be pumped from the drain to the source. In practice this is often not a problem when this capacitance model is used but the user must be wary. If it is a problem the user will see that a periodic response can not be obtained even if the excitation is periodeic. For a further discussion see References [26] and [27].

LEVEL 3 (TOM Model)

The `LEVEL` 3 model is an implementation of the TOM model ("Triquint's Own Model) [24].

`LEVEL` *3 (TOM Model) I/V Characteristics*

The `LEVEL 3` current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.
Normal Mode: ($V_{DS} \geq 0$)

The regions of operation are defined as follows with $V_{GST} = V_{GS} - V_P$ (7.51)
and $V_P = V_{T0} - \gamma V_{DS}$ (7.52)

cutoff region:	$V_{GST}(t-\tau) \leq 0$
linear region:	$V_{GST}(t-\tau) > 0$ and $V_{DS} \leq 3/\alpha$
saturation region:	$V_{GST}(t-\tau) > 0$ and $V_{DS} > 3/\alpha$

Then

$$I_{DS} = Area\, I_{DS0}/(1 + \delta I_{DS0} V_{DS}) \tag{7.53}$$

$$I_{DS0} = \begin{cases} 0 & \text{cutoff region} \\ \beta(1+\lambda V_{DS}) V_{GST}^{Q}(t-\tau)\text{Ktanh} & \text{linear and saturation regions} \end{cases} \tag{7.54}$$

$$\text{Ktanh} = \begin{cases} 1 - \left(1 - V_{DS}\frac{\alpha}{3}\right)^3 & \text{linear region} \\ 1 & \text{saturation region} \end{cases} \tag{7.55}$$

Ktanh is a taylor series approximation to the tanh function of the `LEVEL` 1 model.

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the MESFET I/V characteristics are evaluated as in the normal mode (7.75) but with the drain and source subscripts exchanged, and V_{GD} used as the controlling voltage instead of V_{GS}.
The following description does apply to SuperSpice.
The relationships of the parameters describing the I/V characteristics of the `LEVEL` 3 model are summarized in figure 7.13.

LEVEL 3 (TOM Model) Capacitances

KEYWORD PARAMETERS	
ALPHA	α
BETA	β
DELTA	δ
GAMMA	γ
LAMBDA	λ
U	μ
VTO	V_{T0}

$+$

GEOMETRY PARAMETER
$Area$

$\rightarrow$

DEVICE PARAMETERS
$I_{DS} = f(Area, \alpha, \beta, \delta, \gamma, \lambda, UV_{T0})$

Figure 7.10: LEVEL 3 (TOM model) I/V dependencies.

The drain-source capacitance $C'_{DS} = Area\, C_{DS}$ (7.56)
The gate-source capacitance

$$C'_{GS} = Area \left[C_{GS} F_1 F_2 \left(1 - \frac{V_{\rm NEW}}{V_{BI}} \right)^{-\frac{1}{2}} + C_{GD} F_3 \right] \tag{7.57}$$

The gate-drain capacitance is given by

$$C'_{GD} = Area \left[C_{GS} F_1 F_3 \left(1 - \frac{V_{\rm NEW}}{V_{BI}} \right)^{-\frac{1}{2}} + C_{GD} F_2 \right] \tag{7.58}$$

$$F_1 = \frac{1}{2} \left\{ 1 + \frac{V_{\rm EFF} - V_P}{\sqrt{(V_{\rm EFF} - V_{T0})^2 + V_\Delta^2}} \right\} \tag{7.59}$$

$$F_2 = \frac{1}{2} \left\{ 1 + \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.60}$$

$$F_3 = \frac{1}{2} \left\{ 1 - \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.61}$$

$$V_{\rm EFF} = \frac{1}{2} \left\{ V_{GS} + V_{GD} + \sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}} \right\} \tag{7.62}$$

$$V_{\rm NEW} = \begin{cases} A_1 & A_1 < V_{\rm MAX} \\ V_{\rm MAX} & A_1 \geq V_{\rm MAX} \end{cases} \tag{7.63}$$

and $$A_1 = \tfrac{1}{2} \left[V_{\rm EFF} + V_P + \sqrt{(V_{\rm EFF} - V_P)^2 + V_\Delta^2} \right] \tag{7.64}$$

The capacitance parameter dependencies are summarized in figure 7.14.

LEVEL -1 (TOM-2 Model)

The LEVEL -1 model is an enhancement of the LEVEL 3 TOM model.

LEVEL -1 (TOM Model) I/V Characteristics

KEYWORD PARAMETERS	
ALPHA	α
CGD	C_{GD}
CGS	C_{GS}
CDS	C_{DS}
VBI	V_{BI}
VTO	V_{T0}
LAMBDA	V_{T0}
VMAX	$V_{\rm MAX}$
VDELTA	V_{Δ}

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$C'_{DS} = f(Area, C_{DS})$
$C'_{GD} = f(Area, C_{GD}, \alpha, B, F_C, V_{BI}, V_{T0})$
$C'_{GS} = f(Area, C_{GS}, \alpha, B, F_C, V_{BI}, V_{T0})$

Figure 7.11: LEVEL 3 (TOM model) capacitance dependencies.

The LEVEL -1 current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \geq 0$)

The regions of operation are defined as follows with $V_{GST} = V_{GS} - V_P$ (7.65)

and $V_P = V_{T0} - \gamma V_{DS}$ (7.66)

cutoff region: $V_{GST}(t-\tau) \leq 0$
linear region: $V_{GST}(t-\tau) > 0$ and $V_{DS} \leq 3/\alpha$
saturation region: $V_{GST}(t-\tau) > 0$ and $V_{DS} > 3/\alpha$

Then

$$I_{DS} = Area\, I_{DS0}/(1 + \delta I_{DS0} V_{DS}) \quad (7.67)$$

$$I_{DS0} = \begin{cases} 0 & \text{cutoff region} \\ \dfrac{\beta}{1+UV_{GST}} V^Q_{GST}(t-\tau) F_d(\alpha V_{DS}) & \text{linear and saturation regions} \end{cases} \quad (7.68)$$

$$F_d(\alpha V_{DS}) = \frac{x}{\sqrt{1+x^2}} \quad (7.69)$$

$$V_G = QV_{ST} \ln\left[exp\left(V_{GST}/(QV_{ST}) + 1\right)\right] \quad (7.70)$$

$$V_{ST} = N_{ST}(kT/q) \quad (7.71)$$

$$N_{ST} = N_G + N_D V_{DS} \quad (7.72)$$

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the MESFET I/V characteristics are evaluated as in the normal mode (7.92) but with the drain and source subscripts exchanged, and V_{GD} used as the controlling voltage instead of V_{GS}.

The relationships of the parameters describing the I/V characteristics of the LEVEL -1 model are summarized in figure 7.15.

LEVEL -1 (TOM-2 Model) Capacitances

These are evaluated the same way the LEVEL 3 capacitanceaas described on 135 are evaluated.

LEVEL -1 (TOM2 Model)

KEYWORD PARAMETERS	
ALPHA	α
BETA	β
DELTA	δ
GAMMA	γ
LAMBDA	λ
U	μ
VTO	V_{T0}

$+$

GEOMETRY PARAMETER
$Area$

$\rightarrow$

DEVICE PARAMETERS
$I_{DS} = f(Area, \alpha, \beta, \delta, \gamma, \lambda, UV_{T0})$

Figure 7.12: LEVEL -1 (TOM-2 model) I/V dependencies.

The LEVEL -1 model is an implementation of the TOM model ("Triquint's Own Model) [24].

LEVEL -1 (TOM2 Model) I/V Characteristics

The LEVEL -1 current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \geq 0$)

The regions of operation are defined as follows with
$$V_{GST} = V_{GS} - V_P \tag{7.73}$$
and
$$V_P = V_{T0} - \gamma V_{DS} \tag{7.74}$$

cutoff region:	$V_{GST}(t-\tau) \leq 0$
linear region:	$V_{GST}(t-\tau) > 0$ and $V_{DS} \leq 3/\alpha$
saturation region:	$V_{GST}(t-\tau) > 0$ and $V_{DS} > 3/\alpha$

Then

$$I_{DS} = Area\, I_{DS0}/(1 + \delta I_{DS0} V_{DS}) \tag{7.75}$$

$$I_{DS0} = \begin{cases} 0 & \text{cutoff region} \\ V_G^Q(t-\tau) F_d(\alpha V_{DS}) & \text{linear and saturation regions} \end{cases} \tag{7.76}$$

$$F_d(x) = \frac{x}{\sqrt{1+x^2}} \tag{7.77}$$

$$V_G = QV_{ST} \ln\left[\exp\left(\frac{V_{GST}}{QV_{ST}}\right) + 1)\right] \tag{7.78}$$

$$V_{ST} = N_{ST}\left(\frac{kT}{q}\right) \tag{7.79}$$

$$N_{ST} = N_G + N_D V_{DS} \tag{7.80}$$

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the MESFET I/V characteristics are evaluated as in the normal mode (7.75) but with the drain and source subscripts exchanged, and V_{GD} used as the controlling voltage instead of V_{GS}.
The relationships of the parameters describing the I/V characteristics of the LEVEL -1 model are summarized in figure 7.13.

KEYWORD PARAMETERS	
ALPHA	α
BETA	β
DELTA	δ
GAMMA	γ
LAMBDA	λ
U	μ
VTO	V_{T0}

+

GEOMETRY PARAMETER
$Area$

→

DEVICE PARAMETERS
$I_{DS} = f(Area, \alpha, \beta, \delta, \gamma, \lambda, UV_{T0})$

Figure 7.13: LEVEL 3 (TOM model) I/V dependencies.

LEVEL -1 (TOM2 Model) Capacitances

The LEVEL -1 (TOM2 Model) capacitances are identical to that of the LEVEL 3 model. This capacitance model is based on the model proposed by Statz as used in the level 2 model. This is a charge conserving symmetrical capacitance model.

The drain-source capacitance $C'_{DS} = Area\, C_{DS}$ (7.81)

The gate-source capacitance

$$C'_{GS} = Area \left[C_{GS} F_1 F_2 \left(1 - \frac{V_{\rm NEW}}{V_{BI}} \right)^{-\frac{1}{2}} + C_{GD} F_3 \right] \tag{7.82}$$

The gate-drain capacitance is given by

$$C'_{GD} = Area \left[C_{GS} F_1 F_3 \left(1 - \frac{V_{\rm NEW}}{V_{BI}} \right)^{-\frac{1}{2}} + C_{GD} F_2 \right] \tag{7.83}$$

$$F_1 = \frac{1}{2} \left\{ 1 + \frac{V_{\rm EFF} - V_P}{\sqrt{(V_{\rm EFF} - V_{T0})^2 + V_\Delta^2}} \right\} \tag{7.84}$$

$$F_2 = \frac{1}{2} \left\{ 1 + \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.85}$$

$$F_3 = \frac{1}{2} \left\{ 1 - \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.86}$$

$$V_{\rm EFF} = \frac{1}{2} \left\{ V_{GS} + V_{GD} + \sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}} \right\} \tag{7.87}$$

$$V_{\rm NEW} = \begin{cases} A_1 & A_1 < V_{\rm MAX} \\ V_{\rm MAX} & A_1 \geq V_{\rm MAX} \end{cases} \tag{7.88}$$

and

$$A_1 = \tfrac{1}{2} \left[V_{\rm EFF} + V_P + \sqrt{(V_{\rm EFF} - V_P)^2 + V_\Delta^2} \right] \tag{7.89}$$

The capacitance parameter dependencies are summarized in figure 7.14.

KEYWORD PARAMETERS	
ALPHA	α
CGD	C_{GD}
CGS	C_{GS}
CDS	C_{DS}
VBI	V_{BI}
VTO	V_{T0}
LAMBDA	V_{T0}
VMAX	$V_{\rm MAX}$
VDELTA	V_{Δ}

\+

GEOMETRY PARAMETER
Area

$\rightarrow$

DEVICE PARAMETERS
$C'_{DS} = f(Area, C_{DS})$
$C'_{GD} = f(Area, C_{GD}, \alpha, B, F_C, V_{BI}, V_{T0})$
$C'_{GS} = f(Area, C_{GS}, \alpha, B, F_C, V_{BI}, V_{T0})$

Figure 7.14: LEVEL 3 (TOM model) capacitance dependencies.

LEVEL -1 (TOM-2 Model)

The LEVEL -1 model is an enhancement of the LEVEL 3 TOM model.

LEVEL -1 (TOM Model) I/V Characteristics

The LEVEL -1 current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \geq 0$)

The regions of operation are defined as follows with
$$V_{GST} = V_{GS} - V_P \tag{7.90}$$
and
$$V_P = V_{T0} - \gamma V_{DS} \tag{7.91}$$

cutoff region:	$V_{GST}(t-\tau) \leq 0$
linear region:	$V_{GST}(t-\tau) > 0$ and $V_{DS} \leq 3/\alpha$
saturation region:	$V_{GST}(t-\tau) > 0$ and $V_{DS} > 3/\alpha$

Then

$$I_{DS} = Area\, I_{DS0}/(1 + \delta I_{DS0} V_{DS}) \tag{7.92}$$

$$I_{DS0} = \begin{cases} 0 & \text{cutoff region} \\ \dfrac{\beta}{1 + UV_{GST}} V^Q_{GST}(t-\tau) F_d(\alpha V_{DS}) & \text{linear and saturation regions} \end{cases} \tag{7.93}$$

$$F_d(\alpha V_{DS}) = \frac{x}{\sqrt{1+x^2}} \tag{7.94}$$

$$V_G = QV_{ST} \ln\left[exp\left(V_{GST}/(QV_{ST}) + 1\right)\right] \tag{7.95}$$

$$V_{ST} = N_{ST}(kT/q) \tag{7.96}$$

$$N_{ST} = N_G + N_D V_{DS} \tag{7.97}$$

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the MESFET I/V characteristics are evaluated as in the normal mode (7.92) but with the drain and source subscripts exchanged, and V_{GD} used as the controlling voltage instead of V_{GS}.

The relationships of the parameters describing the I/V characteristics of the LEVEL -1 model are summarized in figure 7.15.

LEVEL -1 (TOM-2 Model) Capacitances

KEYWORD PARAMETERS	
ALPHA	α
BETA	β
DELTA	δ
GAMMA	γ
LAMBDA	λ
U	μ
VTO	V_{T0}

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$I_{DS} = f(Area, \alpha, \beta, \delta, \gamma, \lambda, UV_{T0})$

Figure 7.15: LEVEL -1 (TOM-2 model) I/V dependencies.

These are evaluated the same way the LEVEL 3 capacitanceaas described on 135 are evaluated.

LEVEL 4 (Curtice Cubic Model)

LEVEL *4 (Curtice Cubic Model) I/V Characteristics*

The LEVEL 4 current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff,linear or saturation) of the current (V_{DS}, V_{GS}) operating point. Curtice [11] proposed two DC current models: a quadratic channel current model and a cubic channel current model. The quadratic channel model is implemented as the LEVEL 4 model.

Normal Mode: ($V_{DS} \geq 0$)

The regions of operation are defined as follows with with $V_{GST} = V_{GS} - V_{T0}$ (7.98)

cutoff region: $V_{GS}(t-\tau) < V_{T0}$

linear and saturation regions: $V_{GS}(t-\tau) > V_{T0}$

Then thedrain source current is given by

$$I_{DS} = \begin{cases} 0 & \text{cutoff region} \\ Area\left(A_0 + A_1 V_x + A_2 V_x^2 + A_3 V_x^3\right)\tanh\left(\gamma V_{DS}\right) & \text{linear and saturation regions} \end{cases} \quad (7.99)$$

$$V_x = V_{GS}(t-\tau)\left[1 + \beta(V_{DS0} - V_{DS})\right] \quad (7.100)$$

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the MESFET I/V characteristics are evaluated as in the normal mode (7.99) but with the drain and source subscripts exchanged, and V_{GD} is the controlling voltage instead of V_{GS}.

The relationships of the parameters describing the I/V characteristics for the LEVEL 4 model are summarized in figure 7.16.

LEVEL 4 (Curtice Cubic Model) Capacitances

KEYWORD PARAMETERS	
`A0`	A_0
`A1`	A_1
`A2`	A_2
`A3`	A_3
`BETA`	β
`GAMMA`	γ
`T, TAU`	τ
`VTO`	V_{T0}

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$I_{DS} = f(Area,$ $A_0, A_1, A_2, A_3, \beta, \gamma, V_{T0})$

Figure 7.16: LEVEL 4 (Curtice cubic model) I/V dependencies.

The drain-source capacitance $C'_{DS} = Area\, C_{DS}$ (7.101)
the gate-source and gate-drain capacitances are

$$C'_{GS} = \begin{cases} Area\, C_{GS} \left(1 - \frac{V_{GS}}{V_{BI}}\right)^{-M_{GS}} & V_{GS} \le F_C V_{BI} \\ Area\, C_{GS}\, (1 - F_C)^{-(1+M_{GS})} \left[1 - F_C(1 + M_{GS}) + M_{GS}\frac{V_{GS}}{V_{BI}}\right] & V_{GS} > F_C V_{BI} \end{cases} \quad (7.102)$$

$$C_{GD} = \begin{cases} Area\, C_{GD} \left(1 - \frac{V_{GD}}{V_{BI}}\right)^{-M_{GD}} & V_{GD} \le F_C V_{BI} \\ Area\, C_{GD}\, (1 - F_C)^{-(1+M_{GD})} \left[1 - F_C(1 + M_{GD}) + M_{GD}\frac{V_{GD}}{V_{BI}}\right] & V_{GS} > F_C V_{BI} \end{cases} \quad (7.103)$$

The `LEVEL` 4 capacitance parameter dependencies are summarized in figure 7.17.

KEYWORD PARAMETERS	
`CGD`	C_{GD}
`CGS`	C_{GS}
`CDS`	C_{DS}
`FC`	F_C
`VBI`	V_{BI}
`MGS`	M_{GS}
`MGD`	M_{GD}

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$C'_{DS} = f(Area, C_{DS})$
$C'_{GD} =$ $f(Area, C_{GD}, F_C, V_{BI}, M_{GD})$
$C'_{GS} =$ $f(Area, C_{GS}, F_C, V_{BI}, M_{GS})$

Figure 7.17: LEVEL 4 (Curtice Cubic model) capacitance dependencies.

LEVEL 5 (Materka-Kacprzak Model)

The parameter keywords of the Materka-Kacprzak model are given in table 7.3.

Table 7.3: GASFET level 5 (Materka-Kacprzak) model keywords. SomeVersionsOfSpice only. Parameters that are NOT USED are reserved for future expansion.

Name	Description	Units	Default
AC10	temperature coefficient of C10 (A_{C10})	$°C^{-1}$	0
ACF0	temperature coefficient of CF0 (A_{CF0})	$°C^{-1}$	0
AF	flicker noise exponent (A_F)	-	1
AE	temperature coefficient of E (A_E)	V^{-1}	0
AFAB	slope factor of breakdown current (≥ 0) (A_{FAB})	$°C^{-1}$	0
AFAG	slope factor of gate conduction current (A_{FAG})	V^{-1}	38.696
AGAM	temperature coefficient of GAMA (A_γ)	$°C^{-1}$	0
AIDS	linear temperature coefficient of IDSS (A_λ)	%/°C	0
AKE	temperature coefficient of KE (A_{KE})	$°C^{-1}$	0
AKG	temperature coefficient of KG (A_{KG})	$°C^{-1}$	0
ARD	alternative keyword for TRD1 (A_{RD})	$°C^{-1}$	0
ARG	alternative keyword for TRG1 (A_{RG})	$°C^{-1}$	0
ARS	alternative keyword for TRS1 (A_{RS})	$°C^{-1}$	0
AR10	temperature coefficient of R10 (A_{R10})	$°C^{-1}$	0
ASL	temperature coefficient of SL (A_{SL})	$°C^{-1}$	0
ASS	temperature coefficient of SS (A_{SS})	$°C^{-1}$	0
AT	temperature coefficient of T (A_T)	$°C^{-1}$	0
AVBC	linear temperature coefficient of VDC ($A_{VBC)}$)	$°C^{-1}$	0
AVP0	linear temperature coefficient of VP0 ($A_{VP0)}$)	$°C^{-1}$	0
C10	gate-source Schottky barrier capacitance for $V_{GS} = 0$. (C_{10})	F	0
CF0	gate-drain feedback capacitacne for $V_{GD} = 0$. (C_{F0})	F	0
C1S	constant parasitic component of gate-source capacitance (C_{1S})	F	0
CDE	drain-source electrode capacitance NOT USED (C_{DE})	F	0
CDGE	drain-gate electrode capacitance NOT USED (C_{DGE})	F	0
CDS	drain-source capacitance (C_{DS})	F	0
CDSD	low frequency trapping capacitance (NOT USED) (C_{DSD})	F	0
CGE	gate-source electrode capacitance NOT USED (C_{DE})	F	0
CGS0	zero-bias gate-source p-n capacitance in Raytheon capacitance model. Used if CLVL = 2 (C_{GS})	F	0
CGD0	zero-bias gate-drain p-n capacitance in Raytheon capacitance model. Used if CLVL = 2 (C_{GS})	F	0
CLVL	capacitance model flag CLVLV = 1 use Materka-Kacprzak capacitance model CLVLV = 2 use Raytheon capacitance model (C_{LVL})	-	1
E	constant part of drain current power (E)	-	2
FCC	(F_{CC})	-	0.8
GAMA	Voltage slope parameter of pinch-off voltage (γ)	V^{-1}	0
IB0	current parameter of gate-drain breakdown source (≥ 0) (I_{B0})	A	0
IDSS	frain saturation current for $V_{GS} = 0$ (I_{DSS})	A	0.1

Continued on next page

Table 7.3: GASFET level 5 (Materka-Kacprzak) model keywords. SomeVersionsOfSpice only. Parameters that are NOT USED are reserved for future expansion.

Name	Description	Units	Default
IG0	saturation current of gate-source schottky barrier (I_{G0})	A	0
K1	slope parameter of gate-source capacitance (K_1)	V^{-1}	1.25
KE	dependence of drain current power on V_{GS} (K_E)	V^{-1}	0
KF	slope parameter of gate-drain feedback capacitance (K_F)	-	0
KFL	flicker noise coefficient (K_{FL})	-	0
KG	linearregion drain current V_{GS} dependence (K_G)	V^{-1}	0
KR	slope factor of intrinsic channel resistance (K_R)	V^{-1}	0
LEVEL	model index must be 5	-	1
M	gate p-n grading coefficient (M)	-	0.5
MGS	gate-source p-n grading coefficient (M_{GS})	-	M
MGD	gate-drain p-n grading coefficient (M_{GS})	-	M
R10	intrinsic channel resistance for $V_{GS} = 0$. (R_{10})	Ω	∞
RD	drain resistance (R_D)	Ω	0
RDSD	channel trapping resistance NOT USED (R_{DSD})	Ω	∞
RG	gate resistance (R_G)	Ω	0
RS	source resistance (R_S)	Ω	0
SL	slope of $V_{GS} = 0$ drain current, linear region (S_L)	S	0.15
SS	slope of $V_{GS} = 0$ drain current, saturated region (S_S)	S	0
T	channel transit time delay (τ)	s	0
TJ	junction temperature (version: SomeVersionsOfSpice) (T_J)	K	298
TNOM	model reference temperature (> 0) (version: SomeVersionsOfSpice) (T_{NOM})	K	298
TRG1	temperature coefficient of RG (A_{RG})	$°C^{-1}$	0
TRD1	temperature coefficient of RG (A_{RD})	$°C^{-1}$	0
TRS1	temperature coefficient of RG (A_{RS})	$°C^{-1}$	0
VBC	breakdown voltage (≥ 0 (V_{BC})	V	∞
VBI	gate p-n potential in Raytheon capacitance model. Used if CLVL = 2. ($V_{\rm BI}$)	V	1
VP0	(VP-zero) pinch-off voltage for $V_{DS} = 0$ (V_{P0})	V	-2.5

Temperature Dependence

The Materka-Kacprzak temperature effects are as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K).

$$I_{DSS}(T) = I_{DSS}(T_{\rm NOM}\,(1 + A_{IDSS}(T - T_{\rm NOM})) \tag{7.104}$$
$$C_{10}(T) = C_{10}(T_{\rm NOM}\,(1 + A_{C10}(T - T_{\rm NOM})) \tag{7.105}$$
$$C_{F0}(T) = C_{F0}(T_{\rm NOM}\,(1 + A_{CF0}(T - T_{\rm NOM})) \tag{7.106}$$
$$E(T) = E(T_{\rm NOM}\,(1 + A_{E}(T - T_{\rm NOM})) \tag{7.107}$$
$$K_E(T) = K_E(T_{\rm NOM}\,(1 + A_{KE}(T - T_{\rm NOM})) \tag{7.108}$$
$$K_G(T) = K_G(T_{\rm NOM}\,(1 + A_{KG}(T - T_{\rm NOM})) \tag{7.109}$$
$$\gamma(T) = \gamma(T_{\rm NOM}\,(1 + A_{\gamma}(T - T_{\rm NOM})) \tag{7.110}$$
$$R_{10}(T) = R_{10}(T_{\rm NOM}\,(1 + A_{R10}(T - T_{\rm NOM})) \tag{7.111}$$
$$R_G(T) = R_G(T_{\rm NOM}\,(1 + A_{RG}(T - T_{\rm NOM})) \tag{7.112}$$
$$R_D(T) = R_D(T_{\rm NOM}\,(1 + A_{RD}(T - T_{\rm NOM})) \tag{7.113}$$
$$R_S(T) = R_S(T_{\rm NOM}\,(1 + A_{RS}(T - T_{\rm NOM})) \tag{7.114}$$
$$S_L(T) = S_L(T_{\rm NOM}\,(1 + A_{SL}(T - T_{\rm NOM})) \tag{7.115}$$
$$S_S(T) = S_S(T_{\rm NOM}\,(1 + A_{SS}(T - T_{\rm NOM})) \tag{7.116}$$
$$\tau(T) = \tau(T_{\rm NOM}\,(1 + A_{\tau}(T - T_{\rm NOM})) \tag{7.117}$$
$$V_{BC}(T) = V_{BC}(T_{\rm NOM}\,(1 + A_{VBC}(T - T_{\rm NOM})) \tag{7.118}$$
$$V_{P0}(T) = V_{P0}(T_{\rm NOM}\,(1 + A_{VP0}(T - T_{\rm NOM})) \tag{7.119}$$

Parasitic Resistances

The resistive parasitics R'_S, and R'_D are calculated from the sheet resistivities `RS` ($= R_S$) and `RD` ($= R_D$), and the *Area* specified on the element line. `RG` ($= R_G$) is used as supplied.

$$R'_D = R_D/Area \tag{7.120}$$
$$R'_G = \begin{cases} R_G & \text{PSpice} \\ R_G/Area & \text{SomeVersionsOfSpice} \end{cases} \tag{7.121}$$
$$R'_G = R_G/Area \tag{7.122}$$
$$R'_S = R_S/Area \tag{7.123}$$

The parasitic resistance parameter dependencies are summarized in figure 7.18.

KEYWORD PARAMETERS	
`RD`	R_D
`RG`	R_G
`RS`	R_S

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$R'_D = f(Area, R_D)$
$R'_G = f(Area, R_G$
$R'_S = f(Area, R_S)$

Figure 7.18: MESFET parasitic resistance parameter relationships.

`LEVEL` *5 (Materka-Kacprzak Model) I/V Characteristics*

The `LEVEL 5` current/voltage characteristics are analytic except for the channel resistance R_I determination.

$$I_{DS} = Area I_{DSS}\left[1+S_S\frac{V_{DS}}{I_{DSS}}\right]\left[1-\frac{V_{GS}(t-\tau)}{V_{P0}+\gamma V_{DS}}\right]^{(E+K_E V_{GS}(t-\tau))} \times\tanh\left[\frac{S_L V_{DS}}{I_{DSS}(1-K_G V_{GS}(t-\tau))}\right] \quad (7.124)$$

$$I_{GS} = Area I_{G0}\left[e^{A_{FAG}V_{GS}}-1\right]-I_{B0}\left[e^{-A_{FAB}(V_{GS}+V_{BC}}\right] \quad (7.125)$$

$$I_{GD} = Area I_{G0}\left[e^{A_{FAG}V_{GD}}-1\right]-I_{B0}\left[e^{-A_{FAB}(V_{GD}+V_{BC}}\right] \quad (7.126)$$

$$R_I = \begin{cases} R_{10}(1-K_R V_{GS})/Area & K_R V_{GS} < 1.0 \\ 0 & K_R V_{GS} \geq 1.0 \end{cases} \quad (7.127)$$

The relationships of the parameters describing the I/V characteristics for the `LEVEL` 5 model are summarized in figure **??**.

KEYWORD PARAMETERS	
`AFAG`	A_{FAG}
`AFAB`	A_{FAB}
`GAMA`	γ
`IDSS`	I_{DSS}
`IBO`	I_{B0}
`IGO`	I_{G0}
`KG`	K_G
`KE`	K_E
`KR`	K_R
`R1O`	R_{10}
`SS`	S_S
`SL`	S_L
`T`	τ
`VBC`	V_{BC}
`VPO`	V_{P0}

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$I_{DS} = f(Area, E, \gamma, I_{DSS}, K_E, K_G, S_L, S_S, \tau, V_{P0})$
$I_{GS} = f(Area, A_{FAB}, A_{FAG}, I_{B0}, I_{G0}, V_{BC})$
$I_{GD} = f(Area, A_{FAB}, A_{FAG}, I_{B0}, I_{G0}, V_{BC})$
$R_I = f(Area, K_R, R_{10}, V_{GS})$

Figure 7.19: LEVEL 5 (Materka-Kacprzak model) I/V dependencies.

LEVEL 5 (Materka-Kacprzak Model) Capacitances

Two capacitance models are available depending on the value of the C_{LVL} (`CLVL`) parameter. With $C_{LVL} = 1$ (default) the standard Materka capacitance model described below is used. With $C_{LVL} = 0$ the Raytheon capacitance model described on page 129 is used. The Materka-Kacprzak capacitances are

$$C'_{DS} = Area\, C_{DS} \quad (7.128)$$

$$C'_{GS} = \begin{cases} Area\left[C_{10}(1-K_1 V_{GS})^{M_{GS}}+C_{1S}\right] & K_1 V_{GS} < F_{CC} \\ Area\left[C_{10}(1-F_{CC})^{M_{GS}}+C_{1S}\right] & K_1 V_{GS} \geq F_{CC} \end{cases} \quad (7.129)$$

$$C'_{GD} = \begin{cases} Area\left[C_{F0}(1-K_1 V_1)^{M_{GD}}\right] & K_1 V_1 < F_{CC} \\ Area\left[C_{F0}(1-F_{CC})^{M_{GD}}\right] & K_1 V_1 \geq F_{CC} \end{cases} \quad (7.130)$$

The `LEVEL` 5 capacitance parameter dependencies are summarized in figure 7.20.

KEYWORD PARAMETERS	
CGD	C_{GD}
CGS	C_{GS}
CDS	C_{DS}
FC	F_C
VBI	V_{BI}
M	M

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$C'_{DS} = f(Area, C_{DS})$
$C'_{GD} = f(Area, C_{10}, C_{1S}, F_{CC}, K_1, V_{GS}, M_{GS})$
$C'_{GS} = f(Area, C_{F0}, C_{1S}, F_{CC}, K_1, V_1, V_{GS}, M_{GS})$

Figure 7.20: LEVEL 5 (Materka-Kacprzak model) capacitance dependencies.

LEVEL 6 (Angelov Model)

The parameter keywords of the Materka-Kacprzak model are given in table 7.4.

Table 7.4: GASFET model 6 (Angelov) keywords. Parameters that are NOT USED are reserved for future expansion.

Name	Description		Units	Default
AF	flicker noise exponent	(A_F)	-	1
ACGS	linear temperature coefficient of CGS	(A_{CGS})	$°C^{-1}$	0
ACGD	linear temperature coefficient of CGD	(A_{CGD})	$°C^{-1}$	0
ALFA	saturation voltage parameter (≥ 0)	(α)	V^{-1}	1.5
ARI	linear temperature coefficient of RI	(A_{RI})	$°C^{-1}$	0
ARG	linear temperature coefficient of RG	(A_{RG})	$°C^{-1}$	0
ARD	linear temperature coefficient of RD	(A_{RD})	$°C^{-1}$	0
ARS	linear temperature coefficient of RS	(A_{RS})	$°C^{-1}$	0
B1	unsaturated coefficient of P1	(B_1)	-	0
B2	drain voltage slope parameter for B1	(B_2)	-	3.0
BCGD	quadratic temperature coefficient of CGD	(B_{CGD})	$°C^{-2}$	0
BCGS	quadratic temperature coefficient of CGS	(B_{CGS})	$°C^{-2}$	0
BRI	quadratic temperature coefficient of RI	(B_{RI})	$°C^{-2}$	0
BRG	quadratic temperature coefficient of RG	(B_{RG})	$°C^{-2}$	0
BRS	quadratic temperature coefficient of RS	(B_{RS})	$°C^{-2}$	0
BRD	quadratic temperature coefficient of RD	(B_{RD})	$°C^{-2}$	0
CGD0	gate-source capacitance at $\Psi_3 = \Psi_4 = 0$ (¿= 0)	(C_{GD0})	F	0
CGS0	gate-source capacitance at $\Psi_1 = \Psi_2 = 0$ (¿= 0)	(C_{GS0})	F	0
EG	barrier height at 0 K	(E_G)	V	0.8
GAMA	voltage slope parameter for pinch-off (NO LONGER USED)		V^{-1}	0
IB0	breakdown saturation current (≥ 0)	(I_{B0})	A	0
IPK	drain current at peak g_m (≥ 0)	(I_{PK})	A	0.1
IS	diode saturation current (≥ 0)	(I_S)	A	0
LAMB	slope of the drain characteristic	(λ)	V^{-1}	0

Continued on next page

Table 7.4: GASFET model 6 (Angelov) keywords. Parameters that are NOT USED are reserved for future expansion.

Name	Description		Units	Default
LEVEL	model index	must be 6	-	1
N	diode ideality factor (> 0	(N)	-	1
NR	breakdown ideality factor (> 0	(N_R)	-	10
P2	polynomial coefficient of channel current	(P_2)	V^{-2}	0
P3	polynomial coefficient of channel current	(P_3)	V^{-3}	0
P4	polynomial coefficient of channel current	(P_4)	V^{-4}	0
P5	polynomial coefficient of channel current	(P_5)	V^{-5}	0
P6	polynomial coefficient of channel current	(P_6)	V^{-6}	0
P7	polynomial coefficient of channel current	(P_7)	V^{-7}	0
P8	polynomial coefficient of channel current	(P_7)	V^{-8}	0
P10	polynomial coefficient of g-s capacitance	(P_{10})	-	0
P11	polynomial coefficient of g-s capacitance	(P_{11})	V^{-1}	0
P12	polynomial coefficient of g-s capacitance	(P_{12})	V^{-2}	0
P13	polynomial coefficient of g-s capacitance	(P_{13})	V^{-3}	0
P14	polynomial coefficient of g-s capacitance	(P_{14})	V^{-4}	0
P20	polynomial coefficient of g-s capacitance	(P_{20})	-	0
P21	polynomial coefficient of g-s capacitance	(P_{21})	V^{-1}	0
P22	polynomial coefficient of g-s capacitance	(P_{22})	V^{-2}	0
P23	polynomial coefficient of g-s capacitance	(P_{23})	V^{-3}	0
P24	polynomial coefficient of g-s capacitance	(P_{24})	V^{-4}	0
P30	polynomial coefficient of g-d capacitance	(P_{30})	-	0
P31	polynomial coefficient of g-d capacitance	(P_{31})	V^{-1}	0
P32	polynomial coefficient of g-d capacitance	(P_{32})	V^{-2}	0
P33	polynomial coefficient of g-d capacitance	(P_{33})	V^{-3}	0
P34	polynomial coefficient of g-d capacitance	(P_{34})	V^{-4}	0
P40	polynomial coefficient of g-d capacitance	(P_{40})	-	0
P41	polynomial coefficient of g-d capacitance	(P_{41})	V^{-1}	0
P42	polynomial coefficient of g-d capacitance	(P_{42})	V^{-2}	0
P43	polynomial coefficient of g-d capacitance	(P_{43})	V^{-3}	0
P44	polynomial coefficient of g-d capacitance	(P_{44})	V^{-4}	0
P1CC	polynomial coefficient of g-d capacitance	(P_{1CC})		0
PSAT	polynomial coefficient of channel current PSAT	(P_1)	V^{-1}	1.3
RI	channel resistacne (≥ 0) NOT USED	(R_I)		0
T	channel time delay (≥ 0)	(τ)	s	0
TJ	junction temperature	(T_J)	K	298
TM	I_{DS} linear temperature coefficient	(T_M)		0
TME	I_{DS} power law temperature coefficient	(T_{ME})		0
TNOM	model reference temperature (> 0)	(T_{NOM})	K	298
VBD	breakdown voltage	(V_{BD})		0
VPK0	gate-source voltage for unsaturated peak g_m	(V_{PK0})	V	-0.5
VPKS	gate-source voltage for peak g_m	(V_{PKS})	V	0
XTI	saturation current temperature exponent	(X_{TI})		2

Temperature Dependence

Temperature effects are incorporated as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K) and the thermal voltage $V_{\rm TH} = kT/q$.

$$\alpha(T) = \alpha(T_{\rm NOM})\left(1 + A_\alpha(T - T_{\rm NOM})\right) \quad (7.131)$$
$$C_{GS0}(T) = C_{GS0}(T_{\rm NOM})\left(1 + A_{CGS}(T - T_{\rm NOM}) + B_{CGS}(T - T_{\rm NOM})^2\right) \quad (7.132)$$
$$C_{GD0}(T) = C_{GD0}(T_{\rm NOM})\left(1 + A_{CGD}(T - T_{\rm NOM}) + B_{CGD}(T - T_{\rm NOM})^2\right) \quad (7.133)$$
$$E_G(T) = E_G(0) - F_{\rm GAP1} 4T^2/\left(T + F_{\rm GAP2}\right) \quad (7.134)$$
$$\gamma(T) = \gamma(T_{\rm NOM})\left(1 + A_\gamma(T - T_{\rm NOM})\right) \quad (7.135)$$
$$I_{PK}(T) = I_{PK}(T_{\rm NOM})\left(1 + A_{IPK}(T - T_{\rm NOM})\right) \quad (7.136)$$
$$R_I(T) = R_I(T_{\rm NOM})\left(1 + A_{RI}(T - T_{\rm NOM}) + B_{RI}(T - T_{\rm NOM})^2\right) \quad (7.137)$$
$$R_G(T) = R_G(T_{\rm NOM})\left(1 + A_{RG}(T - T_{\rm NOM}) + B_{RG}(T - T_{\rm NOM})^2\right) \quad (7.138)$$
$$R_D(T) = R_D(T_{\rm NOM})\left(1 + A_{RD}(T - T_{\rm NOM}) + B_{RD}(T - T_{\rm NOM})^2\right) \quad (7.139)$$
$$R_S(T) = R_S(T_{\rm NOM})\left(1 + A_{RS}(T - T_{\rm NOM}) + B_{RS}(T - T_{\rm NOM})^2\right) \quad (7.140)$$
$$V_{PK}(T) = V_{PK}(T_{\rm NOM})\left(1 + A_{VPK}(T - T_{\rm NOM})\right) \quad (7.141)$$
$$V_{BD}(T) = V_{BD}(T_{\rm NOM})\left(1 + A_{VBD}(T - T_{\rm NOM})\right) \quad (7.142)$$
$$\texttt{TM}I_{DS}\texttt{TME}???? \quad (7.143)$$

Parasitic Resistances

The resistive parasitics R'_S, and R'_D are calculated from the sheet resistivities `RS` ($= R_S$) and `RD` ($= R_D$), and the *Area* specified on the element line. `RG` ($= R_G$) is used as supplied.

$$R'_D = R_D/Area \quad (7.144)$$
$$R'_G = R_G/Area \quad (7.145)$$
$$R'_S = R_S/Area \quad (7.146)$$

The parasitic resistance parameter dependencies are summarized in Figure 7.21.

KEYWORD PARAMETERS	
RD	R_D
RG	R_G
RS	R_S

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$R'_D = f(Area, R_D)$
$R'_G = f(Area, R_G)$
$R'_S = f(Area, R_S)$

Figure 7.21: MESFET parasitic resistance parameter relationships.

`LEVEL` *6 (Angelov Model) I/V Characteristics*

The `LEVEL 6` current/voltage characteristics are analytic:

$$
\begin{aligned}
I_{DS} &= I_{PK}[1+\tanh(\Psi)][1+\lambda V_{DS}]\tanh(\alpha V_{DS}) && (7.147)\\
\Psi &= P_1\left(V_{GS}-V_{PK}\right)+P_2\left(V_{GS}-V_{PK}\right)^2+P_3\left(V_{GS}-V_{PK}\right)^3 \\
&\quad +P_4\left(V_{GS}-V_{PK}\right)^4+P_5\left(V_{GS}-V_{PK}\right)^5+P_6\left(V_{GS}-V_{PK}\right)^6 \\
&\quad +P_7\left(V_{GS}-V_{PK}\right)^7+P_8\left(V_{GS}-V_{PK}\right)^8 && (7.148)\\
P_1 &= P_{\rm SAT}\left[1+B_1/\left(\cosh^2(B_2V_{DS})\right)\right] && (7.149)\\
V_{PK} &= V_{PK0}+(V_{PKS}-V_{PK0})\tanh(\alpha V_{DS}) && (7.150)\\
I_{GS} &= I_S\left[e^{V_{GS}/(NV_{\rm TH})}-1\right]-I_{B0}\left[e^{-(V_{GS}+V_{BD})/(N_RV_{\rm TH})}\right] && (7.151)\\
I_{GD} &= I_S\left[e^{V_{GD}/(NV_{\rm TH})}-1\right]-I_{B0}\left[e^{-(V_{GD}+V_{BD})/(N_RV_{\rm TH})}\right] && (7.152)\\
R_I &= R_I'/Area && (7.153)
\end{aligned}
$$

The relationships of the parameters describing the I/V characteristics for the `LEVEL` 6 model are summarized in figure **??**.

KEYWORD PARAMETERS	
`IPK`	I_{PK}
`LAMB`	λ
`ALFA`	α
`P1`	P_1
`P2`	P_2
`P3`	P_3
`P4`	P_4
`P5`	P_5
`P6`	P_6
`P7`	P_7
`P8`	P_8
`IS`	I_S
`N`	N
`IB0`	I_{B0}
`NR`	N_R
`VBD`	V_{BD}
`VPK0`	V_{PK0}
`VPKS`	V_{PKS}

+

GEOMETRY PARAMETER
$Area$

→

DEVICE PARAMETERS
$I_{DS} = f(Area, I_{PK}, \lambda, \alpha,$ $P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8,$ $V_{PK0}, V_{PKS},)$
$I_{GS} =$ $f(Area, I_S, N, I_{B0}, V_{BD}, N_R)$
$I_{GS} =$ $f(Area, I_S, N, I_{B0}, V_{BD}, N_R)$

Figure 7.22: LEVEL 6 (Angelov model) I/V dependencies.

LEVEL 6 (Angelov Model) Capacitances

The Angelov capacitances are

$$C'_{DS} = Area\, C_{DS} \tag{7.154}$$
$$C'_{GS} = AreaC_{GS0}\left[1+\tanh(\Psi_1)\right]\left[1+\tanh(\Psi_2)\right] \tag{7.155}$$
$$\Psi_1 = P_{10}+P_{11}V_{GS}+P_{12}V_{GS}^2+P_{13}V_{GS}^3+P_{14}V_{GS}^4 \tag{7.156}$$
$$\Psi_2 = P_{20}+P_{11}V_{DS}+P_{22}V_{DS}^2+P_{23}V_{DS}^3+P_{24}V_{DS}^4 \tag{7.157}$$
$$C'_{GD} = AreaC_{GD0}\left[1+\tanh(\Psi_3)\right]\left[1-\tanh(\Psi_4)\right] \tag{7.158}$$
$$\Psi_3 = P_{30}+P_{31}V_{GS}+P_{32}V_{GS}^2+P_{33}V_{GS}^3+P_{34}V_{GS}^4 \tag{7.159}$$
$$\Psi_4 = P_{40}+\left(P_{41}+P_{1CC}V_GS\right)V_{DS}+P_{42}V_{DS}^2+P_{43}V_{DS}^3+P_{44}V_{DS}^4 \tag{7.160}$$

The `LEVEL` 6 capacitance parameter dependencies are summarized in figure 7.23.

KEYWORD PARAMETERS	
`CGD`	C_{GD}
`CGS`	C_{GS}
`CDS`	C_{DS}
`FC`	F_C
`VBI`	V_{BI}
`M`	M

+

GEOMETRY PARAMETER
Area

→

DEVICE PARAMETERS
$C'_{DS} = f(Area, C_{DS})$
$C'_{GD} = f(Area, C_{10}, C_{1S}, F_{CC}, K_1, V_{GS}, M_{GS})$
$C'_{GS} = f(Area, C_{F0}, C_{1S}, F_{CC}, K_1, V_1, V_{GS}, M_{GS})$

Figure 7.23: LEVEL 6 (Angelov model) capacitance dependencies.

AC Analysis The AC analysis uses the model of figure 7.24 with the capacitor values evaluated at the `DC`

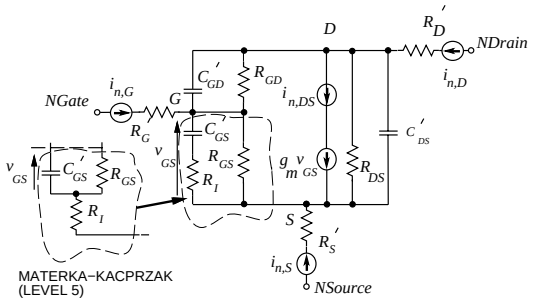


Figure 7.24: Small signal GASFET model showing noise sources $I_{n,G}$, $I_{n,D}$, $I_{n,S}$, and $I_{n,DS}$ used in the noise analysis. R_I is not used in PSpice.

operating point with

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \qquad R_{GD} = \frac{\partial I_{GD}}{\partial V_{GD}} \qquad R_{GS} = \frac{\partial I_{GS}}{\partial V_{GS}} \qquad R_{DS} = \frac{\partial I_{DS}}{\partial V_{DS}} \tag{7.161}$$

Noise Analysis

The MESFET noise model, see figure 7.24, accounts for thermal noise generated in the parasitic resistamces and shot and flicker noise generated in the drain source current generator. The rms (root-mean-square) values of thermal noise current generators shunting the three parasitic resistance R_D, R_G and R_S are

$$I_{n,D} = \sqrt{4kT/R'_D}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.162}$$

$$I_{n,G} = \sqrt{4kT/R'_G}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.163}$$

$$I_{n,S} = \sqrt{4kT/R'_S}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.164}$$

Shot and flicker noise are modeled by a noise current generator in series with the drain-source current generator I_{DS}. The rms value of this noise generator is given by

$$I_{n,DS} = \sqrt{I^2_{\mathrm{SHOT},DS} + I^2_{\mathrm{FLICKER},DS}} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.165}$$

$$I_{\mathrm{SHOT},DS} = \sqrt{4kTg_m\frac{2}{3}} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.166}$$

$$I_{\mathrm{FLICKER},DS} = \sqrt{\frac{K_F I_{DS}^{A_F}}{f}} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.167}$$

where f is the analysis frequency.

C Capacitor

Form

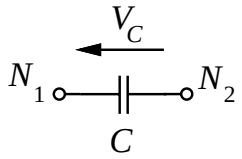


Figure 7.25: C — capacitor element.

C*name* N_1 N_2 *CapacitorValue* [IC=V_C] [L= *Length*] [W= *Width*]

SPICE3*Form*

C*name* N_1 N_2 [*ModelName*] *CapacitorValue* [IC=V_C] [L= *Length*] [W= *Width*]

PSPICE*Form*

C*name* N_1 N_2 [*ModelName*] *CapacitorValue* [IC=V_C]

N_1 is the positive element node,

N_2 is the negative element node, and

ModelName is the optional model name.

CapacitorValue is the capacitance. This is modified if *ModelName* is specified.
(Units: F; Required; Symbol: $CapacitorValue$)

L is the length *Length* of the integrated capacitor.
(Units: m; Required in SPICE3 if *ModelName* specified; Symbol: L;)

W is the width *Length* of the integrated capacitor.
(Units: m; Optional; Default: Default width DEFW specified in model *ModelName*; Symbol: L;)

IC is the optional initial condition specification Using IC=V_C is intended for use with the UIC option on the .TRAN line, when a transient analysis is desired with an initial voltage V_C across the capacitor rather than the quiescent operating point. Specification of the transient initial conditions using the .IC statement (see page 66) is preferred and is more convenient.

Example

```
CAP1 1 GND 12.3PF
C1 node1 0 ((12.3 + 2.1)/2)
```

Model Type

CAP

CAP Model — SPICE3 Only — Capacitor Model

Form

`.MODEL` *ModelName* `CAP(` [[*keyword* = *value*] ...] `)`

Example

```
.MODEL SMALLCAP CAP( CJ CJSW DEFW10^-6 NARROW)
```

Model Keywords

Name	Description	Units	Default
CJ	junction bottom capacitance (C_J)	F/m^2	REQUIRED
CJSW	junction sidewall capacitance ($C_{J,\mathrm{SW}}$)	F/m	REQUIRED
DEFW	default device width (W_{DEF})	meters	10^{-6}
NARROW	narrowing due to side etching (X_{NARROW})	meters	0

The SPICE3 capacitance model is a process model for a monolithicly fabricated capacitor enabling the capacitance to be determined from geometric information. If the parameter `W` is not specified on the element line then $Width$ defaults to `DEFW` = W_{DEF}. The effective dimensions are reduced by etching so that the effective length of the capacitor is

$$L_{\mathrm{EFF}} = (Length - X_{\mathrm{NARROW}}) \tag{7.168}$$

and the effective width is

$$W_{\mathrm{EFF}} = (Width - X_{\mathrm{NARROW}}) \tag{7.169}$$

The new value of the capacitance

$$NewCapacitorValue = C_J L_{\mathrm{EFF}} W_{\mathrm{EFF}} + 2C_{J,\mathrm{SW}}(L_{\mathrm{EFF}} + W_{\mathrm{EFF}}) \tag{7.170}$$

CAP Model — PSPICE Only — Capacitor Model

Form

`.MODEL` *ModelName* `CAP(` [[*keyword* = *value*] ...] `)`

Example

```
.MODEL SMALLCAP GASFET(C=2.5 VC1=0.01 VC2=0.001 TC1=0.02 TC2=0.005)
```

Model Keywords

Name	Description	Units	Default
C	capacitance multiplier ($C_{\text{MULTIPLIER}}$)	-	1
VC1	linear voltage coefficient (V_{C1})	1/V	0
VC2	quadratic voltage coefficient (V_{C2})	1/V	0
TC1	linear temperature coefficient (T_{C1})	$1/°C$	0
TC2	quadratic temperature coefficient (T_{C1})	$1/°C$	0

The PSpice capacitance model is a nonlinear temperature dependent capacitor model. It is assumed that the model parameters were determined or measured at the nominal temperature T_{NOM} (default 27°C) specified in the most recent `.OPTIONS` statement preceeding the `.MODEL` statement.

If the CAP model is specified then a new capacitance is evaluated as

$$C = CapacitorValue\, C_{\text{MULTIPLIER}} \left[1 + V_{C1}V_C + V_{C2}V_C^2\right] \tag{7.171}$$

$$\times \left[1 + T_{C1}(T - T_{\text{NOM}}) + T_{C2}(T - T_{NOM})^2\right] \tag{7.172}$$

where V_C is the voltage across the capacitor as in figure 7.25 and T is the current temperature. If PSpice CAP model is not specified then the capacitance specified on the element line is used.

D Diode

Figure 7.26: D — diode element.

Form

D*name* n_1 n_2 *ModelName* [*Area*] [`OFF`] [`IC`=V_D]

PSPICE*Form*

D*name* n_1 n_2 *ModelName* [*Area*] [`OFF`]

n_1 is the positive (anode) diode node.

n_1 is the negative (cathode) diode node.

ModelName is the model name.

Area is an optional relative area factor.
(Units: none; Optional; Default: 1, Symbol: *Area*)

`OFF` indicates an (optional) starting condition on the device for `DC` operating point analysis. If specified the `DC` operating point is calculated with the terminal voltages set to zero. Once convergence is obtained, the program continues to iterate to obtain the exact value of the terminal voltages. The OFF option is used to enforce the solution to correspond to a desired state if the circuit has more than one stable state.

`IC` is the (optional) initial condition specification. Using `IC` = V_D is intended for use with the `UIC` option on the other than the quiescent operating point. Specification of the transient initial conditions using the `.IC` statement (see page 66) is preferred and is more convenient.

Example

```
DBRIDGE 2 10 DIODE1
DCLMP 3 7 DMOD 3.0 IC=0.2
```

Model Type

`DIODE`

DIODE Model Diode Model

Note

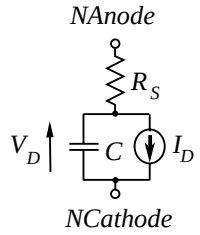


Figure 7.27: Schematic of diode element model.

1. In some SPICE implementations `XTI` is called `PT` and `VJ` is called `PB`.
2. It is assumed that the model parameters were determined or measured at the nominal temperature $T_{\rm NOM}$ (default $27°C$) specified in the most recent `.OPTIONS` statement preceeding the `.MODEL` statement.

The physical constants used in the model evaluation are

k	Boltzman's constant	$1.3806226\,10^{-23}$ J/K
q	electronic charge	$1.6021918\,10^{-19}$ C

Temperature Dependence

Temperature effects are incorporated as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K). The thermal voltage

$$V_{\rm TH} = \frac{kT}{q} \tag{7.173}$$

and

$$I_S(T) = I_S e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/(nV_{\rm TH})}\left(\frac{T}{T_{\rm NOM}}\right) \tag{7.174}$$

$$\phi_J(T) = \phi_J\frac{T}{T_{\rm NOM}} - 3V_{\rm TH}\ln\frac{T}{T_{\rm NOM}} + E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.175}$$

$$C_{J0}(T) = C_{J0}\{1 + M[0.0004(T - T_{\rm NOM}) + (1 - \phi_J(T)/\phi_J)]\} \tag{7.176}$$

$$\tag{7.177}$$

Parasitic Resistance

The parasitic diode series resistance R_S, is calculated by scaling the sheet resistivity R'_S (= `RS`) by the *Area* parameter on the element line

$$R_S = R'_S/Area \tag{7.178}$$

Current Characteristics

$$I_D = \begin{cases} AreaI_S\left(e^{\frac{V_D}{nV_{\rm TH}}} - 1\right) + V_D G_{\rm MIN} & V_D \geq -10nV_{\rm TH} \\ -Area\, I_S + V_D G_{\rm MIN} & -30V_{\rm TH} - V_B \leq V_D < -10nV_{\rm TH} \\ -Area\left(I_{BV}e^{-\frac{V_B + V_D}{V_{\rm TH}}} - I_S\right) + V_D G_{\rm MIN} & V_D < -30V_{\rm TH} - V_B \end{cases} \tag{7.179}$$

where $G_{\rm MIN}$ is `GMIN`, the minimum conductance between nodes. `GMIN` is set by a `.OPTIONS` statement (see page ??.)

Capacitance

Table 7.5: DIODE model parameters.

Name	Description	Units	Default	Area
AF	flicker noise exponent (A_F)	-	1	
BV	magnitude of reverse breakdown voltage (positive) (V_B)	V	∞	
CJO	zero-bias p-n junction capacitance per unit area (CJ-oh) (C_{J0})	F	0	*
EG	bandgap voltage (barrier height) at 0 K ($E_G(0)$) Schottky Barrier Diode: 0.69 Silicon: 1.16 Gallium Arsenide: 1.52 Germanium: 0.67	eV	1.11	
FC	forward-bias depletion capacitance (F_C)	-	0.5	
FC	coefficient for forward-bias depletion capacitance formula (F_C)	-		
IBV	magnitude of current at breakdown voltage per unit area (positive) (I_{BV})	A	$1\,10^{-10}$	*
IS	saturation current per unit area (I_S)	A	$1\,10^{-14}$	*
KF	flicker noise coefficient (K_F)	-	0	
M	p-n junction grading coefficient (M_J)	-	0.5	
N	emission coefficient (n)	-	1	
RS	ohmic resistance per unit area (R_S)	Ω	0	*
TT	transit time (τ_T)	s	0	
VJ	junction potential (ϕ_J)	V	1	
XTI	saturation current (I_S) temperature exponent (X_{TI})	-	3.0	

$$C = Area(C_J + C_D) \tag{7.180}$$

and the depletion capacitance at the junction per unit area is

$$C_J = \begin{cases} C_{J0}\left(1 - \frac{V_D}{\phi_J}\right)^{-M} & V_D < F_C\phi_J \\ \frac{C_{J0}}{F_2}\left(F_3 + \frac{mV_D}{\phi_J}\right) & V_D \geq F_C\phi_J \end{cases} \tag{7.181}$$

where ϕ_J, F_C, M and C_{J0} are the model parameters VJ, FC, M and CJO, and

$$F_2 = (1 - F_C)^{(1+m)} \tag{7.182}$$

$$F_3 = 1 - F_C(1 + m) \tag{7.183}$$

The diffusion capacitance per unit area due to charges in transit in the depleted region is

$$C_D = \tau\frac{\partial I_D}{\partial V_D} \tag{7.184}$$

where τ is the transit time model parameter TT.

AC Analysis

The AC analysis uses the model of figure 7.46 with the capacitor values evaluated at the `DC` operating point with the current source replaced by a linear resistor

$$R_D = \frac{\partial I_D}{\partial V_D} \tag{7.185}$$

Noise Analysis

The DIODE noise model accounts for thermal noise generated in the parasitic series resistance and shot and flicker noise generated in the the junction. The rms (root-mean-square) value of the thermal noise current generators shunting R_S is

$$I_{n,S} = \sqrt{4kT/R_S}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.186}$$

The RMS value of noise current generators shunting R_D in the `AC` model is

$$I_{n,J} = \left(I_{\mathrm{SHOT}}^2 + I_{\mathrm{FLICKER}}^2\right) \tag{7.187}$$

where the RMS shot noise current is

$$I_{\mathrm{SHOT}} = \sqrt{2qI_D} \quad \mathrm{A}/\sqrt{\mathrm{Hz}}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.188}$$

and the RMS flicker noise current is

$$I_{\mathrm{FLICKER}} = \sqrt{\frac{K_F I_D^{A_F}}{f K_{\mathrm{CHANNEL}}}} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.189}$$

where f is the analysis frequency and R_D is the `AC`junction resistance in (7.185)

E Voltage-Controlled Voltage Source

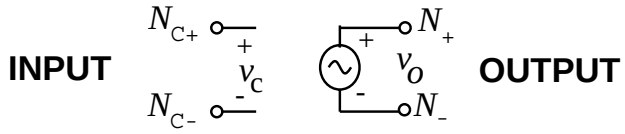


Figure 7.28: E — voltage-controlled voltage source element.

Form

E*name* N_+ N_- N_{C+} N_{C-} *Gain*
E*name* N_+ N_- `POLY(` *D* `)` N_{C+} N_{C-} *PolynomialCoefficients*

PSpice*Form*

E*name* N_+ N_- N_{C+} N_{C-} *Gain*
E*name* N_+ N_- `POLY(` *D* `)` (N_{C1+} N_{C1-}) ... (N_{CD+} N_{CD-}) *PolynomialCoefficients* E*name* N_+ N_- `VALUE=` { *Expression* }
E*name* N_+ N_- `TABLE` { *Expression* }`=(` *TableInput* `,` *TableOutput* `)` ...
E*name* N_+ N_- `LAPLACE` { *Expression* }`={` *TransformExpression* }
E*name* N_+ N_- `FREQ` { *Expression* }`=(` *Frequency*, *Magnitude*, *Phase* `)` ...
E*name* N_+ N_- `CHEBYSHEV` { *Expression* }`=` *Type*, *CutoffFrequency* ... , *Phase* ...

N_+ is the positive voltage source node.

N_- is the negative voltage source node.

N_{C+} is the positive controlling node.

N_{C-} is the negative controlling node.

Gain is the voltage gain.

`POLY` is the identifier for the polynomial form of the element.

D is the degree of the poynomial. The number of pairs of controlling nodes must be equal to *Degree*.

N_{Ci+} the positive node of the i th controlling node pair.

N_{Ci-} the negative node of the i th controlling node pair.

PolynomialCoefficients is the set of polynomial coefficients which must be specified in the standard polynomial coefficient format discussed on page 47.

`VALUE` is the identifier for the value form of the element.

Expression This is an expression of the form discussed on page 47.

`TABLE` is the identifier for the table form of the element.

TableInput This is the independent input of the table. See the `TABLE` parameter above.

TableInput This is the dependent output of the table. See the `TABLE` parameter above.

`LAPLACE` is the identifier for the laplace form of the element.

TransformExpression

`FREQ` is the identifier for the frequency form of the element.

Frequency

Magnitude

Phase

`CHEBYSHEV` is the identifier for the chebyshev form of the element.

Type

CutoffFrequency

Phase

Example	
`E1 2 3 14 1 2.0`	*Note*

1. Several form of the voltage-controlled voltage source element are supported in addition to the Linear Gain form which is the default. The other forms are selected based on the the identifier `POLY`, `VALUE`, `TABLE`, `LAPLACE`, `FREQ` or `CHEBYSHEV`.

Linear Gain Instance

E*name* N_+ N_- N_{C+} N_{C-} *Gain*

The value of the voltage generator is linearly proportional to the controlling voltage:

$$v_o = Gain\, v_c \tag{7.190}$$

`POLY`nomial Instance

E*name* N_+ N_- `POLY(` D `)` `(`N_{C1+} N_{C1-}`)` ... `(`N_{CD+} N_{CD-}`)` *PolynomialCoefficients*

The value of the voltage generator is a polynomial function of the controlling voltages:

$$v_o = f(v_{c1}, ..., v_{ci}, ...v_{cD}) \tag{7.191}$$

where the number of controlling voltages is D — the degree of the polynomial specified on the element line. v_{ci} is the ith controlling voltage and is the voltage of the n_{ci+} node with respect to the n_{ci+} node.

`VALUE` Instance — PSpice92 only

E*name* N_+ N_- `VALUE= {` *Expression* `}`

The value of the voltage generator is the resultant of an expression evaluation.

$$v_o = f(v_c) \tag{7.192}$$

`TABLE` Instance — PSpice92 only

E*name* N_+ N_- `TABLE {` *Expression* `}=(` *TableInput* `,` *TableOutput* `)` ...

$$v_o = f(v_c) \tag{7.193}$$

`LAPLACE` Instance — PSpice92 only

E*name* N_+ N_- `LAPLACE {` *Expression* `}={` *TransformExpression* `}`

$$v_o = f(v_c) \tag{7.194}$$

`FREQ` — PSpice92 only

E*name* N_+ N_- `FREQ {` *Expression* `}=(` *Frequency*, *Magnitude*, *Phase* `)` ...

$$v_o = f(v_c) \tag{7.195}$$

CHEBYSHEV — PSPICE92 only

E*name* N_+ N_- `CHEBYSHEV` { *Expression* }`=` *Type*`,` *CutoffFrequency ...* `,` *Phase ...*

F Current-Controlled Current Source

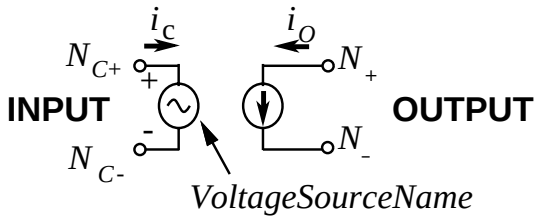


Figure 7.29: F — current-controlled current source element.

Form

F*name* N_+ N_- *VoltageSourceName Gain*
F*name* N_+ N_- POLY(D) *VoltageSourceName*$_1$ ... *VoltageSourceName*$_D$ *PolynomialCoefficients*

N_+ is the positive voltage source node.

N_- is the negative voltage source node.

VoltageSourceName is the name of the voltage source the current through which is the controlling current. The voltage source must be a V element.

Gain is the current gain.

POLY is the identifier for the polynomial form of the element

D is the degree of the poynomial. The number of pairs of controlling nodes must be equal to *Degree*.

VoltageSourceName$_i$ is the name of the voltage source the current through which is the ith controlling current. The voltage source must be a V element.

PolynomialCoefficients is the set of polynomial coefficients which must be specified in the standard polynomial coefficient format discussed on page 47.

Example

```
E1 2 3 14 1 2.0
```
Linear Gain Instance

F*name* N_+ N_- N_{C+} N_{C-} *Gain*

The value of the voltage generator is linearly proportional to the controlling current:

$$v_o = Gain\, v_c \tag{7.196}$$

POLYnomial Instance

`F`*name* N_+ N_- `POLY(` D `)` (N_{C1+} N_{C1-}) ... (N_{CD+} N_{CD-}) *PolynomialCoefficients*

The value of the voltage generator is a polynomial function of the controlling voltages:

$$v_o = f(i_{c1}, ..., i_{ci}, ...i_{cD}) \tag{7.197}$$

where the number of controlling currents is D — the degree of the polynomial specified on the element line. i_{ci} is the ith controlling current and is the current flowing from the + terminal to the − terminal in the ith voltage source of name *VoltageSourceName*.

G Voltage-Controlled Current Source

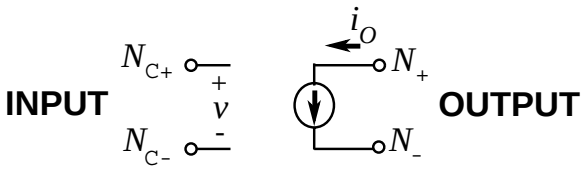


Figure 7.30: G — voltage-controlled current source element.

Form

G*name* N_+ N_- N_{C+} N_{C-} *Transconductance*
G*name* N_+ N_- `POLY(` *D* `)` N_{C+} N_{C-} *PolynomialCoefficients*

PSpice*Form*

G*name* N_+ N_- N_{C+} N_{C-} *Transconductance*
G*name* N_+ N_- `POLY(` *D* `)` (N_{C1+} N_{C1-}) ... (N_{CD+} N_{CD-}) *PolynomialCoefficients* G*name* N_+ N_- `VALUE=` { *Expression* }
G*name* N_+ N_- `TABLE` { *Expression* }`=(` *TableInput* , *TableOutput* `)` ...
G*name* N_+ N_- `LAPLACE` { *Expression* }={ *TransformExpression* }
G*name* N_+ N_- `FREQ` { *Expression* }`=(` *Frequency*, *Magnitude*, *Phase* `)` ...
G*name* N_+ N_- `CHEBYSHEV` { *Expression* }`=` *Type*, *CutoffFrequency* ... , *Phase* ...

N_+ is the positive voltage source node.

N_- is the negative voltage source node.

N_{C+} is the positive controlling node.

N_{C-} is the negative controlling node.

Transconductance is the transconductance.

`POLY` is the identifier for the polynomial form of the element

D is the degree of the poynomial. The number of pairs of controlling nodes must be equal to *Degree*.

N_{Ci+} the positive node of the i th controlling node pair.

N_{Ci-} the negative node of the i th controlling node pair.

PolynomialCoefficients is the set of polynomial coefficients which must be specified in the standard polynomial coefficient format discussed on page 47.

`VALUE` is the identifier for the value form of the element.

Expression This is an expression of the form discussed on page 47.

`TABLE` is the identifier for the table form of the element.

TableInput This is the independent input of the table. See the `TABLE` parameter above.

TableInput This is the dependent output of the table. See the `TABLE` parameter above.

`LAPLACE` is the identifier for the laplace form of the element.

TransformExpression

`FREQ` is the identifier for the frequency form of the element.

Frequency

Magnitude

Phase

`CHEBYSHEV` is the identifier for the chebyshev form of the element.

Type

CutoffFrequency

Phase

Example

```
G1 2 3 14 1 2.0
```

Note

1. Several form of the voltage-controlled voltage source element are supported in addition to the Linear Transconductance form which is the default. The other forms are selected based on the the identifier `POLY`, `VALUE`, `TABLE`, `LAPLACE`, `FREQ` or `CHEBYSHEV`.

Linear Transconductance Instance

G*name* N_+ N_- N_{C+} N_{C-} *Transconductance*

The value of the voltage generator is linearly proportional to the controlling voltage:

$$v_o = Transconductance\, v_c \tag{7.198}$$

`POLY`nomial Instance

G*name* N_+ N_- `POLY(` D `)` `(`N_{C1+} N_{C1-}`)` ... `(`N_{CD+} N_{CD-}`)` *PolynomialCoefficients*

The value of the voltage generator is a polynomial function of the controlling voltages:

$$v_o = f(v_{c1}, ..., v_{ci}, ...v_{cD}) \tag{7.199}$$

where the number of controlling voltages is D — the degree of the polynomial specified on the element line. v_{ci} is the ith controlling voltage and is the voltage of the n_{ci+} node with respect to the n_{ci+} node.

`VALUE` Instance — PSpice92 only

G*name* N_+ N_- `VALUE= {` *Expression* `}`

The value of the voltage generator is the resultant of an expression evaluation.

$$v_o = f(v_c) \tag{7.200}$$

`TABLE` Instance — PSpice92 only

G*name* N_+ N_- `TABLE {` *Expression* `}=(` *TableInput* `,` *TableOutput* `)` ...

$$v_o = f(v_c) \tag{7.201}$$

`LAPLACE` Instance — PSpice92 only

G*name* N_+ N_- `LAPLACE {` *Expression* `}={` *TransformExpression* `}`

$$v_o = f(v_c) \tag{7.202}$$

`FREQ` — PSpice92 only

G*name* N_+ N_- `FREQ {` *Expression* `}=(` *Frequency*`,` *Magnitude*`,` *Phase* `)` ...

$$v_o = f(v_c) \tag{7.203}$$

CHEBYSHEV — PSPICE92 only

G*name* N_+ N_- CHEBYSHEV { *Expression* }= *Type*, *CutoffFrequency ...* , *Phase ...*

H Current-Controlled Voltage Source

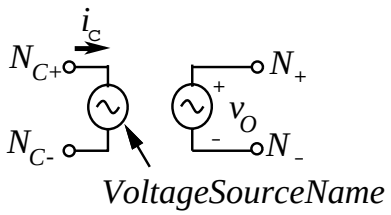


Figure 7.31: H — current-controlled voltage source element.

Form

H*name* N_+ N_- *VoltageSourceName Transresistance*
H*name* N_+ N_- `POLY(` D `)` *VoltageSourceName*$_1$ ... *VoltageSourceName*$_D$ *PolynomialCoefficients*

N_+ is the positive voltage source node.

N_- is the negative voltage source node.

VoltageSourceName is the name of the voltage source the current through which is the controlling current. The voltage source must be a `V` element.

Transresistance is the Transresistance of the element.

`POLY` is the identifier for the polynomial form of the element

D is the degree of the poynomial. The number of pairs of controlling nodes must be equal to *Degree*.

VoltageSourceName$_i$ is the name of the voltage source the current through which is the ith controlling current. The voltage source must be a `V` element.

PolynomialCoefficients is the set of polynomial coefficients which must be specified in the standard polynomial coefficient format discussed on page 47.

Example

```
E1 2 3 14 1 2.0
```

Linear Transresistance Instance

H*name* N_+ N_- N_{C+} N_{C-} *Transresistance*

The value of the voltage generator is linearly proportional to the controlling current:

$$v_o = Transresistance\, v_c \tag{7.204}$$

`POLY`nomial Instance

H*name* N_+ N_- POLY(D) (N_{C1+} N_{C1-}) ... (N_{CD+} N_{CD-}) *PolynomialCoefficients*

The value of the voltage generator is a polynomial function of the controlling voltages:

$$v_o = f(i_{c1}, ..., i_{ci}, ...i_{cD} \tag{7.205}$$

where the number of controlling currents is D — the degree of the polynomial specified on the element line. i_{ci} is the ith controlling current and is the current flowing from the + terminal to the − terminal in the ith voltage source of name *VoltageSourceName*.

I Independent Current Source

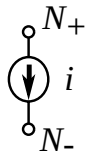


Figure 7.32: I — independent current source.

Form

I*name* N_+ N_- [[*DC*] [*DCvalue*]
+ [AC [*ACmagnitude* [*ACphase*]]]
+ [DISTOF1 [*F1Magnitude* [*F1Phase*]]] + [DISTOF2 [*F2Magnitude* [*F2Phase*]]
]

SPICE3*Form*

I*name* N_+ N_- [[*DC*] [*DCvalue*]
+ [AC [*ACmagnitude* [*ACphase*]]]
+ [*TransientSpecification*]
+ [DISTOF1 [*F1Magnitude* [*F1Phase*]]]
+ [DISTOF2 [*F2Magnitude* [*F2Phase*]]]

PSPICE*Form*

I*name* N_+ N_- [[*DC*] [*DCvalue*] [AC [*ACmagnitude* [*ACphase*]]]
+ [*TransientSpecification*]

Example

```
IBIAS 1 0 1.0M
ICLOCK 20 5 PULSE(0M 10M 1N 2N 1.5N 21.9N 5N 20N)
ISSIGNAL AC 1U 90
```

N_+ is the positive current source node. (Current flow is out of the positive to the negative node

N_- is the negative current source node.

`DC` is the optional keyword for the `DC` value of the source.

DCvalue is the `DC` current value of the source.
(Units: A; Optional; Default: 0; Symbol: I_{DC})

`AC` is the keyword for the `AC` value of the source.

ACmagnitude is the `AC` magnitude of the source used during `AC`analysis. That is, it is the peak `AC` current so that the `AC` signal is $ACmagnitude \sin(\omega t + \text{ACphase})$. *ACmagnitude* is ignored for other types of analyses.
(Units: A; Optional; Default: 1; Symbol: I_{AC})

ACphase is the ac phase of the source. It is used only in `AC` analysis.
(Units: Degrees; Optional; Default: 0; Symbol: ϕ_{AC})

`DISTOF1` is the distortion keyword for distortion component 1 which hass frequency `F1`. (see the description of the `.DISTO` statement on page 58).

F1magnitude is the magnitude of the distortion component at `F1`. See `.DISTOF1` keyword above.
(Units: A; Optional; Default: 1; Symbol: I_{F1})

F1phase is the phase of the distortion component at `F1`. See `.DISTOF1` keyword above.
(Units: Degrees; Optional; Default: 0; Symbol: ϕ_{F1})

`DISTOF2` is the distortion keyword for distortion component 2 which hass frequency `F2`. (see the description of the `.DISTO` statement on page 58).

F2magnitude is the magnitude of the distortion component at `F2`. See `.DISTOF2` keyword above.
(Units: A; Optional; Default: 1; Symbol: I_{F2})

F2phase is the phase of the distortion component at `F2`. See `.DISTOF2` keyword above.
(Units: Degrees; Optional; Default: 0; Symbol: ϕ_{F2})

TransientSpecification is the optional transient specification described more fully below.

Note

1. The independent current source has three different sets of parameters to describe the source for DC analysis (see `.DC` on page 55), AC analysis (see `.AC` on page 53), and transient analysis (see `.TRAN` on page 109). The `DC` value of the source is used during bias point evaluation and `DC` analysis is *DCValue*. It is also the constant value of the current source if no *TransientSpecification* is supplied. It may also be used in conjunction with the `PWL` transient specification if a time zero value is not provided as part of the transient specification. The `AC` specification, indicated by the keyword `AC` is independent of the `DC` parameters and the *Transient Specification*.

2. The original documentation distributed with SPICE2G6 and SPICE3 incorrectly stated that if a *TransientSpecification* was supplied then the time-zero transient current was used in `DC` analysis and in determiniong the operating point.

Transient Specification

Five transient specification forms are supported: pulse (`PULSE`), exponential (`EXP`), sinusoidal (`SIN`), piece-wise linear (`PWL`), and single-frequency FM (`SFFM`). The default values of some of the parameters of these transient specifications include `TSTEP` which is the printing increment and `TSTOP` which is the final time (see the `.TRAN` statement on page 109 for further explanation of these quantities). In the following t is the transient analysis time.

Exponential:

Form

`EXP(` I_1 I_2 [T_{D1}] [τ_1] [T_{D2}] [τ_2] `)`

Name	Description	Units	Default
I_1	initial current	A	REQUIRED
I_2	pulsed current	A	REQUIRED
T_{D1}	rise delay time	s	0.0
τ_1	rise time constant	s	TSTEP
T_{D2}	fall delay time	s	T_{D1} + TSTEP
τ_2	fall time constant	s	TSTEP

The exponential transient is a single-shot event specifying two exponentials. The current is I_1 for the first T_{D1} seconds at which it begins increasing exponentially towards I_2 with a time constant of τ_1 seconds. At time T_{D2} the current exponentially decays towards I_1 with a time constant of τ_2. That is,

$$i = \begin{cases} I_1 & t \le T_{D1} \\ I_1 + (I_2 - I_1)(1 - e^{(-(t - T_{D1})/\tau_1)}) & T_{D1} < t \le T_{D2} \\ I_1 + (I_2 - I_1)(1 - e^{(-(t - T_{D1})/\tau_1)}) + (I_1 - I_2)(1 - e^{(-(t - T_{D2})/\tau_2)}) & t > T_{D2} \end{cases} \quad (7.206)$$

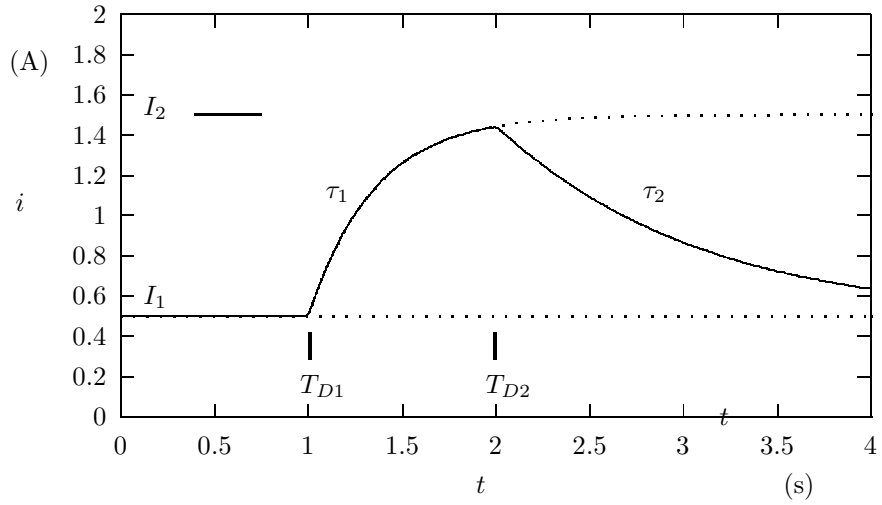


Figure 7.33: Current source exponential (`EXP`) waveform for `EXP(0.1 0.8 1 0.35 2 1)`

Single-Frequency FM:

Form

`SFFM(` V_O V_A F_C μ F_S `)`

Name	Description	Units	Default
I_O	offset current	A	
I_A	peak amplitude of AC current	A	
F_C	carrier frequency	Hz	1/TSTOP
μ	modulation index	-	0
F_S	signal frequency	Hz	1/TSTOP

The single frequency frequency modulated transient response is described by

$$i = I_O + I_A \sin\left(2\pi\, F_C t + \mu \sin\left(2\pi F_S t\right)\right) \tag{7.207}$$

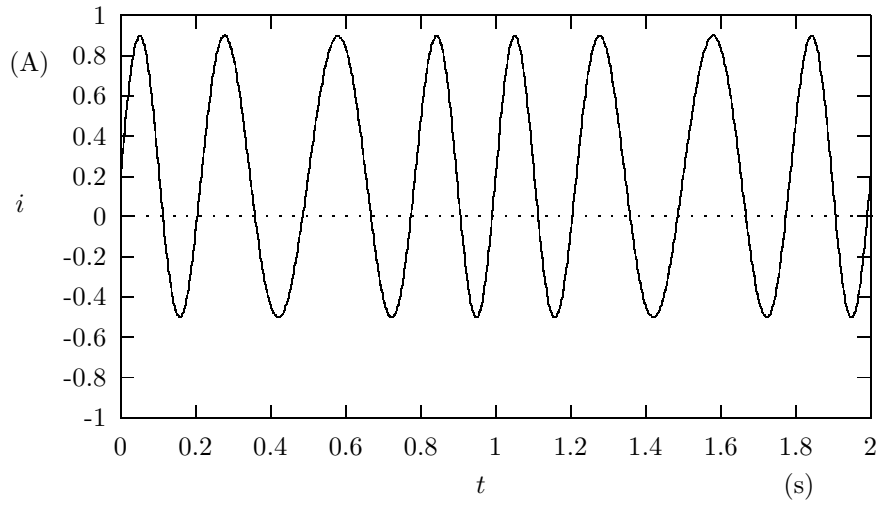


Figure 7.34: Current source single frequency frequency modulation (SFFM) waveform for
SFFM(0.2 0.7 4 0.9 1) .

Pulse:

Form

PULSE(V_1 V_2 [T_D] [T_R] [T_F] [W] [T])

Name	Description	Units	Default
I_1	initial current	A	REQUIRED
I_2	pulsed current	A	REQUIRED
T_D	delay time	s	0.0
T_R	rise time	s	TSTEP
T_F	fall time	s	TSTEP
W	pulse width	s	TSTOP
T	period	s	TSTOP

The pulse transient waveform is defined by

$$i = \begin{cases} I_1 & t \le T_D \\ I_1 + \frac{t'}{T_R}(I_2 - I_1) & 0 < t' \le T_R \\ I_2 & T_R < t' < (T_R + W) \\ I_2 - \frac{t' - W}{T_F}(I_1 - I_2) & (T_R + W) < t' < (T_R + W + T_F) \\ I_1 & (T_R + W + T_F) < t' < T \end{cases} \tag{7.208}$$

where

$$t' = t - T_D - (n-1)T \tag{7.209}$$

and t is the current analysis time and n is the cycle index. The effect of this is that after an initial time delay T_D the transient waveform repeats itself every cycle.

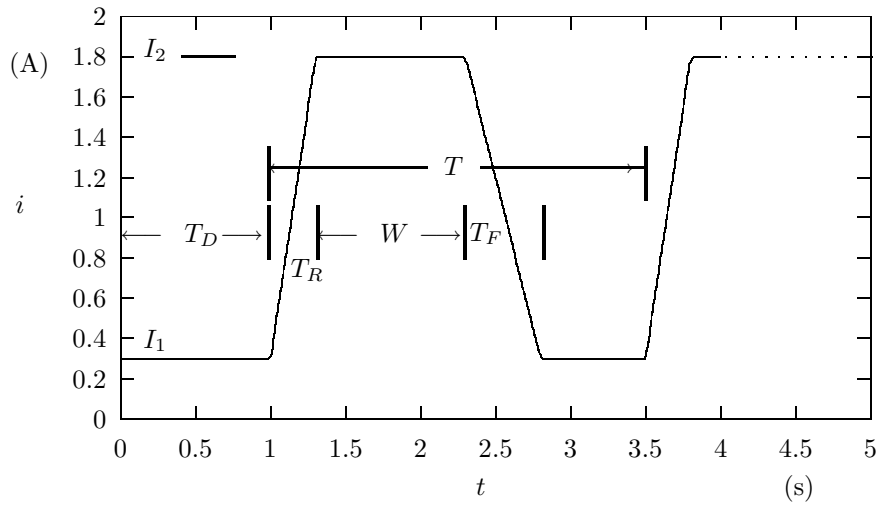


Figure 7.35: Current source transient pulse (PULSE) waveform for PULSE(0.3 1.8 1 2.5 0.3 1 0.7)

Piece-Wise Linear:

Form

PWL(T_1 I_1 [T_2 I_2 ... T_i I_i ... T_N I_N])

Each pair of values (T_i, I_i) specifies that the value of the source is I_i at time = T_i. At times between T_i and T_{i+1} the values are linearly interpolated. If $T_1 > 0$ then the current is constant at *DCValue* (specified on the element line) until time T_1.

$$i = \begin{cases} DCvalue & t < T_1 \\ I_i & t = T_i \\ I_{i+1} & t = T_{i+1} \\ I_i + \left(\frac{t-T_i}{T_{i+1}-T_i}\right)(I_{i+1} - I_i) & T_i < t \leq T_{i+1} \\ I_N & t > T_N \end{cases} \tag{7.210}$$

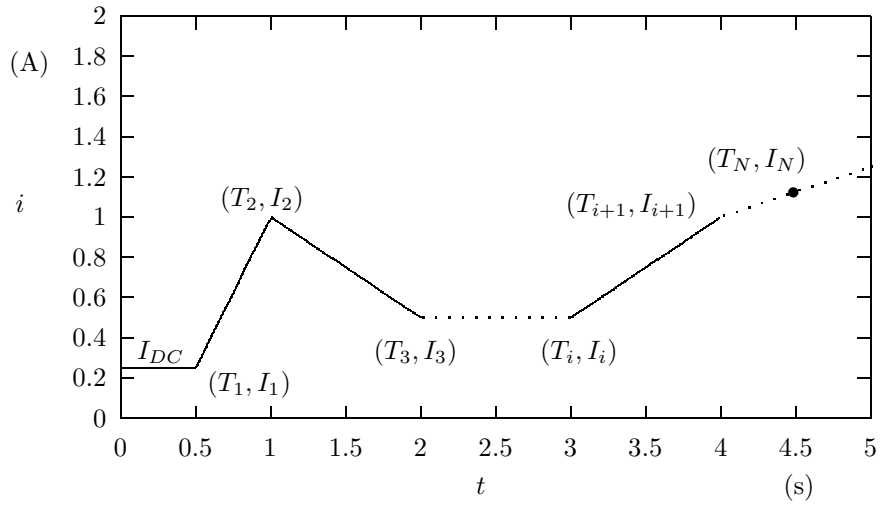


Figure 7.36: Current source transient piece-wise linear (`PWL`) waveform for
`PWL(1 0.25 1 1 2 0.5 ... 3 0.5 4 1 ... 4.5 1.25 ...)` with *DCValue = 0.25*.

Sinusoidal:

Form

SIN(V_O V_A [F] [T_D] [θ])

PSPICE*Form*

SIN(V_O V_A [F] [T_D] [θ ϕ])

PSPICE*Form*

SIN(V_O V_A [F] [T_D] [θ ϕ])

Name	Description	Units	Default
I_O	current offset	A	REQUIRED
I_A	current amplitude	A	REQUIRED
F	frequency	Hz	1/TSTOP
T_D	time delay	s	0
Θ	damping factor	1/s	0
ϕ	phase	degree	0

The sinusoidal transient waveform is defined by

$$i = \begin{cases} I_0 & t \leq T_D \\ I_0 + I_1 e^{-[(t-T_D)\Theta]} \sin 2\pi[F(t-T_D) + \phi/360] & t > T_D \end{cases} \tag{7.211}$$

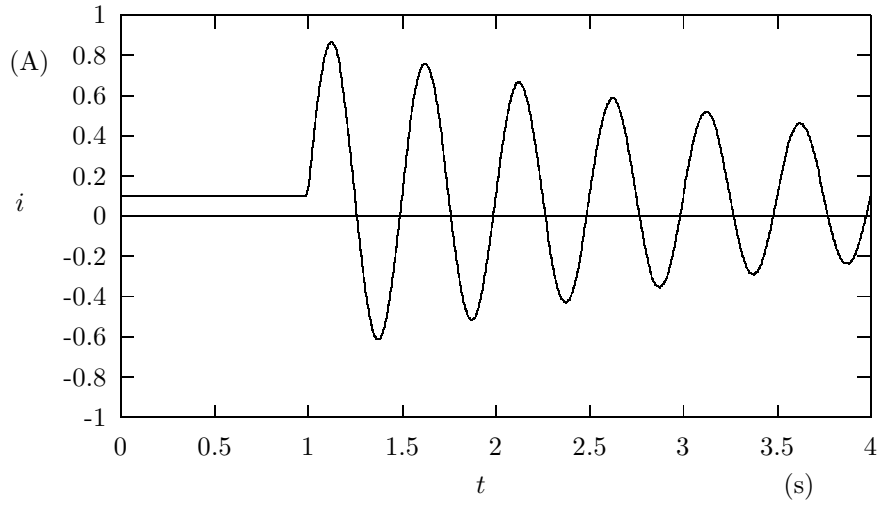


Figure 7.37: Current source transient sine (SIN) waveform for SIN(0.1 0.8 2 1 0.3).

J Junction Field-Effect Transistor

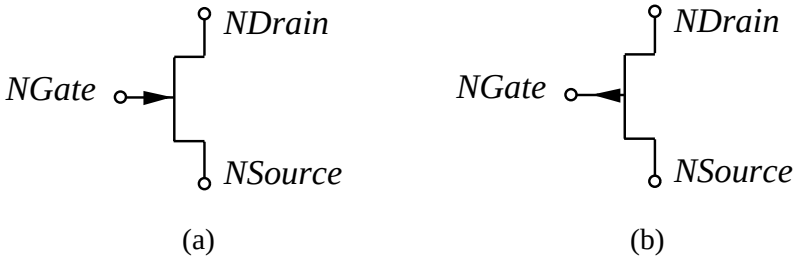


Figure 7.38: J — Junction field effect transistor element: (a) n channel JFET; (b) p channel JFET.

Form

`J`*name NDrain NGate NSource ModelName* [*Area*] [`OFF`] [`IC=`V_{DS}, V_{GS}]

PSPICE*Form*

`J`*name NDrain NGate NSource ModelName* [*Area*]

Example

```
J1 7 2 3 JM1 OFF
```

NDrain is the drain node

NGate is the gate node

NSource is the source node

`ModelName` is the model name

Area is the area factor. (Units: none; Optional; Default: 1; Symbol: *Area*)

`OFF` indicates an (optional) initial condition on the device for `DC` operating point analysis. If specified the `DC` operating point is calculated with the terminal voltages set to zero. Once convergence is obtained, the program continues to iterate to obtain the exact value of the terminal voltages. The OFF option is used to enforce the solution to correspond to a desired state if the circuit has more than one stable state.

`IC` is the optional initial condition specification. Using `IC=`V_{DS}, V_{GS} is intended for use with the `UIC` option on the `.TRAN` line, when a transient analysis is desired starting from other than the quiescent operating point. Specification of the transient initial conditions using the `.IC` statement (see page 66) is preferred and is more convenient.

NJF Model N-Channel JFET Model

PJF Model P-Channel JFET Model

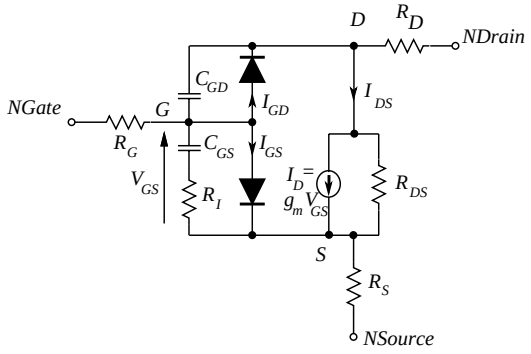


Figure 7.39: Schematic of the JFET model. V_{GS}, V_{DS}, and V_{GD} are intrinsic gate-source, drain-source and gate-drain voltages between the internal gate, drain, and source terminals designated G, D, and C respectively.

The parameters of the *n*-channel (NJF) and of the *p*-channel (PJF) models are the same and are given in table 7.6.

The parameters of the JFET can be completely specified in the model *ModelName*. This facilitates the use of standard transistors by using absolute quantities in the model. Alternatively scalable process parameters can be specified in the model *ModelName* and these scaled by the *Area* parameter on the JFET element line. The parameters that can be scaled by *Area* are `BETA`, `CGS`, `CGD`, `IS`, `RD` and `RS`.

The physical constants used in the model evaluation are

k	Boltzman's constant	$1.3806226\,10^{-23}$ J/K
q	electronic charge	$1.6021918\,10^{-19}$ C

Standard Calculations

Absolute temperatures (in kelvins, K) are used. The thermal voltage

$$V_{\mathrm{TH}} = \frac{kT_{\mathrm{NOM}}}{q}. \tag{7.212}$$

The silicon bandgap energy

$$E_G = 1.16 - 0.000702\frac{4T_{\mathrm{NOM}}^2}{T_{\mathrm{NOM}} + 1108}. \tag{7.213}$$

Temperature Dependence

Table 7.6: NJF and PJF model keywords for the junction field effect transistor. The Area column indicates parameters that are scaled by *Area*.

Name	Description	Units	Default	Area
AF	flicker noise exponent (A_F)	-	1	
BETA	transconductance parameter (β)	A/V^2	1.0E-4	*
CGS	zero-bias G-S junction capacitance per unit area (C'_{GS})	F	0	*
CGD	zero-bias G-D junction capacitance per unit area (C'_{GD})	F	0	*
FC	coefficient for forward-bias depletion capacitance formula (F_C)	-	0.5	
IS	gate junction saturation current (I_S)	A	1.0E-14	*
KF	flicker noise coefficient (K_F)	-	0	
LAMBDA	channel length modulation parameter (λ)	1/V	0	
PB	gate junction potential (ϕ_J)	V	1	
RD	drain ohmic sheet resistance (R_D)	Ω	0	*
RS	source ohmic sheet resistance (R_S)	Ω	0	*
VTO	threshold voltage (VT-oh) (V_{T0}) VTO < 0 indicates a depletion mode JFET VTO ≥ 0 indicates an enhancement mode JFET	V	-2.0	
BETATC	temperature coefficient of the transconductance parameter BETA ($T_{C,\beta}$)	%/°C	0	
M	gate p-n junction grading coefficient (M)	-	0.5	
VTOTC	temperature coefficient of threshold voltage VTO ($T_{C,VT0}$)	V/°C	0	

Temperature effects are incorporated as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K).

$$V_{\rm TH} = \frac{kT}{q} \tag{7.214}$$

$$I_S(T) = I_S e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/(nV_{\rm TH})} \tag{7.215}$$

$$V_{BI}(T) = V_{BI}\frac{T}{T_{\rm NOM}} - 3V_{\rm TH}\ln\frac{T}{T_{\rm NOM}} + E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.216}$$

$$C'_{GS}(T) = C'_{GS}\{1 + M[0.0004(T - T_{\rm NOM}) + (1 - V_{BI}(T)/V_{BI})]\} \tag{7.217}$$

$$C'_{GD}(T) = C'_{GD}\{1 + M[0.0004(T - T_{\rm NOM}) + (1 - V_{BI}(T)/V_{BI})]\} \tag{7.218}$$

$$\beta(T) = \beta 1.01^{T_{C,\beta}(T - T_{\rm NOM})} \tag{7.219}$$

$$V_{T0}(T) = V_{T0} + T_{C,VT0}\,(T - T_{\rm NOM}) \tag{7.220}$$

Parasitic Resistances

The parasitic resistances are calculated from the sheet resistivities RS, RG, RD, and the *Area* specified on the element line.

$$R_S = R'_S Area \tag{7.221}$$

$$R_G = R'_G Area \tag{7.222}$$

$$R_D = R'_D Area \tag{7.223}$$

The parasitic resistance parameter dependencies are summarized in figure 7.40.

PROCESS PARAMETERS	
RSH	R_{SH}

+

GEOMETRY PARAMETERS	
NRS	N_{RS}
NRD	N_{RD}
NRG	N_{RG}
NRB	N_{RB}

→

DEVICE PARAMETERS	
RD	$R_D = f(R_{\text{SH}}, N_{RD})$
RS	$R_S = f(R_{\text{SH}}, N_{RS})$
RG	$R_B = f(R_{\text{SH}}, N_{RG})$
RB	$R_B = f(R_{\text{SH}}, N_{RB})$

Figure 7.40: JFET parasitic resistance parameter relationships.

Leakage Currents

Current flows across the normally reverse biased source-bulk and drain-bulk junctions. The gate-source leakage current

$$I_{GS} = Area\, I_S e^{(V_{GS}/V_{\text{TH}} - 1)} \tag{7.224}$$

and the gate-source leakage current

$$I_{GD} = Area\, I_S e^{(V_{GD}/V_{\text{TH}} - 1)} \tag{7.225}$$

The dependencies of the parameters describing the leakage current in the JFET model are summarized in figure 7.41.

PROCESS PARAMETERS	
IS	I_S

+

GEOMETRY PARAMETERS
Area

→

DEVICE PARAMETERS
$I_{GS} = f(I_S, Area)$
$I_{GD} = f(I_S, Area)$

Figure 7.41: JFET leakage current parameter dependecies.

I/V Characteristics

The current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \geq 0$)

The regions are as follows:

cutoff region:	$V_{GS} < V_{T0}$
linear region:	$V_{GS} \geq V_{T0}$ and $V_{GS} > V_{DS} + V_{T0}$
saturation region:	$V_{GS} \geq V_{T0}$ and $V_{GS} \leq V_{DS} + V_{T0}$

Then

$$I_D = \begin{cases} 0 & \text{cutoff region} \\ Area\,\beta\,(1+\lambda V_{DS})\,V_{DS}\,[2\,(V_{GS}-V_{T0})-V_{DS}] & \text{linear region} \\ Area\,\beta\,(1+\lambda V_{DS})\,(V_{GS}-V_{T0})^2 & \text{saturation region} \end{cases} \tag{7.226}$$

Inverted Mode: ($V_{DS} < 0$)

In the inverted mode the JFET I/V characteristics are evaluated as in the normal mode (7.226) but with the drain and source subscripts exchanged. The relationships of the parameters describing the I/V characteristics are summarized in figure 7.42.

PROCESS PARAMETERS	
ALPHA	α
BETA	β
LAMBDA	λ
VTO	V_{T0}

+

GEOMETRY PARAMETERS
–
Optional
Area

→

DEVICE PARAMETERS
$\{I_D = f(Area, \beta, \lambda, V_{T0}, \alpha)\}$

Figure 7.42: I/V dependencies.

Capacitances

The drain-source capacitance

$$C_{DS} = Area\,C'_{DS} \tag{7.227}$$

The gate-source capacitance

$$C_{GS} = \begin{cases} Area\,C'_{GS}\left(1-\frac{V_{GS}}{\phi_J}\right)^{-M} & V_{GS} \le F_C\phi_J \\ Area\,C'_{GS}\,(1-F_C)^{-(1+M)}\left[1-F_C(1+M)+M\frac{V_{GS}}{\phi_J}\right]^{-M} & V_{GS} > F_C\phi_J \end{cases} \tag{7.228}$$

models charge storage at the gate-source depletion layer. The gate-drain capacitance

$$C_{GD} = \begin{cases} Area\,C'_{GD}\left(1-\frac{V_{GD}}{\phi_J}\right)^{-M} & V_{GD} \le F_C\phi_J \\ Area\,C'_{GD}\,(1-F_C)^{-(1+M)}\left[1-F_C(1+M)+M\frac{V_{GD}}{\phi_J}\right]^{-M} & V_{GD} > F_C\phi_J \end{cases} \tag{7.229}$$

models charge storage at the gate-drain depletion layer. The capacitance parameter dependencies are summarized in figure 7.43.

AC Analysis

The AC analysis uses the model of figure 7.46 with the capacitor values evaluated at the DC operating point with

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \tag{7.230}$$

and

$$R_{DS} = \frac{\partial I_{DS}}{\partial V_{DS}} \tag{7.231}$$

Noise Analysis

The JFET noise model accounts for thermal noise generated in the parasitic resistances and shot and flicker noise generated in the drain source current generator. The rms (root-mean-square) values of thermal noise current generators shunting the four parasitic resistance R_D, R_G and R_S are

$$I_{n,D} = \sqrt{4kT/R_D}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.232}$$

$$I_{n,S} = \sqrt{4kT/R_S}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.233}$$

Shot and flicker noise are modeled by a noise current generator in series with the drain-source current generator. The rms value of this noise generator is

$$I_{n,DS} = \left(I^2_{\mathrm{SHOT},DS} + I^2_{\mathrm{FLICKER},DS}\right) \tag{7.234}$$

$$I_{\mathrm{SHOT},DS} = \sqrt{4kTg_m\frac{2}{3}}\quad \mathrm{A}/\sqrt{\mathrm{Hz}}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.235}$$

$$I_{\mathrm{FLICKER},DS} = \sqrt{\frac{K_F I_{DS}^{A_F}}{f}}\quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.236}$$

where the transconductance

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \tag{7.237}$$

is evaluated at the `DC` operating point, and f is the analysis frequency.

PROCESS PARAMETERS	
`FC`	F_C
`M`	M
`PB`	ϕ_J

+

GEOMETRY PARAMETERS
–
Area

→

DEVICE PARAMETERS	
`CDS`	C'_{DS}
`CGS`	C'_{GS}
`CGD`	C'_{GD}
$\{C_{DS} = f(Area, C'_{DS})\}$	
$\{C_{GS} = f(Area, C'_{GS}, FC, M, PB)\}$	
$\{C_{GD} = f(Area, C'_{GD}, FC, M, PB)\}$	

Figure 7.43: JFET capacitance dependencies.

K Mutual Inductor

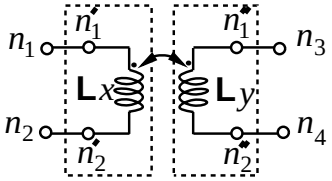


Figure 7.44: K — Mutual inductor element.

Form

K*name* L*name1* L*name2* *CouplingValue*

PSPICE*Form*

K*name* L*name1* L*name2* [... L*nameN*] *CouplingValue*

K*name* L*name1* [L*name2* ... L*nameN*] *CouplingValue* [[*ModelName* [*size*]]

L*name1* is the name of the first inductor of the coupled inductor list. The first node of L*name1* is dotted using the dot convention. In the mutual coupled inductor model (the default model) the value of L*name1* is the self inductance L_1. In the transformer `CORE` model (which is used if a *ModelName* is supplied) the value of L*name1* is the number of turns N_1. (Note, *ModelName* can not be specified with the SPICE2G6 and SPICE3 simulators.)
(Required)

L*name2* is the name of the second inductor in the coupled inductor list. The first node of L*name1* is dotted using the dot convention. In the mutual coupled inductor model the value of L*name1* is the self inductance L_2. In the transformer `CORE` model (which is used if a *ModelName* is supplied the value of L*name2* is the number of turns N_2.
(SPICE2G6 and SPICE3: Required.)
(PSPICE: Required if *Modelname* not supplied; Optional if *Modelname* supplied.)

L*nameN* is the name of Nth inductor in the coupled inductor list. The first node of L*nameN* is dotted using the dot convention. In the mutual inductor model the value of L*nameN* is the self inductance L_N. In the transformer `CORE` model (which is used if a *ModelName* is supplied the value of L*name2* is the number of turns N_N.
Not valid in SPICE2G6 or SPICE3 for $N > 2$.
(PSPICE: Optional if *Modelname* supplied.)

CouplingValue is the coefficient of mutual coupling of the inductors.
(Units: none; Required; Symbol: K_{COUPLING}; $0 < K_{\text{COUPLING}} \leq 1$)

ModelName is the optional model name. PSPICE only.

Size is the size scaling factor. It scales the magnetic cross-section and represents the number of lamination layers.
(Units: none; Optional; Default: 1; Symbol: *Size*)

Example

```
K43 LAA LBB 0.999
KXFRMR L1 L2 0.87
```

PSpice *Example*

```
KTFMR LAA LBB LCC LDD TRANSFORMER 2.5
```

Model Type

`IND` PSpice `only`

Note

1. The mutual coupled inductor model represents coupled inductors by self inductances L_i and mutual inductances M_{ij}. This is the model used in Spice2g6 and spicthree and in PSpice if a `CORE model` is not supplied. Here L_i is the self inductance of the ith inductor element and M_{ij} is the mutual inductance of the ith and jth inductor elements. The mathematical model of the coupled element consists of voltage sources controlled by the time derivatives of current. If two inductors are coupled

$$V_1 = L_1 \frac{dI_1}{dt} + M_{12} + \frac{dI_2}{dt} \tag{7.238}$$

and

$$V_2 = L_2 \frac{dI_2}{dt} + M_{21} + \frac{dI_1}{dt} \tag{7.239}$$

If N inductors are coupled, as supported in PSpice , the mathematical model is

Note

$$V_i = L_i \frac{dI_i}{dt} + \sum_{\substack{j=1 \\ j \neq i}}^{N} M_{ij} \frac{dI_j}{dt} \tag{7.240}$$

2 The mutual inductance M_{ij} is determined from the self-inductances L_i and L_j of the inductors and the coupling coefficient K_{COUPLING} supplied as an element parameter by

$$K_{\text{COUPLING}} = \sqrt{\frac{M_{ij}}{L_i L_j}} \tag{7.241}$$

K_{COUPLING} may have any value between 0 and 1 including 1. Ferrite core provides almost ideal coupling with $K = 0.999$ or higher.

3 In SPICE2G6 and SPICE3 a transformer with several coils must be represented by several `K` elements. For example, a transformer with one primary and two secondaries is specified as

```
* PRIMARY
L1 1 2 100U
* FIRST SECONDARY
L2 3 4 100U
* SECOND SECONDARY
L3 5 6 100U
* TRANSFORMER
K1 L1 L2 0.999
K2 L1 L3 0.999
K2 L2 L3 0.999
```

4 In PSPICE the transformer above can be either represented using the SPICE2G6 and SPICE3 format above or by the more compact format

```
* PRIMARY
L1 1 2 100U
* FIRST SECONDARY
L2 3 4 100U
* SECOND SECONDARY
L3 5 6 100U
* TRANSFORMER
K1 L1 L2 L3 0.999
```

CORE Model — PSPICE Only — Magnetic Core Model

Form

`.MODEL` *ModelName* `CORE(` [[*keyword* = *value*] ...] `)`

Example

```
.MODEL TRANSFORMER CORE(AREA=1 PATH=9.8 GAP=0.1 MS=1.250M)
```

Table 7.7: Model parameters.

Name	Description	Units	Default
A	shape parameter (A)	A/M	10^3
ALPHA	(α) interdomain coupling parameter	-	0.001
AREA	mean magnetic crossection $(Area)$	cm^2	0.1
GAMMA	domain damping parameter. (γ)	-	∞
C	domain flexing parameter (C)	-	0.2
GAP	effective air-gap length $(L_{\rm GAP})$	cm	0
K	domain anisotopy parameter (pinning constant) (K)	A/M	500
MS	magnetization saturation (M_S)	A/M	10^6
PACK	pack (stacking) factor $(F_{\rm PACK})$	cm	0
PATH	mean magnetic path length in the core $(L_{\rm PATH})$	cm	1

The CORE model models a transformer core. It is assumed that the model parameters were determined or measured at the nominal temperature $T_{\rm NOM}$ (default $27°C$) specified in the most recent .OPTIONS statement preceeding the .MODEL statement.

The CORE model uses the Jiles-Atherton model described in [25]. This model is based on domain wall motion and includes flexing of the domain wall, interdomain coupling, coercivity, remanence and magnetic saturation. Hysteresis due to domain wall pinning at defect sites is modeled. This impedance to domain wall motion dominates the characteristics of magnetic devices.

As with the default mutually coupled inductor model, the CORE model calculates the voltage across the ith set of windings from the total ampere turns which is the magnetomotive force MMF. Thus

$$V_i = \frac{d\phi_i}{dt} = f(MMF) \tag{7.242}$$

where

$$MMF = \sum_{j=1}^{N} N_j I_j \tag{7.243}$$

Here the number of turns of the jth winding, N_j, is the "*InductanceValue*" of L_j the name of which is the jth L*name* given on the K element line. I_i is the curent flowing through the ith winding. $A_{\rm TURNS}$ produces the magnetic field $H_{\rm CORE}$ in the core. This in turn produces the B field. The B field is proportional to the flux, in the core and hence to the voltage V_i. The relationship between B and H in the core is nonlinear and hysteretic. The airgap also affects the B-H relationship.

Air-Gap Effect

Along the complete magnetic path

$$H_{\rm CORE} L_{\rm PATH} + H_{\rm GAP} L_{\rm GAP} = MMF \tag{7.244}$$

where $H_{\rm CORE}$ is the magnetic field in the core and $H_{\rm GAP}$ is the magnetic field in the air gap. $L_{\rm PATH}$ and $L_{\rm GAP}$ are the model parameters PATH and GAP. If the air gap is small then all of the flux in the core passes through the air gap so that $B_{\rm GAP} = B_{\rm CORE}$. In the air-gap the magnetization is negligible so that $B_{\rm GAP} = H_{\rm GAP}$

This leads to a relationship beween the B and H fields in the core:

$$H_{\rm CORE}L_{\rm PATH} + B_{\rm CORE}L_{\rm GAP} = MMF \tag{7.245}$$

It is a simple matter to solve for $B_{\rm CORE}$ and $H_{\rm CORE}$ if $L_{\rm GAP}$ =0 as then

$$H_{\rm CORE} = \frac{MMF}{L_{\rm PATH}} \tag{7.246}$$

If $L_{\rm GAP} > 0$ then (7.245) must be solved in conjunction with the relationship between $H_{\rm CORE}$ and magnetization M in the core. This relationship is based on the theory of loosely coupled domains developed by Jiles and Atherton.

Jiles-Atherton Model

The B-H curve of a magnetic material biased by `AC` and `DC` magnetic fields is called the anhysteric and is mathematically described by the Jiles-Atherton model. This model determines an anhysteric magnetization $M_{\rm AN}$ which is related to the saturation magnetization M_S by

$$M_{\rm AN} = M_S\left[\coth\left(\frac{H_{\rm EFF}}{Size\,A}\right) - \frac{Size\,A}{H_{\rm EFF}}\right] \tag{7.247}$$

where A is the shape parameter and the effective field in the core

$$H_{\rm EFF} = H_{\rm CORE} + \alpha M_{\rm AN} \tag{7.248}$$

Here H is the magnetizing influence. Domain wall flux is magnetic current which is proportional to the change in magnetization. The change in magnetization consists of a reversible component due to flexing of the domain walls and an irreversible component due to movement of domain walls from one pinning location to another. Energy is dissipated (hence the motion is irreversible) in moving the domain wall from one pinning location to another but energy is stored (hence reversible) when the domain wall flexs. This is mathematicly modeled by

$$\frac{dM}{dH_{\rm CORE}} = \left(\frac{dM}{dH_{\rm CORE}}\right)_{\rm REVERSIBLE} + \left(\frac{dM}{dH_{\rm CORE}}\right)_{\rm IRREVERSIBLE} \tag{7.249}$$

where the reversible component

$$\left(\frac{dM}{dH_{\rm CORE}}\right)_{\rm REVERSIBLE} = C\frac{d(M_{\rm AN} - M)}{dH} \tag{7.250}$$

and the irreversible component

$$\left(\frac{dM}{dH_{\rm CORE}}\right)_{\rm IRREVERSIBLE} = \frac{M_{\rm AN} - M}{K} \tag{7.251}$$

where K is the pinning energy per volume and is akin to mechanical drag. M and $H_{\rm CORE}$ are found by solving (7.249) and (7.245) simultaneously.

The small signal relative permeability of the core is

$$\mu_r = \left\{\left[\left(\frac{dM}{dH_{\rm CORE}} + 1\right)F_{\rm PACK}\right]^{-1} + \frac{L_{\rm GAP}}{L_{\rm PATH}}\right\}^{-1} \tag{7.252}$$

and the flux passing through the ith winding is

$$\phi_i = \mu_0(M + H_{\rm CORE})N_i F_{\rm PACK}\,Size\,Area \tag{7.253}$$

The voltage across the ith winding is then found as

$$V_i = \frac{d\phi_i}{dt} \tag{7.254}$$

AC Analysis

For `AC` analysis the mutual inductor model is used even if a `CORE` model is specified. This allows a different coefficient of mutual coupling to be used in `AC` analysis than would otherwise be determined by nonlinear model evaluation.

Noise Analysis

The `K` element does not contribute to noise.

L Inductor

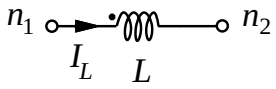


Figure 7.45: L — Inductor element.

Form

L*name* N_1 N_2 *InductorValue* [IC=I_L]

PSPICE*Form*

L*name* N_1 N_2 [*ModelName*] *InductorValue* [IC=I_L]

N_1 is the positive element node,

N_2 is the negative element node, and

ModelName is the optional model name.

InductorValue is the inductance. (Units: H; Required; Symbol: $InductorValue$)

IC is the optional initial condition specification Using IC=I_L is intended for use with the UIC option on the .TRAN line, when a transient analysis is desired with an initial current I_L through the inductor rather than the quiescent operating current. Specification of the transient initial conditions using the .IC statement (see page 66) is preferred and is more convenient.

Example

```
IND1 1 2 1.3N
IND1 1 2 1.3NH IC=1M
```

Model Type

IND PSPICE only

IND Model PSPICE Only Inductor Model

Form

.MODEL *ModelName* IND([[*keyword* = *value*] ...])

Example

```
.MODEL SMALLIND IND(L=4.5 IL1=0.1 IL2 = 0.01 TC1=0.01 TC2=0.001)
```

Model Keywords

Name	Description	Units	Default
L	inductance multiplier ($L_{\text{MULTIPLIER}}$)	-	1
IL1	linear current coefficient (I_{C1})	1/A	0
IL2	quadratic current coefficient (I_{C2})	1/A	0
TC1	linear temperature coefficient (T_{C1})	$1/°C$	0
TC2	quadratic temperature coefficient (T_{C1})	$°C^{-2}$	0

The PSpice inductance model is a nonlinear temperature dependent inductor model. It is assumed that the model parameters were determined or measured at the nominal temperature T_{NOM} (default $27°C$) specified in the most recent `.OPTIONS` statement preceding the `.MODEL` statement.

The inductance is

$$\begin{aligned} L &= InductorValue\, L_{\text{MULTIPLIER}} \left[1 + I_{L1} I_L + I_{L2} I_L^2\right] \\ &\quad \times \left[1 + T_{C1}(T - T_{\text{NOM}}) + T_{C2}(T - T_{NOM})^2\right] \end{aligned} \tag{7.255}$$

where I_L is the current flowing through the inductor as in figure 7.45 and T is the current temperature.

M MOSFET

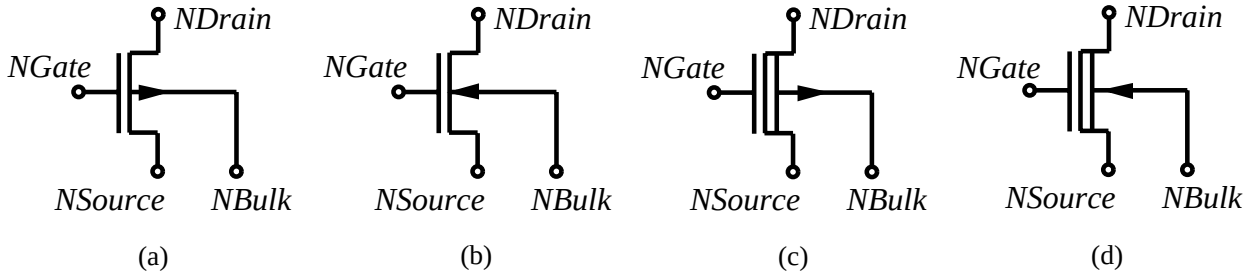


Figure 7.46: M — MOSFET element: (a) *n*-channel enhancement-mode MOSFET; (b) *p*-channel enhancement-mode MOSFET; (c) *n*-channel depletion-mode MOSFET; and (d) *p*-channel depletion-mode MOSFET.

Form

Mname *NDrain NGate NSsource NBulk ModelName* [L=*Length*] [W=*Width*]
+ [AD=*DrainDiffusionArea*] [AS=*SourceDiffusionArea*]
+ [PD=*DrainPerimeter*] [PS=*SourcePerimeter*]
+ [NRD=*RelativeDrainResistivity*] [NRS=*RelativeSourceResistivity*]
+ [OFF] [IC=V_{DS}, V_{GS}, V_{BS}]

PSpice*Form*

Mname *NDrain NGate NSsource NBulk ModelName* [L=*Length*] [W=*Width*]
+ [AD=*DrainDiffusionArea*] [AS=*SourceDiffusionArea*]
+ [PD=*DrainPerimeter*] [PS=*SourcePerimeter*]
+ [NRD=*RelativeDrainResistivity*] [NRS=*RelativeSourceResistivity*]
+ [NRG=*RelativeGateResistivity*] [NRB=*RelativeBulkResistivity*]

NDrain is the drain node.

NGate is the gate node.

NSource is the source node.

NBulk is the bulk or substrate node.

`ModelName` is the model name.

L is the channel lateral diffusion *Length*.
(Units: m; Optional; Symbol: L; The default is version dependent.
SPICE2G6 and SPICE3 Default: the length `DEFL` most recently specified in a `.OPTION` statement which in-turn defaults to 100 μm (100U); PSPICE Default: the length `L` – length specified in model *ModelName* which in turn defaults to the default length `DEFL` most recently specified in a `.OPTION` statement which in-turn defaults to 100 μm (100U).)

W is the channel lateral diffusion *Width*.
(Units: m; Optional; Symbol: W; The default is version dependent.
SPICE2G6 and SPICE3 Default: the width `DEFW` most recently specified in a `.OPTION` statement which in-turn defaults to 100 μm (100U);
PSPICE Default: the width `W` – width specified in model *ModelName* which in turn defaults to the default width `DEFW` most recently specified in a `.OPTION` statement which in-turn defaults to 100 μm (100U).)

`AD` is the area of the drain diffusion (*DrainDiffusionArea*). The default is `DEFAD` most recently specified in a `.OPTIONS` statement.
(Units: m^2; Optional; Default: `DEFAD`; Symbol: A_D)

`AS` is the area of the source diffusion (*SourceDiffusionArea*). The default is `DEFAS` most recently specified in a `.OPTIONS` statement.
(Units: m^2; Optional; Default: `DEFAS`; Symbol: A_S)

`PD` is the perimeter of the drain junction (*DrainPerimeter*).
(Units: m; Optional; Default: 0; Symbol: P_D)

`PS` is the perimeter of the source junction (*SourcePerimeter*).
(Units: m; Optional; Default: 0; symbol: P_S)

`NRD` is the relative resistivity in squares of the drain region (*RelativeDrainResistivity*). The sheet resistance `RSH` specified in the model *ModelName* is divided by this factor to obtain the parasitic drain resistance.
(Units: squares; Optional; Default: 0; Symbol: N_{RD})

`NRS` is the relative resistivity in squares of the source region (*RelativeSourceResistivity*). The sheet resistance `RSH` specified in the model *ModelName* is divided by this factor to obtain the parasitic source resistance.
(Units: squares; Optional; Default: 0; Symbol: N_{RS})

`NRG` is the relative resistivity in squares of the gate region (*RelativeGateResistivity*). The sheet resistance `RSH` specified in the model *ModelName* is divided by this factor to obtain the parasitic gate resistance.
(Units: squares; Optional; Default: 0; Symbol: N_{RG})

`NRB` is the relative resistivity in squares of the bulk (substrate) region (*RelativeBulkResistivity*). The sheet resistance `RSH` specified in the model *ModelName* is divided by this factor to obtain the parasitic bulk resistance.
(Units: squares; Optional; Default: 0; Symbol: N_{RB})

`OFF` indicates an (optional) initial condition on the device for `DC` analysis. If specified the `DC` operating point is calculated with the terminal voltages set to zero. Once convergence is obtained, the program continues to iterate to obtain the exact value of the terminal voltages. The OFF option is used to enforce the solution to correspond to a desired state if the circuit has more than one stable state.

`IC` is the optional initial condition specification. Using `IC=`V_{DS}, V_{GS}, V_{BS} is intended for use with the `UIC` option on the `.TRAN` line, when a transient analysis is desired starting from other than the quiescent operating point. Specification of the transient initial conditions using the `.IC` statement (see page 66) is preferred and is more convenient.

Example

```
M1 24 2 0 20 TYPE1
M31 2 17 6 10 MODM L=5U W=2U
M1 2 9 3 0 MOD1 L=10U W=5U AD=100P AS=100P PD=40U PS=40U
```

Note

1. The parameters of a MOSFET can be completely specified in the model *ModelName*. This facilitates the use of standard transistors by using absolute quantities in the model. Alternatively scalable process parameters can be specified in the model *ModelName* and these scaled by geometric parameters on the MOSFET element line.

2. In Spice2g6 and Spice3 the width `W` can not be specified in the model statement. For these simulators absolute device parameters must be specified in the model statement if parameters are not input on the element line.

Model Type

```
NMOS
PMOS
```

Example

```
M1 5 5 1 1 PCH L=2.0U W=20U AD=136P AS=136P
.
.
.
.MODEL PCH PMOS LEVEL=2 VTO=-0.76 GAMMA=0.6 CGSO=3.35E-10
+ CGDO=3.35E-10 CJ=4.75E-4 MJ=0.4 TOX=225E-10 NSUB=1.6E16
+ XJ=0.2E-6 LD=0 UO=139 UEXP=0 KF=5E-30 LAMBDA=0.02
```

NMOS Model N-CHANNEL MOSFET MODEL

PMOS Model P-CHANNEL MOSFET MODEL

Two groups of model parameters define the linear and nonlinear elements of the MOSFET models. One group defines absolute quantities and another group defines quantities that are multiplied by scaling parameters related to area and dimension which are specified on the element line. This enables the MOSFET element to be used in two ways. Using the absolute quantities the characteristics of a device can be defined independent of the parameters on the element line. Thus the model of a standard transistor, perhaps resident in a library, can be used without user-knowledge required. Using the scalable quantities the parameters of a fabrication process can be defined in the model statement and scaling parameters such as the lateral diffusion length (specified by `L`) and the lateral diffusion width (specified by `W`), and the drain and source diffusion areas (specfified by `AD` and `AS` specified on the element line). An example is the specification of the drain-bulk saturation current $I_{D,\mathrm{SAT}}$. This parameter can be specified by the absolute parameter I_S specified by the `IS` model keyword. It can also be determined as $I_S = J_S \cdot A_D$ using the scalable parameter J_S specified by the `JS` model keyword and A_D specified by the `AD` element keyword.

SPICE provides four MOSFET device models. The first three models, known as `LEVEL`s 1, 2 and 3 differ in the formulation of the I-V characteristic. The fourth model, known as the BSIM model, uses a completely different formulation utilizing extensive semiconductor parameters. The parameter `LEVEL` specifies the model to be used:

LEVEL = 1 → "Shichman-Hodges", MOS1
This model was the first SPICE MOSFET model and was developed in 1968 [3]. It is an elementary model and has a limited scaling capability. It is applicable to fairly large devices with gate lengths greater than 4 μm. Its main attribute is that only a few parameters need be specified and so it is good for preliminary analyses.

LEVEL = 2 → MOS2
This is an analytical model which uses a combination of processing parameters and geometry. The major development over the `LEVEL` 1 model is improved treatment of the capacitances due to the channel charge. [4–6]. The model dates from 1980 and is applicable for chanel lengths of 2 μm and higher [7].
The `LEVEL` 2 model has convergence problems and is slower and less accurate than the `LEVEL` 3 model.

LEVEL = 3 → MOS3
This is a semi-empirical model developed in 1980 [7]. It is also used for gate lengths of 2 μm and more. The parameters of this model are determined by experimental characterization and so it is more accurate than the `LEVEL` 1 and 2 models that use the more indirect process parameters.

LEVEL = 4 → BSIM
The BSIM model is an advanced empirical model which uses process information and a larger number of parameters (more than 60) to describe the operation of devices with gate lengths as short as 1 μm. It was developed in 1985 [9].

Other MOSFET models or `LEVEL`s are available in various versions of SPICE. These `LEVEL`s are optimized for MOSFETs fabricated in a particular foundary or provide a proprietary edge for the advanced commercial SPICE programs. The reader interested in more advanced MOSFET models is refered to [8].

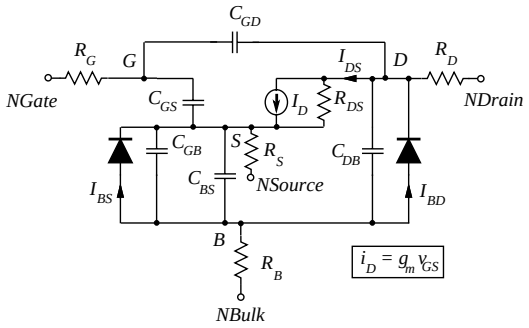


Figure 7.47: Schematic of `LEVEL` 1, 2 and 3 MOSFET models. V_{GS}, V_{DS}, V_{GD}, V_{GB}, V_{DB} and V_{BS} are voltages between the internal gate, drain, bulk and source terminals designated G, D, B and C respectively.

LEVEL 1, 2 and 3 MOSFET models.

The schematic of the LEVEEL 1, 2 and 3 MOSFET models is shown in Figure 7.47.
The LEVEL 1, 2 and 3 models have much in common. These models evaluate the junction depletion capacitances and parasitic resistances of a transistor in the same way. They differ in the procedure used to evaluate the overlap capacitances (C_{GD}, C_{GS} and C_{GB}) and that used to determine the current-voltage characteristics of the active region of a transistor. The overlap capacitances model charge storage as nonlinear thin-oxide capacitance distributed among the gate, source drain and bulk regions. These capacitances are important in describing the operation of MOSFETs. The `LEVEL` 1, 2 and 3 models are intimately intertwined as combinations of parameters can result in using equations from more than one model. The `LEVEL` parameter resolves conflicts when there is more than one way to calculate the transistor characteristics with the parameters specified by the user. Antognetti and Massobrio provide a comprehensive discussion of the development of the `LEVEL` 1, 2 and 3 models [2].

The parameters of the `LEVEL` 1, 2 and 3 models are given in table 7.9. Parameter extensions are given in table **??**. It is assumed that the model parameters were determined or measured at the nominal temperature $T_{\rm NOM}$ (default $27°C$) specified in the most recent `.OPTIONS` statement preceeding the `.MODEL` statement. In PSpice this is overwritten by the `T_MEASURED` parameter. Most of the parameters have default values. Those parameters that have INFERRED defaults are calculated from other parameters.

Table 7.9: MOSFET model keywords for `LEVEL`s 1, 2, 3.

Name	Description	Units	Default
`AF`	flicker noise exponent (A_F)	-	1
`CBD`	zero-bias B-D junction capacitance (C'_{BD})	F	0
`CBS`	zero-bias B-S junction capacitance (C'_{BS})	F	0
`CGBO`	gate-bulk overlap capacitance per meter of channel length (PARASITIC) (C_{GBO})	F/m	0
`CGDO`	gate-drain overlap capacitance per meter of channel width (PARASITIC) (C_{GDO})	F/m	0
`CGSO`	gate-source overlap capacitance per meter of channel width (PARASITIC) (C_{GSO})	F/m	0

Continued on next page

Table 7.9: MOSFET model keywords for LEVELs 1, 2, 3.

Name	Description	Units	Default
CJ	zero-bias bulk junction bottom capcitance per square meter of junction area (PARASITIC) (C_J)	F/m^2	0
CJSW	zero-bias bulk junction sidewall capacitance per meter of junction perimeter (PARASITIC) ($C_{J,SW}$)	F/m	0
DELTA	width effect on threshold voltage (LEVEL=2 and LEVEL=3) (δ)	-	0
ETA	static feedback (LEVEL=3 only) (η)	-	INFERRED
FC	coefficient for forward-bias depletion capacitance formula (PARASITIC) (F_C)	-	0.5
GAMMA	bulk threshold parameter (γ)	$V^{\frac{1}{2}}$	INFERRED
IS	bulk junction saturation current (PARASITIC) (I_S)	A	10^{-14}
JS	bulk junction saturation current per sq-meter of junction area (PARASITIC) (J_S)	A/m^2	0
KAPPA	saturation field factor (LEVEL=3 only) (κ)	-	0.2
KF	flicker noise coefficient (K_F)	-	0
KP	transconductance parameter (K_P)	A/V^2	2.10^{-5}
LAMBDA	channel-length modulation (LEVEL=1, 2 only) (λ)	1/V	0
LD	lateral diffusion (X_{JL})	m	0
LEVEL	model index	-	1
MJ	bulk junction bottom grading coefficient (PARASITIC) (M_J)	-	0.5
MJSW	bulk junction sidewall grading coefficient (PARASITIC) ($M_{J,SW}$)	-	0.33
NSUB	substrate doping (N_B)	cm^{-3}	INFERRED
NSS	surface state density (N_{SS})	cm^{-2}	INFERRED
NFS	fast surface state density (N_{FS})	cm^{-2}	0
NEFF	total channel charge (fixed and mobile) coefficient (LEVEL=2 only) (N_{EFF})	-	1
PB	bulk junction potential (ϕ_J) (This is the interface potential in the channel relative to the source at threshold.)	V	0.8
PHI	surface inversion potential ($2\phi_B$)	V	0.6
RD	drain ohmic resistance (PARASITIC) (R_D)	Ω	0
RS	source ohmic resistance (PARASITIC) (R_S)	Ω	0
RSH	drain and source diffusion sheet resistance (PARASITIC) (R_{SH})	Ω/square	0
THETA	mobility modulation (LEVEL=3 only) (θ)	1/V	0

Continued on next page

Table 7.9: MOSFET model keywords for LEVELs 1, 2, 3.

Name	Description	Units	Default
TOX	oxide thickness (T_{OX}) Default for LEVEL 2 and 3 is 0.1 μm. If LEVEL 1 and TOX is omitted then the process oriented model is not used.	m	-
TPG	type of gate material: ($T_{\rm PG}$) 1 → polysilicon, opposite type to substrate −1 → polysilicon, same type as substrate 0 → aluminum gate	-	1
UCRIT	critical field for mobility degradation (LEVEL=2 only) (U_C)	V/cm	10^4
UEXP	critical field exponent in mobility degradation (LEVEL=2 only) ($U_{\rm EXP}$)	-	0
UO	surface mobility (U-oh) (μ_0)	cm^2/V-s	600
UTRA	transverse field coefficient (mobility) (LEVEL = 1 and 3 only) ($U_{\rm TRA}$)	-	0
VMAX	maximum drift velocity of carriers ($V_{\rm MAX}$)	m/s	0
VTO	zero-bias threshold voltage N-channel devices: positive for enhancement mode and negative for depletion mode devices. P-channel devices: negative for enhancement mode and positive for depletion mode devices. (VT-oh) (V_{T0})	V	0
XJ	metallurgical junction depth (X_J)	m	0
JSSW	bulk p-n junction sidewall current per unit length (PARASITIC) ($J_{S,\rm SW}$)	A/m	0
L	channel length (L)	m	DEFL
N	bulk p-n emission coefficient (PARASITIC) (N)	-	0
PBSW	bulk p-n sidewall potential (PARASITIC) ($\phi_{J,\rm SW}$)	V	PB
RB	bulk ohmic resistance (PARASITIC) (R_B)	Ω	0
RG	gate ohmic resistance (PARASITIC) (R_B)	Ω	0
RDS	drain-source shunt resistance (R_{DS})	Ω	∞
T_ABS	($T_{\rm ABS}$)	°C	current temp.
T_MEASURED	($T_{\rm MEASURED}$)	°C	TNOM
T_REL_GLOBAL	($T_{\rm REL\_GLOBAL}$)	°C	0
T_REL_LOCAL	($T_{\rm REL\_LOCAL}$)	°C	0
TT	bulk p-n transit time (τ_T)	s	0
W	channel width (W)	m	DEFW
WD	lateral diffusion width (W_D)	m	0

Continued on next page

Table 7.9: MOSFET model keywords for `LEVEL`s 1, 2, 3.

Name	Description	Units	Default
`XQC`	fraction of channel charge attributable to drain in saturation region (X_{QC}) If $X_{QC} > 0.5$ the Meyer Capacitance Model is used. If $X_{QC} \leq 0.5$ the Ward-Dutton Capacitance Model is used.	-	1

The MOSFET `LEVEL` 1,2 and 3 parameters fall into three categories: absolute device parameters, scalable and process parameters and geometric parameters. In most cases the absolute device parameters can be derived from the scalable and process parameters and the geometry parameters. However, if specified, the values of the device parameters are used.

The physical constants used in the model evaluation are

k	Boltzman's constant	$1.3806226\,10^{-23}$ J/K
q	electronic charge	$1.6021918\,10^{-19}$ C
ϵ_0	free space permittivity	$8.854214871\,10^{-12}$ F/m
ϵ_{Si}	permittivity of silicon	$11.7\epsilon_0$
ϵ_{OX}	permittivity of silicon dioxide	$3.9\epsilon_0$
n_i	intrinsic concentration of silicon @ 300 K	$1.45\,10^{16}$ m^{-3}

Standard Calculations

Absolute temperatures (in kelvins, K) are used. The thermal voltage

$$V_{\rm TH}(T_{\rm NOM}) = \frac{kT_{\rm NOM}}{q}. \tag{7.256}$$

The silicon bandgap energy

$$E_G(T_{\rm NOM}) = 1.16 - 0.000702\frac{4T_{\rm NOM}^2}{T_{\rm NOM} + 1108}. \tag{7.257}$$

The difference of the gate and bulk contact potentials

$$\phi_{\rm MS} = \phi_{\rm GATE} - \phi_{\rm BULK}. \tag{7.258}$$

The gate contact potential

$$\phi_{\rm GATE} = \begin{cases} 3.2 & T_{\rm PG} = 0 \\ 3.25 & \text{NMOS \& } T_{\rm PG} = 1 \\ 3.25 + E_G & \text{NMOS \& } T_{\rm PG} = -1 \\ 3.25 + E_G & \text{PMOS \& } T_{\rm PG} = 1 \\ 3.25 & \text{PMOS \& } T_{\rm PG} = -1 \end{cases}. \tag{7.259}$$

The potential drop across the oxide

$$\phi_{\rm OX} = -\frac{Q_0'}{C_{\rm OX}'}. \tag{7.260}$$

The contact potential of the bulk material

$$\phi_{\rm BULK} = \begin{cases} 3.25 + E_G & \text{if NMOS} \\ 3.25 & \text{if PMOS} \end{cases}. \tag{7.261}$$

The equivalent gate oxide interface charge per unit area

$$Q_0' = qN_{\rm SS}. \tag{7.262}$$

The capacitance per unit area of the oxide is

$$C_{OX}' = \frac{\epsilon_{OX}}{T_{OX}}. \tag{7.263}$$

The effective length $L_{\rm EFF}$ of the channel is reduced by the amount X_{JL} (= `LD`) of the lateral diffusion at the source and drain regions:

$$L_{\rm EFF} = L - 2X_{JL} \tag{7.264}$$

Similarly the effective length $W_{\rm EFF}$ of the channel is reduced by the amount W_D (= `WD`) of the lateral diffusion at the edges of the channel.

$$W_{\rm EFF} = W - 2W_D \tag{7.265}$$

κ is limited: if the specified value of κ is less than or equal to zero the following parameters are set:

$$\kappa = 0.2 \tag{7.266}$$
$$\lambda = 0 \tag{7.267}$$
$$U_C = 0 \tag{7.268}$$
$$U_{\rm EXP} = 0 \tag{7.269}$$
$$U_{\rm TRA} = 0 \tag{7.270}$$

Process Oriented Model

If omitted, device parameters are computed from process parameters using defaults if necessary provided that both `TOX` $= T_{OX}$ and `NSUB` $= N_B$ are specified. If either `TOX` or `NSUB` is not specified then the critical device parameters must be specified. Which parameters are critical depends on the model `LEVEL`.

If `VTO` is not specified in the model statement then it is evaluated as

$$\texttt{VTO} = V_{T0} = \begin{cases} V_{\rm FB} + \gamma\sqrt{2\phi_B} + 2\phi_B & \text{if NMOS} \\ V_{\rm FB} - \gamma\sqrt{2\phi_B} + 2\phi_B & \text{if PMOS} \end{cases} \tag{7.271}$$

where

$$V_{\rm FB} = \phi_{\rm MS} - \phi_{\rm OX} \tag{7.272}$$

is the flat-band voltage. Otherwise if `VTO` is specified in the model statement

$$V_{\rm FB} = \begin{cases} V_{T0} - \gamma\sqrt{2\phi_B} + 2\phi_B & \text{if NMOS} \\ V_{T0} + \gamma\sqrt{2\phi_B} + 2\phi_B & \text{if PMOS} \end{cases} \tag{7.273}$$

If `GAMMA` is not specified in the model statement then

$$\texttt{GAMMA} = \gamma = \frac{\sqrt{2\epsilon_{\rm Si} q N_B}}{C'_{OX}} \tag{7.274}$$

If `PHI` is not specified in the model statement then

$$\texttt{PHI} = 2\phi_B = 2V_{\rm TH} \ln \frac{N_B}{n_i} \tag{7.275}$$

and is limited to 0.1 if calculated. $N_B =$ `NSUB` as supplied in the model statement and n_i at 300 K are used. If `KP` is not specified in the model statement then

$$\texttt{KP} = K_P = \mu_0 C'_{OX} \tag{7.276}$$

If `UCRIT` is not specified in the model statement then

$$\texttt{UCRIT} = U_C = \frac{\epsilon_{Si}}{T_{OX}} \tag{7.277}$$

$$X_d = \sqrt{\frac{2\epsilon_{\rm Si}}{qN_B}} \tag{7.278}$$

is proportional to the depletion layer widths at the source and rdain regions.

Temperature Dependence

Temperature effects are incorporated as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K).

$$V_{\rm TH} = \frac{kT}{q} \tag{7.279}$$

$$I_S(T) = I_S e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/V_{\rm TH}} \tag{7.280}$$

$$J_S(T) = J_S e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/V_{\rm TH}} \tag{7.281}$$

$$J_{S,\rm SW}(T) = J_{S,\rm SW} e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/V_{\rm TH}} \tag{7.282}$$

$$\phi_J(T) = \phi_J \frac{T}{T_{\rm NOM}} - 3V_{\rm TH}\ln\frac{T}{T_{\rm NOM}} + E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.283}$$

$$\phi_{J,\rm SW}(T) = \phi_{J,\rm SW} \frac{T}{T_{\rm NOM}} - 3V_{\rm TH}\ln\frac{T}{T_{\rm NOM}} + E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.284}$$

$$2\phi_B(T) = 2\phi_B \frac{T}{T_{\rm NOM}} - 3V_{\rm TH}\ln\frac{T}{T_{\rm NOM}} + E_G(T_{\rm NOM} - E_G(T) \tag{7.285}$$

$$C'_{BD}(T) = C'_{BD}\{1 + M_J[0.0004(T - T_{\rm NOM}) + (1 - \phi_J(T)/\phi_J)]\} \tag{7.286}$$

$$C'_{BS}(T) = C'_{BS}\{1 + M_J[0.0004(T - T_{\rm NOM}) + (1 - \phi_J(T)/\phi_J)]\} \tag{7.287}$$

$$C_J(T) = C_J\{1 + M_J[0.0004(T - T_{\rm NOM}) + (1 - \phi_{J,\rm SW}(T)/\phi_{J,\rm SW})]\} \tag{7.288}$$

$$C_{J,\rm SW}(T) = C_{J,\rm SW}\{1 + M_{J,\rm SW}[0.0004(T - T_{\rm NOM}) + (1 - \phi_J(T)/\phi_J)]\} \tag{7.289}$$

$$K_P(T) = K_P(T_{\rm NOM}/T)^{3/2} \tag{7.290}$$

$$\mu_0(T) = \mu_0(T_{\rm NOM}/T)^{3/2} \tag{7.291}$$

$$E_g(T) = 1.16 - 0.000702\frac{T^2}{T + 1108} \tag{7.292}$$

$$\tag{7.293}$$

Parasitic Resistances

The resistive parasitics R_S, R_G, R_D and R_B are treated in the same way for the `LEVEL` 1, 2 and 3 models. They may be specified as the absolute device parameters `RS`, `RG`, `RD`, and `RB` or calculated from the sheet resistivity $R_{\rm SH}$ (= `RSH`) and area parameters N_{RS} (= `NRS`), N_{RG} (= `NRG`), N_{RD} (= `NRD`) and N_{RB} (= `NRB`). As always the absolute device parameters take precedence if they are specified. Otherwise

$$R_S = N_{RS}R_{\rm SH} \tag{7.294}$$

$$R_G = N_{RG}R_{\rm SH} \tag{7.295}$$

$$R_D = N_{RD}R_{\rm SH} \tag{7.296}$$

$$R_B = N_{RB}R_{\rm SH} \tag{7.297}$$

Note that neither geometry parameters nor process parameters are required if the absolute device resistances are specified. The parasitic resistance parameter dependencies are summarized in figure 7.48.

Leakage Currents

PROCESS PARAMETERS	
RSH	$R_{\rm SH}$

+

GEOMETRY PARAMETERS	
NRS	N_{RS}
NRD	N_{RD}
NRG	N_{RG}
NRB	N_{RB}

→

DEVICE PARAMETERS	
RD	$R_D = f(R_{\rm SH}, N_{RD})$
RS	$R_S = f(R_{\rm SH}, N_{RS})$
RG	$R_B = f(R_{\rm SH}, N_{RG})$
RB	$R_B = f(R_{\rm SH}, N_{RB})$

Figure 7.48: MOSFET LEVEL 1, 2 and 3 parasitic resistance parameter relationships.

PROCESS PARAMETERS	
JS	J_S

+

GEOMETRY PARAMETERS	
AD	A_D
AS	A_S
PD	P_D
PS	P_S

→

DEVICE PARAMETERS	
IS	$I_S = f(J_S, J_{S,\rm SW}, A_D, A_S, P_D, P_S)$

Figure 7.49: MOSFET leakage current parameter dependecies.

Current flows across the normally reverse biased source-bulk and drain-bulk junctions. The bulk-source leakage current

$$I_{BS} = I_{BSS}\left(e^{(V_{BS}/V_{\rm TH})} - 1\right) \tag{7.298}$$

where

$$I_{BSS} = \begin{cases} I_S & \text{if IS specified} \\ A_S J_S + P_S J_{S,\rm SW} & \text{if IS not specified} \end{cases} \tag{7.299}$$

The bulk-drain leakage current

$$I_{BD} = I_{BDS}\left(e^{(V_{BD}/V_{\rm TH})} - 1\right) \tag{7.300}$$

where

$$I_{BDS} = \begin{cases} I_S & \text{if IS specified} \\ A_D J_S + P_S J_{S,\rm SW} & \text{if IS not specified} \end{cases} \tag{7.301}$$

The parameter dependencies of the parameters describing the leakage current in the LEVEL 1, 2 and 3 MOSFET models are summarized in figure 7.49.

Depletion Capacitances

C_{BS} and C_{BD} are the depletion capacitances at the bulk-source and bulk-drain depletion regions respectively. These depletion capacitances are calculated and used in the same way in all three (`LEVEL` = 1, 2 and 3) models. Although they may be specified as absolute device parameters they are strong functions of the voltages across the junction and are complex functions of geometry and of semiconductor doping. As such they are usually calculated from process parameters. They are the sum of component capacitances

$$C_{BS} = C_{BS,\mathrm{JA}} + C_{BS,\mathrm{SW}} + C_{BS,\mathrm{TT}} \tag{7.302}$$

where the sidewall capacitance

$$C_{BS,\mathrm{SW}} = P_S C_{J,\mathrm{SW}} C_{BSS} \tag{7.303}$$

$$C_{BSS} = \begin{cases} \left(1 - \frac{V_{BS}}{\phi_{J,\mathrm{SW}}}\right)^{-M_{J,\mathrm{SW}}} & \text{for } V_{BS} \le F_C \phi_J \\ (1-F_C)^{-(1+M_{J,\mathrm{SW}})} \left(1 - F_C(1+M_{J,\mathrm{SW}}) + \frac{M_{J,\mathrm{SW}} V_{BS}}{\phi_{J,\mathrm{SW}}}\right) & \text{for } V_{BS} > F_C \phi_J \end{cases} \tag{7.304}$$

(7.305)

the area capacitance

$$C_{BS,\mathrm{JA}} = \begin{cases} C'_{BS} C_{BSJ} & \text{if } \texttt{CBS}\ (= C'_{BS}) \text{ is specified in the model} \\ A_S C_J C_{BSJ} & \text{otherwise} \end{cases} \tag{7.306}$$

$$C_{BSJ} = \begin{cases} \left(1 - \frac{V_{BS}}{\phi_J}\right)^{-M_J} & \text{for } V_{BS} \le F_C \phi_J \\ (1-F_C)^{-(1+M_J)} \left(1 - F_C(1+M_J) + \frac{M_J V_{BS}}{\phi_J}\right) & \text{for } V_{BS} > F_C \phi_J \end{cases} \tag{7.307}$$

and the transit time capacitance

$$C_{BS,\mathrm{TT}} = \tau_T G_{BS} \tag{7.308}$$

where the bulk-source conductance $G_{BS} = \partial I_{BS} / \partial V_{BS}$ and I_{BS} is defined in (7.298).

$$C_{BD} = C_{BD,\mathrm{JA}} + C_{BD,\mathrm{SW}} \tag{7.309}$$

where the sidewall capacitance

$$C_{BD,\mathrm{SW}} = P_D C_{J,\mathrm{SW}} C_{BDS} \tag{7.310}$$

$$C_{BDS} = \begin{cases} \left(1 - \frac{V_{BD}}{\phi_{J,\mathrm{SW}}}\right)^{-M_{J,\mathrm{SW}}} & \text{for } V_{BD} \le F_C \phi_J \\ (1-F_C)^{-(1+M_{J,\mathrm{SW}})} \left(1 - F_C(1+M_{J,\mathrm{SW}}) + \frac{M_{J,\mathrm{SW}} V_{BD}}{\phi_{J,\mathrm{SW}}}\right) & \text{for } V_{BD} > F_C \phi_J \end{cases} \tag{7.311}$$

(7.312)

the area capacitance

$$C_{BD,\mathrm{JA}} = \begin{cases} C'_{BD} C_{BDJ} & \text{if } \texttt{CBD}\ (= C'_{BD}) \text{ is specified in the model} \\ A_D C_J C_{BDJ} & \text{otherwise} \end{cases} \tag{7.313}$$

$$C_{BDJ} = \begin{cases} \left(1 - \frac{V_{BD}}{\phi_J}\right)^{-M_J} & \text{for } V_{BS} \le F_C \phi_J \\ (1-F_C)^{-(1+M_J)} \left(1 - F_C(1+M_J) + \frac{M_J V_{BD}}{\phi_J}\right) & \text{for } V_{BD} > F_C \phi_J \end{cases} \tag{7.314}$$

and the transit time capacitance

PROCESS PARAMETERS	
`CJ`	C_J
`CJSW`	$C_{J,\mathrm{SW}}$
`MJ`	M_J
`MJSW`	$M_{J,\mathrm{SW}}$
`PB`	ϕ_J
`PBSW`	$\phi_{J,\mathrm{SW}}$
`FC`	F_C

+

GEOMETRY PARAMETERS	
`AD`	A_D
`AS`	A_S
`PD`	P_D
`PS`	P_S

→

DEVICE PARAMETERS	
`CBD`	$C'_{BD} = f(C_J, A_D)$
`CBS`	$C'_{BS} = f(C_J, A_S)$
$\{C_{BS}$	$= f(P_S, C'_{BS}, C_{J,\mathrm{SW}}, \tau_T$
$M_J, M_{J,\mathrm{SW}}, \phi_J, \phi_{J,\mathrm{SW}}, F_C)\}$	
$\{C_{BD}$	$=$
$f(P_D, C'_{BD}, C_{J,\mathrm{SW}}, \tau_T$	
$M_J, M_{J,\mathrm{SW}}, \phi_J, \phi_{J,\mathrm{SW}}, F_C)\}$	

Figure 7.50: MOSFET `LEVEL` 1, 2 and 3 junction depletion capacitance parameter relationships.

$$C_{BS,\mathrm{TT}} = \tau_T G_{BS} \tag{7.315}$$

where the bulk-source conductance $G_{BD} = \partial I_{BD}/\partial V_{BD}$ and I_{BD} is defined in (7.300).

In the `LEVEL` 1 MOSFET model the depletion capacitances are piecewise linear. They are calculated at the current operating point and then treated as linear. In the `LEVEL` 2 and 3 models they are treated as nonlinear. The depletion capacitance parameter dependencies are summarized in figure 7.50.

`LEVEL 1` *I/V Characteristics*

For the `LEVEL` 1 model the device parameters (other than capacitances and resistances) are evaluated using T_{OX} (`TOX`), μ_0 (`UO`), N_{SS} (`NSS` and N_B (`NSUB`) if they are not specified in the `.MODEL` statement.

The `LEVEL 1` current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \geq 0$)

The regions are as follows:

cutoff region:	$V_{GS} < V_T$
linear region:	$V_{GS} > V_T$ and $V_{DS} < V_{GS} - V_T$
saturation region:	$V_{GS} > V_T$ and $V_{DS} > V_{GS} - V_T$

where the threshold voltage

$$V_T = \begin{cases} V_{\mathrm{FB}} + 2\phi_B + \gamma\sqrt{2\phi_B - V_{BS}} & V_{BS} \geq 2\phi_B \\ V_{\mathrm{FB}} + 2\phi_B & V_{BS} < 2\phi_B \end{cases} \tag{7.316}$$

Then

$$I_D = \begin{cases} 0 & \text{cutoff region} \\ \dfrac{W_{\mathrm{EFF}}}{L_{\mathrm{EFF}}} \dfrac{K_P}{2}(1 + \lambda V_{DS}) V_{DS} \left[2(V_{GS} - V_T) - V_{DS}\right] & \text{linear region} \\ \dfrac{W_{\mathrm{EFF}}}{L_{\mathrm{EFF}}} \dfrac{K_P}{2}(1 + \lambda V_{DS}) \left[V_{GS} - V_T\right]^2 & \text{saturation region} \end{cases} \tag{7.317}$$

Inverted Mode: ($V_{DS} < 0$)

PROCESS PARAMETERS	
NSUB	N_B
TOX	T_{OX}
NSS	N_{SS}
UO	μ_0

+

GEOMETRY PARAMETERS	
–	
Required	
L	L
W	W
Optional	
LD	X_{JL}
WD	W_D

→

DEVICE PARAMETERS	
VTO	$V_{T0} = f(2\phi_B, N_{SS}, T_{OX}, \gamma)$
KP	$K_P = f(\mu_0, T_{OX})$
LAMBDA	λ
PHI	$2\phi_B = f(N_B)$
GAMMA	$\gamma = f(T_{OX}, N_B)$
	$\{I_D = f(W, L, W_D, X_{JL}$ $V_{T0}, K_P, \lambda, 2\phi_B, \gamma)\}$

Figure 7.51: LEVEL 1 I/V dependencies.

PROCESS PARAMETERS
–

+

GEOMETRY PARAMETERS	
Required	
W	W
Optional	
WD	W_D

→

DEVICE PARAMETERS	
Overlap Capacitances	
CGSO	C_{GSO}
CGDO	C_{GDO}
CGBO	C_{GBO}
	$\{C_{GS} = f(C_{GSO}, W, W_D)\}$
	$\{C_{GD} = f(C_{GDO}, W, W_D)\}$
	$\{C_{GB} = f(C_{GBO}, W, W_D)\}$

Figure 7.52: MOSFET LEVEL 1 overlap capacitance parameter relationships.

In the inverted mode the MOSFET I/V characteristics are evaluated as in the normal mode (7.317) but with the drain and source subscripts interchanged. The relationships of the parameters describing the I/V characteristics for the LEVEL 1 model are summarized in figure 7.51.

LEVEL *1 Overlap Capacitances*

In the LEVEL 1 model the gate overlap capacitances C_{GS}, C_{GD} and C_{GB} are constant and are calculated using the per unit width overlap capacitances C_{GSO} (CGSO), C_{GDO} (CGDO) and C_{GBO} (CGBO):

$$C_{GS} = C_{GSO}W \tag{7.318}$$

$$C_{GD} = C_{GDO}W \tag{7.319}$$

$$C_{GB} = C_{GBO}W \tag{7.320}$$

The overlap capacitance parameter dependencies are summarized in Figure 7.52.

`LEVEL 2` *I/V Characteristics*

The `LEVEL 2` I/V characteristics are based on empirical fits resulting in a more accurate description of the I/V response than obtained with the `LEVEL 1` model. The `LEVEL 2` current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.
Normal Mode: ($V_{DS} \geq 0$)

The regions are as follows:

cutoff region:	$V_{GS} < V_T$
weak inversion region:	$V_T < V_{GS} \leq V_{\rm ON}$
linear region (strong inversion):	$V_{GS} > V_{\rm ON}$ and $V_{DS} < V_{DS,\rm SAT}$
saturation region (strong inversion):	$V_{GS} > V_{\rm ON}$ and $V_{DS} > V_{DS,\rm SAT}$

where

$$V_T = V'_{\rm FB} + \gamma_{\rm EFF} X_S \tag{7.321}$$

$$V_{\rm ON} = \begin{cases} V_T & N_{\rm FS} = 0 \\ V_T + V_{\rm TH} x_n & N_{\rm FS} \neq 0 \end{cases} \tag{7.322}$$

$$X_S = \begin{cases} \dfrac{\sqrt{2\phi_B}}{[1 + \frac{1}{2} V_{BS}/(2\phi_B)]} & V_{BS} > 0 \\ \sqrt{2\phi_B - V_{BS}} & V_{BS} \leq 0 \end{cases} \tag{7.323}$$

where

$$x_n = 1 + F_N - \gamma_{\rm EFF} X_1 - X_2 X_S + W_{\rm EFF} L_{\rm EFF} \frac{q N_{\rm FS}}{C'_{OX}} \tag{7.324}$$

$$\eta = 1 + F_N \tag{7.325}$$

the effect of channel width on threshold voltage is modeled by

$$V'_{\rm FB} = V_{\rm FB} + F_N (2\phi_B - V_{BS}) \tag{7.326}$$

and the flat band voltage, $V_{\rm FB}$, is calculated using (7.272) or (7.273).

$$V_{GST} = V_{GS} - V_{ON} \tag{7.327}$$

The factor describing the effect of channel width on threshold is

$$F_N = \frac{\epsilon_s \delta \pi}{4 C'_{OX} W_{\rm EFF}} \tag{7.328}$$

The effective bulk threshold parameter is affected by charge in the drain and source depletion regions. This is important for short channels. The factor describing short channel effects is

$$\gamma_{\rm EFF} = \begin{cases} \gamma & \gamma \leq 0 \text{ or } N_B \leq 0 \\ \gamma(1 - F_{DD} - F_{SD}) & \gamma > 0 \text{ and } N_B > 0 \end{cases} \tag{7.329}$$

where the effect of depletion charge at the drain is described by

$$F_{DD} = \begin{cases} \frac{1}{2}\left(\sqrt{1+2X_D X_B}-1\right) & V_{DS} \le V_{DS,\mathrm{SAT}} \\ \frac{1}{2}\left(\sqrt{1+2X_D X_{B,\mathrm{SAT}}}-1\right)\frac{X_J}{L_{\mathrm{EFF}}} & V_{DS} > V_{DS,\mathrm{SAT}} \end{cases} \tag{7.330}$$

the effect of depletion charge at the source is described by

$$F_{SD} = \frac{1}{2}\left(\sqrt{1+2X_D X_S}-1\right)\frac{X_J}{L_{\mathrm{EFF}}} \tag{7.331}$$

and

$$X_B = \begin{cases} \frac{\sqrt{2\phi_B}}{[1+\frac{1}{2}(V_{BS}-V_{DS})/(2\phi_B)]} & V_{DS} < V_{BS} \\ \sqrt{2\phi_B + V_{DS} - V_{BS}} & V_{DS} \le V_{BS} \end{cases} \tag{7.332}$$

and for saturation

$$X_{B,\mathrm{SAT}} = \begin{cases} \frac{\sqrt{2\phi_B}}{[1+\frac{1}{2}(V_{BS}-V_{DS,\mathrm{SAT}})/(2\phi_B)]} & V_{DS,\mathrm{SAT}} < V_{BS} \\ \sqrt{2\phi_B + V_{DS,\mathrm{SAT}} - V_{BS}} & V_{DS,\mathrm{SAT}} \ge V_{BS} \end{cases} \tag{7.333}$$

X_S is evaluated using (7.323) and X_D using (7.402).

$$X_1 = \begin{cases} \frac{-X_S^2}{2(2\phi_B)^{(3/2)}} & V_{BS} > 0 \\ -\frac{1}{2X_S} & V_{BS} \le 0 \end{cases} \tag{7.334}$$

and

$$X_2 = \begin{cases} -\gamma\frac{1}{2}\frac{X_D X_1}{L_{\mathrm{EFF}} X_S} & X_J > 0 \\ 0 & X_J \le 0 \end{cases} \tag{7.335}$$

The effective mobility due to modulation by the gate

$$\mu_{\mathrm{EFF}} = \begin{cases} \mu_0\left(\frac{U_C}{V_{GS}-V_{\mathrm{ON}}}\right)^{U_{\mathrm{EXP}}} & C_{\mathrm{OX}} \ne 0 \text{ and } (V_{GS}-V_{ON}) > U_C \\ \mu_0 & C_{\mathrm{OX}} = 0 \text{ or } (V_{GS}-V_{ON}) \le U_C \end{cases} \tag{7.336}$$

and the factor describing this effect

$$F_G = \frac{\mu_{\mathrm{EFF}}}{\mu_0} \tag{7.337}$$

The channel shortening factor

$$F_D = \frac{L_{\mathrm{EFF}}}{L'_{EFF}} \tag{7.338}$$

where the effective channel length due to channel shortening is

$$L'_{\mathrm{EFF}} = \begin{cases} \frac{X_{WB}}{1+(\Delta_L - L_{\mathrm{EFF}} + X_{WB})/X_{WB}} & (1-\lambda V_{DS})L_{\mathrm{EFF}} < X_{WB} \\ (1-\lambda V_{DS})L_{\mathrm{EFF}} & (1-\lambda V_{DS})L_{\mathrm{EFF}} \ge X_{WB} \end{cases} \tag{7.339}$$

The expression for L'_{EFF} when $(1-\lambda V_{DS})L_{\rm EFF} < X_{WB}$ limits chanel shortening at punch-through. In (7.339)

$$\Delta_L = \lambda V_{DS} L_{\rm EFF} \tag{7.340}$$

and the distance that the depletion region at the drain extends into the channel is

$$X_{WB} = \begin{cases} X_D\sqrt{\phi_J} & N_B \neq 0 \\ 0.25\,10^{-6} & N_B = 0 \end{cases} \tag{7.341}$$

The pinch-off voltage

$$V_P = \begin{cases} MAX((V_{GSX} - V'_{\rm FB}), 0) & \frac{\gamma_{\rm EFF}}{\eta} \leq 0 \\ V_T & \frac{\gamma_{\rm EFF}}{\eta} > 0 \text{ and } X_V \leq 0 \\ V_T & \frac{\gamma_{\rm EFF}}{\eta} > 0 \text{ and } X_3 < 0 \\ V'_{\rm FB} + \eta V_{DS} + \sqrt{X_3} & \frac{\gamma_{\rm EFF}}{\eta} > 0 \text{ and } X_3 \geq 0 \text{ and } X_V > 0 \end{cases} \tag{7.342}$$

where

$$X_3 = (V_{DS} + 2\phi_B - V_{BS}) \left(\frac{\gamma_{\rm EFF}}{\eta}\right)^2 \tag{7.343}$$

The drain-source saturation voltage

$$V_{DS,\rm SAT} = \begin{cases} V_P \text{ (Grove Frohman approximation)} & V_{\rm MAX} \leq 0 \\ V_{DS,\rm SAT} \text{ from Baum's theory of velocity saturation} & V_{\rm MAX} > 0 \end{cases} \tag{7.344}$$

In Baum's theory of velocity saturation the saturation voltage, $V_{DS,\rm SAT}$, is the solution of the quartic equation

$$aV^4_{DS,\rm SAT} + bV^3_{DS,\rm SAT} + cV^2_{DS,\rm SAT} + dV_{DS,\rm SAT} = 0 \tag{7.345}$$

where

$$a = \frac{3}{4}\frac{\gamma_{\rm EFF}}{\eta} \tag{7.346}$$

$$b = -2(V_1 + V_X) \tag{7.347}$$

$$c = -2\frac{\gamma_{\rm EFF}}{\eta} V_X \tag{7.348}$$

$$d = 2V_1(V_2 + V_X) - V_2^2 - \frac{3}{4}\frac{\gamma_{\rm EFF}}{\eta} X_S^3 \tag{7.349}$$

$$V_1 = \frac{(V_{GSX} - V'_{FB})}{\eta} + 2\phi_B - V_{BS} \tag{7.350}$$

$$V_2 = 2\phi_B - V_{BS} \tag{7.351}$$

$$V_X = \frac{V_{\rm MAX} L_{\rm EFF}}{\mu_{\rm EFF}} \tag{7.352}$$

The body effect factor is

$$F_B = X_B^3 - X_S^3 \tag{7.353}$$

and the body effect factor in saturation is

$$F_{B,\rm EFF} = X^3_{B,\rm SAT} - X_S^3 \tag{7.354}$$

cutoff region

$$I_D = 0 \tag{7.355}$$

PROCESS PARAMETERS	
DELTA	δ
LAMBDA	λ
NSS	$N_{\rm SS}$
NSUB	N_B
PB	ϕ_J
TOX	T_{OX}
UCRIT	U_C
UEXP	$U_{\rm EXP}$
VMAX	$V_{\rm MAX}$
UO	μ_0
XJ	X_J

+

GEOMETRY PARAMETERS	
Required	
L	L
W	W
Optional	
LD	X_{JL}
WD	W_D

→

DEVICE PARAMETERS	
I/V	
VTO	$V_{T0} = f(2\phi_B, N_{\rm SS}, T_{OX})$
KP	$K_P = f(\mu_0, T_{OX})$
PHI	$2\phi_B = f(N_B)$
GAMMA	$\gamma = f(\mu_0, T_{OX}, N_B)$
	$\{I_D = f(W, L, W_D, X_{JL},$
	$K_P, 2\phi_B, \mu_0, \gamma, V_{T0}, \phi_J$
	$V_{\rm MAX}, X_J, \lambda, U_{\rm EXP}, U_C, \delta)\}$

Figure 7.53: MOSFET LEVEL 2 I/V parameter relationships.

linear region

$$I_D = K_P \frac{W_{\rm EFF}}{L_{\rm EFF}} F_G F_D \left[\left(V_{GS} - V'_{\rm FB} - \frac{\eta}{2} V_{DS} \right) V_{DS} - \frac{3}{2} \frac{\gamma_{\rm EFF}}{\eta} F_B \right] \tag{7.356}$$

weak inversion region

When V_{GS} is slightly above V_T, I_D increases slowly over a few thermal voltages $V_{\rm TH}$ in exponential manner becoming I_D calculated for strong inversion. This effect is handled empirically by defining two exponential which, as well as ensuring an exponential increase in I_D, also ensure that the transconductance G_M ($= \partial I_D / \partial V_{GS}$) is continuous at $V_{GS} = V_{\rm ON}$.

$$I_D = \begin{cases} I_{D,\rm ON} \left[\frac{10}{11} e^{(V_{GS} - V_{\rm ON})/(x_n V_{TH})} + \frac{1}{11} e^{\alpha (V_{GS} - V_{\rm ON})} \right] & \alpha > 0 \\ I_{D,\rm ON} e^{(V_{GS} - V_{\rm ON})/(x_n V_{TH})} & \alpha \le 0 \end{cases} \tag{7.357}$$

where

$$\alpha = 11 \left(\frac{G_{M,\rm ON}}{I_{D,\rm ON}} - \frac{1}{x_n V_{\rm TH}} \right) \tag{7.358}$$

$$G_{M,\rm ON} = \frac{\partial I_{D,\rm ON}}{\partial V_{GS}} \tag{7.359}$$

$$I_{D,\rm ON} = \begin{cases} I_D \text{ in (7.356) with } V_{GS} = V_{ON} & V_{DS} \le V_{DS,\rm SAT} \\ I_D \text{ in (7.361) with } V_{GS} = V_{ON} \text{ and } V_{DS} = V_{DS,\rm SAT} & V_{DS} \le V_{DS,\rm SAT} \end{cases} \tag{7.360}$$

saturation region

$$I_D = \frac{L_{\rm EFF}}{L_{\rm EFF - \Delta_L}} I_{D,\rm SAT} \tag{7.361}$$

$$I_{D,\rm SAT} = K_P \frac{W_{\rm EFF}}{L_{\rm EFF}} F_G F_D \left[\left(V_{GS} - V'_{\rm FB} - \frac{\eta}{2} V_{DS,\rm SAT} \right) V_{DS,\rm SAT} - \frac{3}{2} \frac{\gamma_{\rm EFF}}{\eta} F_{B,\rm SAT} \right] \tag{7.362}$$

The LEVEL 3 current-voltage parameter dependencies are summarized in figure 7.53.

LEVEL *2 Overlap Capacitances*

In the LEVEL 2 model the gate overlap capacitances asre strong functions of voltage. Two overlap capacitance models are available in PSPICEthe Meyer model based on the model originally proposed by Meyer [4] and the Ward-Dutton model [5,6]. SPICE2G6and SPICE3use just the Meyer model. The Meyer and Ward-Dutton models differ in the derivation of the channel charge.

LEVEL *2 Meyer Model*

This model is selected when the parameter XQC $= X_{QC}$ is not specified or $X_{QC} < 0.5$.

The voltage dependent thin-oxide capacitances are used only if T_{OX} is specified in the model statement. Four operating regions are defined in the Meyer model:

accumulation region:	$V_{GS} < V_{\rm ON} - 2\phi_B$
depletion region:	$V_{\rm ON} - 2\phi_B < V_{GS} < V_{\rm ON}$
saturation region:	$V_{\rm ON} < V_{GS} < V_{\rm ON} + V_{DS}$
linear region:	$V_{GS} > V_{\rm ON} + V_{DS}$

where

$$V_{\rm ON} = \begin{cases} V_T + x_n V_{\rm TH} & \text{if } N_{\rm FS} = \texttt{NFS} \text{ specified} \\ V_T & \text{if } N_{\rm FS} = \texttt{NFS} \text{ not specified} \end{cases} \tag{7.363}$$

$$V_T = V_{T0} + \gamma\left[\sqrt{2\phi_B - V_{BS}} - \sqrt{2\phi_B}\right] \tag{7.364}$$

$$x_n = 1 + \frac{qN_{\rm FS}}{C'_{OX}} + \frac{C_D}{C'_{OX}} \tag{7.365}$$

$$C'_{OX} = \frac{\epsilon_{OX}}{T_{OX}} \tag{7.366}$$

$$C_{OX} = C'_{OX} W_{\rm EFF} L_{\rm EFF} \tag{7.367}$$

$$C_D = \frac{\gamma}{2\sqrt{2\phi_B - V_{BS}}} \tag{7.368}$$

$$C_{GS} = \begin{cases} C_{GSO}W & \text{accumulation region} \\ \frac{2}{3}C_{OX}\left(1 + \frac{V_{\rm ON} - V_{GS}}{2\phi_B}\right) + C_{GSO}W_{\rm EFF} & \text{depletion region} \\ \frac{2}{3}C_{OX} + C_{GSO}W_{\rm EFF} & \text{saturation region} \\ C_{OX}\left\{1 - \left[\frac{V_{GS} - V_{DS} - V_{\rm ON}}{2(V_{GS} - V_{\rm ON}) - V_{DS}}\right]^2\right\} + C_{GSO}W_{\rm EFF} & \text{linear region} \end{cases} \tag{7.369}$$

$$C_{GD} = \begin{cases} C_{GDO}W_{\rm EFF} & \text{accumulation region} \\ C_{GDO}W_{\rm EFF} & \text{depletion region} \\ C_{GDO}W_{\rm EFF} & \text{saturation region} \\ C_{OX}\left\{1 - \left[\frac{V_{GS} - V_{\rm ON}}{2(V_{GS} - V_{\rm ON}) - V_{DS}}\right]^2\right\} + C_{GDO}W_{\rm EFF} & \text{linear region} \end{cases} \tag{7.370}$$

$$C_{GB} = \begin{cases} C_{OX} + C_{GBO}L_{\rm EFF} & \text{accumulation region} \\ C_{OX}\left(\frac{V_{\rm ON} - V_{GS}}{2\phi_B}\right) + C_{GBO}L_{\rm EFF} & \text{depletion region} \\ C_{GBO}L_{\rm EFF} & \text{saturation region} \\ C_{GBO}L_{\rm EFF} & \text{linear region} \end{cases} \tag{7.371}$$

LEVEL *2 Ward-Dutton Model* PSPICE *only*

This model is selected when the parameter `XQC` is specified and less than 0.5. The charge in the gate Q_G and the substrate Q_B is calculated and the difference of these is taken as the channel charge $Q_{\rm CHANNEL}$. This charge is then partitioned and allocated between the source as Q_S and the drain Q_D as follows:

$$\begin{aligned} Q_{\rm CHANNEL} &= Q_D + Q_S && (7.372)\\ Q_D &= X_{QC} Q_{\rm CHANNEL} && (7.373)\\ & && (7.374)\end{aligned}$$

so that $Q_S = (1 - X_{QC})Q_{\rm CHANNEL}$. This partitioning is somewhat arbitrary but produces transient results that more closely match measurements than does the Meyer capacitance model. However this is at the price of poorer convergence properties and sometimes error. This is particularly so when V_{DS} is changing sign.

Two operating regions are defined in the Ward-Dutton model:

$$\begin{array}{ll} \text{off region:} & V_{GS} \le V_T \\ \text{on region:} & V_{GS} > V_T \end{array}$$

where

$$\begin{aligned} V_T &= V_{T0} + \gamma\left[\sqrt{2\phi_B - V_{BS}} - \sqrt{2\phi_B}\right] && (7.375)\\ C_{OX} &= C'_{OX} W_{\rm EFF} L_{\rm EFF} && (7.376)\\ C'_{OX} &= \frac{\epsilon_{OX}}{T_{OX}} && (7.377)\\ & && (7.378)\end{aligned}$$

In the charge evaluations the following terms are used:

$$\begin{aligned} v_G &= v_{GB} - V'_{FB} + 2\phi_B && (7.379)\\ v_D &= \begin{cases} 2\phi_B - v_{BD} & v_{BD} > 2\phi_B \\ 0 & v_{BD} \le 2\phi_B \end{cases} && (7.380)\\ v_S &= \begin{cases} 2\phi_B - v_{BS} & v_{BS} > 2\phi_B \\ 0 & v_{BS} \le 2\phi_B \end{cases} && (7.381)\\ v_E &= \begin{cases} v_D & v_D < v_{DS,\rm SAT} \\ v_{DS,\rm SAT} & v_D \ge v_{DS,\rm SAT} \end{cases} && (7.382)\\ x_5 &= (v_E + vs)(\sqrt{v_E} + \sqrt{v_S}) && (7.383)\\ x_6 &= ((v_E^2 + v_S^2) + v_E v_S) + \sqrt{v_E}\sqrt{v_S}(v_E + v_S) && (7.384)\\ D &= v_G(\sqrt{v_E} + \sqrt{v_S}) - \gamma_{\rm EFF}((v_E + v_S) + \sqrt{v_E}\sqrt{v_S})/1.5 \\ & \quad -.5(\sqrt{v_E} + \sqrt{v_S})(v_E + v_S) && (7.385)\end{aligned}$$

where V'_{FB} is defined in (7.326) and $V_{DS,\rm SAT}$ in (7.344).

off region

$$\begin{aligned} Q_G &= \begin{cases} \gamma_{\rm EFF} C_{OX}(\sqrt{\frac{1}{4}\gamma_{\rm EFF}^2 + v_G} - \gamma_{\rm EFF}/2) & v_G > 0 \\ C_{OX} v_G & v_G \le 0 \end{cases} && (7.386)\\ Q_B &= -Q_G && (7.387)\\ Q_{\rm CHANNEL} &= -(Q_G + Q_B) && (7.388)\end{aligned}$$

on region

PROCESS PARAMETERS	
CJ	C_J
CJSW	$C_{J,\mathrm{SW}}$
MJ	M_J
MJSW	$M_{J,\mathrm{SW}}$
PB	ϕ_J
PBSW	ϕ_J
FC	ϕ_J

+

GEOMETRY PARAMETERS	
AD	A_D
AS	A_S
PD	A_D
PS	A_S

→

DEVICE PARAMETERS	
Constant Overlap Capacitances	
CGSO	C_{GSO}
CGDO	C_{GDO}
CGBO	C_{GBO}

Figure 7.54: MOSFET LEVEL 2 overlap capacitance parameter relationships.

$$Q_B = -\gamma_{\mathrm{EFF}} C_{OX} \left(\frac{vg\frac{2}{3}((v_E + v_S) + \sqrt{v_E}\sqrt{v_S}) - \frac{1}{2}\gamma_{\mathrm{EFF}} x_5 - .4x_6}{D} \right) \tag{7.389}$$

$$Q_G = C_{OX} \left(vg - \frac{.5vgx_5 - .4\gamma_{\mathrm{EFF}} x_6 - ((v_E^2 + v_S^2) + v_E v_S)(\sqrt{v_E} + \sqrt{v_S})/3}{D} \right) \tag{7.390}$$

where V'_{FB} is defined in (7.326). The overlap capacitances are then evaluated as

$$C_{GDB} = \partial Q_G / \partial v_D \tag{7.391}$$

$$C_{GSB} = \partial Q_G / \partial v_S \tag{7.392}$$

$$C_{GGB} = \partial Q_G / \partial v_G \tag{7.393}$$

$$C_{BDB} = \partial Q_B / \partial v_D \tag{7.394}$$

$$C_{BSB} = \partial Q_B / \partial v_S \tag{7.395}$$

$$C_{BGB} = \partial Q_B / \partial v_G \tag{7.396}$$

The LEVEL 2 overlap capacitance parameter dependencies are summarized in figure 7.54.

`LEVEL 3` *I/V Characteristics*

The `LEVEL 3` I/V characteristics are based on empirical fits resulting in a more accurate description of the I/V response than obtained with the `LEVEL 2` model. The `LEVEL 3` current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \geq 0$)

The regions are as follows:

cutoff region:	$V_{GS} < V_T$
weak inversion region:	$V_T < V_{GS} \leq V_{\rm ON}$
linear region (strong inversion):	$V_{GS} > V_{\rm ON}$ and $V_{DS} < V_{DS,\rm SAT}$
saturation region (strong inversion):	$V_{GS} > V_{\rm ON}$ and $V_{DS} > V_{DS,\rm SAT}$

where

$$V_T = V_{T0} - \sigma V_{DS} + F_C \tag{7.397}$$

the effect of short and narrow channel on threshold voltage

$$F_C = \gamma F_S \sqrt{2\phi_B - V_{BS}} + F_N(2\phi_B - V_{BS}) \tag{7.398}$$

and

$$V_{\rm ON} = \begin{cases} V_T & NFS = 0 \\ V_T + V_{TH} x_n & NFS \neq 0 \end{cases} \tag{7.399}$$

The effect of the short channel is described by

$$F_S = 1 - \frac{X_J}{L_{\rm EFF}} \left(\frac{X_{JL} + W_C}{X_J} \sqrt{1 - \frac{W_P}{X_J + W_P}} - \frac{X_{JL}}{X_J} \right) \tag{7.400}$$

where

$$W_P = X_D \sqrt{\phi_J - V_{BS}} \tag{7.401}$$

$$X_D = \sqrt{\frac{2\epsilon_s}{qN_B}} \tag{7.402}$$

$$W_C = X_J \left[0.0831353 + 0.8013929 \frac{W_P}{X_J} + 0.0111077 \frac{W_P}{X_J} \right] \tag{7.403}$$

and

$$\sigma = \eta \frac{8.15^{-22}}{C'_{OX} L^3_{\rm EFF}} \tag{7.404}$$

The effect of channel width on threshold is

$$F_N = \frac{\epsilon_s \delta \pi}{4 C'_{OX} W_{\rm EFF}} \tag{7.405}$$

The effective mobility due to modulation by the gate

$$\mu_S = \mu_0 F_G \tag{7.406}$$

and the factor describing mobility modulation by the gate is

$$F_G = \frac{1}{1 + \theta(V_{GSX} - V_T)} \tag{7.407}$$

where

$$V_{GSX} = \begin{cases} V_{GS} & V_{GS} < V_{\rm ON} \\ V_{\rm ON} & V_{GS} \geq V_{\rm ON} \end{cases} \tag{7.408}$$

The drain-source saturation voltage

$$V_{DS,\mathrm{SAT}} = \begin{cases} V_A + V_B - \sqrt{V_A^2 + V_B^2} & V_{\mathrm{MAX}} > 0 \\ V_P & V_{\mathrm{MAX}} \leq 0 \end{cases} \tag{7.409}$$

where

$$V_A = \frac{V_{GSX} - V_T}{1 + F_B} \tag{7.410}$$

$$V_B = \frac{v_{\mathrm{MAX}} L_{\mathrm{EFF}}}{\mu_S} \tag{7.411}$$

$$V_P = V_{GSX} - V_T \tag{7.412}$$

The body effect factor

$$F_B = \frac{\gamma F_S}{4\sqrt{\phi_{BS}}} + F_N \tag{7.413}$$

where

$$\phi_{BS} = \begin{cases} 2\phi_B - V_{BS} & V_{BS} \leq 0 \\ \dfrac{2\phi_B}{\sqrt{1 + \frac{1}{2} V_{BS}/(2\phi_B)}} & V_{BS} > 0 \end{cases} \tag{7.414}$$

The velocity saturation factor is

$$F_D = \begin{cases} \dfrac{1}{1 + V_{DS}/V_B} & \text{for} V_{\mathrm{MAX}} \neq 0 \\ 1 & \text{for} V_{\mathrm{MAX}} = 0 \end{cases} \tag{7.415}$$

cutoff region

$$I_D = 0 \tag{7.416}$$

linear region

$$I_D = K_P \frac{W_{\mathrm{EFF}}}{L_{\mathrm{EFF}}} F_G F_D \left[V_{GSX} - V_T - \frac{1 + F_{\mathrm{B}}}{2} V_{DS} \right] V_{DS} \tag{7.417}$$

weak inversion region

When V_{GS} is slightly above V_T, I_D increases slowly over a few thermal voltages V_{TH} in exponential manner becoming I_D calculated for strong inversion. This effect is handled empirically by defining two exponential which, as well as ensuring an exponential increase in I_D, also ensure that the transconductance G_M ($= \partial I_D / \partial V_{GS}$) is continuous at $V_{GS} = V_{\mathrm{ON}}$.

$$I_D = \begin{cases} I_{D,\mathrm{ON}} \left[\frac{10}{11} e^{(V_{GS} - V_{\mathrm{ON}})/(x_n V_{TH})} + \frac{1}{11} e^{\alpha (V_{GS} - V_{\mathrm{ON}})} \right] & \alpha > 0 \\ I_{D,\mathrm{ON}} e^{(V_{GS} - V_{\mathrm{ON}})/(x_n V_{TH})} & \alpha \leq 0 \end{cases} \tag{7.418}$$

where

$$\alpha = 11 \left(\frac{G_{M,\mathrm{ON}}}{I_{D,\mathrm{ON}}} - \frac{1}{x_n V_{\mathrm{TH}}} \right) \tag{7.419}$$

$$G_{M,\mathrm{ON}} = \frac{\partial I_{D,\mathrm{ON}}}{\partial V_{GS}} \tag{7.420}$$

and

$$I_{D,\mathrm{ON}} = \begin{cases} I_D \text{ in (7.417) with } V_{GS} = V_{ON} & V_{DS} \leq V_{DS,\mathrm{SAT}} \\ I_D \text{ in (7.422) with } V_{GS} = V_{ON} \text{ and } V_{DS} = V_{DS,\mathrm{SAT}} & V_{DS} \leq V_{DS,\mathrm{SAT}} \end{cases} \tag{7.421}$$

PROCESS PARAMETERS	
KAPPA	κ
NSS	$N_{\rm SS}$
NSUB	N_B
PB	ϕ_J
THETA	θ
TOX	T_{OX}
UO	μ_0
VMAX	$V_{\rm MAX}$
XJ	X_J

+

GEOMETRY PARAMETERS	
Required	
L	L
W	W
Optional	
LD	X_{JL}
WD	W_D

→

DEVICE PARAMETERS	
I/V	
VTO	$V_{T0} = f(2\phi_B, N_{\rm SS}, T_{OX})$
KP	$K_P = f(\mu_0, T_{OX})$
PHI	$2\phi_B = f(N_B)$
GAMMA	$\gamma = f(\mu_0, T_{OX}, N_B)$
$\{I_D = f(W, L, W_D, X_{JL}$	
$K_P, 2\phi_B, N_B, T_{OX}, \mu_0, \theta$	
$\gamma, V_{T0}, \phi_J, V_{\rm MAX}, X_J, \kappa, \eta)\}$	

Figure 7.55: MOSFET LEVEL 3 I/V parameter relationships.

saturation region

$$I_D = \frac{L_{\rm EFF}}{L_{{\rm EFF}-\Delta_L}} I_{D,{\rm SAT}} \tag{7.422}$$

$$I_{D,{\rm SAT}} = K_P \frac{W_{\rm EFF}}{L_{\rm EFF}} F_G F_D \left[V_{GSX} - V_T - \frac{1+F_{\rm B}}{2} V_{DS,{\rm SAT}} \right] V_{DS,{\rm SAT}} \tag{7.423}$$

The reduction in the channel length due to V_{DS} modulation is

$$\Delta_L = \begin{cases} \Delta'_L & \Delta'_L < \frac{1}{2} L_{\rm EFF} \\ L_{\rm EFF} - \frac{L_{\rm EFF}}{\Delta'_L} & \Delta'_L \geq \frac{1}{2} L_{\rm EFF} \end{cases} \tag{7.424}$$

where the punch through approximation is used for $\Delta'_L \geq \frac{1}{2} L_{\rm EFF}$¿ In (7.424) the distance that the depletion region at the drain extends into the channel is

$$\Delta'_L = \sqrt{\left(\frac{E_P X_D^2}{2} \right)^2 + \kappa X_D^2 (V_{DS} - V_{DS,{\rm SAT}})} - \frac{E_P X_D^2}{2} \tag{7.425}$$

and

$$E_P = \frac{I_{D,{\rm SAT}}}{G_{DS,{\rm SAT}} L_{\rm EFF}} \tag{7.426}$$

Here

$$G_{DS,{\rm SAT}} = \frac{\partial I_{D,{\rm SAT}}}{\partial V_{DS,{\rm SAT}}} \tag{7.427}$$

The LEVEL 3 current-voltage parameter dependencies are summarized in Figure 7.55.

LEVEL *3 Overlap Capacitances*

In the LEVEL 3 model the gate overlap capacitances asre strong functions of voltage. Two overlap capacitance models are available the Meyer model based on the model originally proposed by Meyer [4] and the Ward-Dutton model [5, 6]. These models differ in the derivation of the channel charge.

LEVEL *3 Meyer Model*

This model is selected when the parameter `XQC` $= X_{QC}$ is not specified or $X_{QC} < 0.5$.

The voltage dependent thin-oxide capacitances are used only if T_{OX} is specified in the model statement. Four operating regions are defined in the Meyer model:

accumulation region:	$V_{GS} < V_{\rm ON} - 2\phi_B$
depletion region:	$V_{\rm ON} - 2\phi_B < V_{GS} < V_{\rm ON}$
saturation region:	$V_{\rm ON} < V_{GS} < V_{\rm ON} + V_{DS}$
linear region:	$V_{GS} > V_{\rm ON} + V_{DS}$

where

$$V_{\rm ON} = \begin{cases} V_T + x_n V_{\rm TH} & \text{if } N_{\rm FS} = \texttt{NFS} \text{ specified} \\ V_T & \text{if } N_{\rm FS} = \texttt{NFS} \text{ not specified} \end{cases} \tag{7.428}$$

$$V_T = V_{T0} + \gamma\left[\sqrt{2\phi_B - V_{BS}} - \sqrt{2\phi_B}\right] \tag{7.429}$$

$$x_n = 1 + \frac{qN_{\rm FS}}{C'_{OX}} + \frac{C_D}{C'_{OX}} \tag{7.430}$$

$$C'_{OX} = \frac{\epsilon_{OX}}{T_{OX}} \tag{7.431}$$

$$C_{OX} = C'_{OX} W_{\rm EFF} L_{\rm EFF} \tag{7.432}$$

$$C_D = \frac{\gamma}{2\sqrt{2\phi_B - V_{BS}}} \tag{7.433}$$

$$C_{GS} = \begin{cases} C_{GSO}W & \text{accumulation region} \\ \frac{2}{3}C_{OX}\left(1 + \frac{V_{\rm ON} - V_{GS}}{2\phi_B}\right) + C_{GSO}W_{\rm EFF} & \text{depletion region} \\ \frac{2}{3}C_{OX} + C_{GSO}W_{\rm EFF} & \text{saturation region} \\ C_{OX}\left\{1 - \left[\frac{V_{GS} - V_{DS} - V_{\rm ON}}{2(V_{GS} - V_{\rm ON}) - V_{DS}}\right]^2\right\} + C_{GSO}W_{\rm EFF} & \text{linear region} \end{cases} \tag{7.434}$$

$$C_{GD} = \begin{cases} C_{GDO}W_{\rm EFF} & \text{accumulation region} \\ C_{GDO}W_{\rm EFF} & \text{depletion region} \\ C_{GDO}W_{\rm EFF} & \text{saturation region} \\ C_{OX}\left\{1 - \left[\frac{V_{GS} - V_{\rm ON}}{2(V_{GS} - V_{\rm ON}) - V_{DS}}\right]^2\right\} + C_{GDO}W_{\rm EFF} & \text{linear region} \end{cases} \tag{7.435}$$

$$C_{GB} = \begin{cases} C_{OX} + C_{GBO}L_{\rm EFF} & \text{accumulation region} \\ C_{OX}\left(\frac{V_{\rm ON} - V_{GS}}{2\phi_B}\right) + C_{GBO}L_{\rm EFF} & \text{depletion region} \\ C_{GBO}L_{\rm EFF} & \text{saturation region} \\ C_{GBO}L_{\rm EFF} & \text{linear region} \end{cases} \tag{7.436}$$

LEVEL *3 Ward-Dutton Model*

This model is selected when the parameter `XQC` is specified and less than 0.5. The charge in the gate Q_G and the substrate Q_B is calculated and the difference of these is taken as the channel charge $Q_{\rm CHANNEL}$. This charge is then partitioned and allocated between the source as Q_S and the drain Q_D as follows:

$$\begin{aligned} Q_{\rm CHANNEL} &= Q_D + Q_S && (7.437)\\ Q_D &= X_{QC} Q_{\rm CHANNEL} && (7.438)\\ & && (7.439) \end{aligned}$$

so that $Q_S = (1 - X_{QC})Q_{\rm CHANNEL}$. This partitioning is somewhat arbitrary but produces transient results that more closely match measurements than does the Meyer capacitance model. However this is at the price of poorer convergence properties and sometimes error. This is particularly so when V_{DS} is changing sign. Two operating regions are defined in the Ward-Dutton model:

off region: $V_{GS} \le V'_T$
on region: $V_{GS} > V'_T$

where

$$\begin{aligned} V'_T &= v_{BIX} + F_C && (7.440)\\ v_{BIX} &= V_{FB} - \sigma V_{DS} && (7.441) \end{aligned}$$

F_C is defined in (7.398), σ in (7.404) and V_{FB} is defined in (7.272) or (7.272). off region

$$Q_G = \begin{cases} \gamma F_S C_{\rm OX} \left[\sqrt{\left(\frac{\gamma F_S}{2}\right)^2 + (v_{GB} - V_{FB} + 2\phi_B)} - \frac{\gamma F_S}{2} \right] & \\ \qquad\qquad v_{GB} > (V_{FB} - 2\phi_B) & \\ C_{\rm OX}(v_{GB} - V_{FB} + 2\phi_B) \qquad v_{GB} \le (V_{FB} - 2\phi_B) & \end{cases} \quad (7.442)$$

$$Q_B = -Q_G \quad (7.443)$$

on region

PROCESS PARAMETERS	
NSUB	N_B
CJ	C_J
CJSW	$C_{J,\mathrm{SW}}$
MJ	M_J
MJSW	$M_{J,\mathrm{SW}}$
PB	ϕ_J
PBSW	$\phi_{J,\mathrm{SW}}$
FC	F_C

+

GEOMETRY PARAMETERS	
AD	A_D
AS	A_S
PD	P_D
PS	P_S

→

DEVICE PARAMETERS	
Constant Overlap Capacitances	
CGSO	C_{GSO}
CGDO	C_{GDO}
CGBO	C_{GBO}

Figure 7.56: MOSFET LEVEL 3 overlap capacitance parameter relationships.

$$Q_G = \begin{cases} C_{\mathrm{OX}}(V_{GS} - V_{FB}) & V_{DSX} = 0 \\ C_{\mathrm{OX}}(V_{GS} - v_{BIX} - \frac{1}{2}V_{DSX} + x_a) & V_{DSX} \neq 0 \end{cases} \tag{7.444}$$

$$Q_B = \begin{cases} -C_{\mathrm{OX}}F_C & V_{DSX} = 0 \\ -C_{\mathrm{OX}}(F_C + \frac{1}{2}F_B V_{DSX} - x_a F_B) & V_{DSX} \neq 0 \end{cases} \tag{7.445}$$

where

$$x_a = \frac{(1+F_B)V_{DSX}^2}{12V_{GSX} - V_T' - \frac{1}{2}(1+F_B)V_{DSX}} \tag{7.446}$$

$$Q_{\mathrm{CHANNEL}} = -(Q_G + Q_B) \tag{7.447}$$

where F_B is defined in (7.413), V_{GSX} in (7.408) and

$$V_{DSX} = \begin{cases} V_{DS,\mathrm{SAT}} & V_{DS} > V_{D,\mathrm{SAT}} \\ V_{DS} & V_{DS} \leq V_{D,\mathrm{SAT}} \end{cases} \tag{7.448}$$

The overlap capacitances are then evaluated as

$$C_{GDB} = \partial Q_G/\partial v_D \tag{7.449}$$

$$C_{GSB} = \partial Q_G/\partial v_S \tag{7.450}$$

$$C_{GGB} = \partial Q_G/\partial v_G \tag{7.451}$$

$$C_{BDB} = \partial Q_B/\partial v_D \tag{7.452}$$

$$C_{BSB} = \partial Q_B/\partial v_S \tag{7.453}$$

$$C_{BGB} = \partial Q_B/\partial v_G \tag{7.454}$$

The overlap capacitance parameter dependencies are summarized in figure 7.56.

BSIM (LEVEL 4) MOSFET models.

The parameters of the BSIM (LEVEL 4 model are all values obtained from process characterization, and can be generated automatically. J. Pierret [10] describes a means of generating a 'process' file, and the program Proc2Mod provided in the UC Berkeley standard SPICE3 distribution converts this file into a sequence of .MODEL lines suitable for inclusion in a SPICE circuit file. Parameters marked below with an * in the L/W column also have corresponding parameters with a length and width dependency.

Unlike most other models 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.

Table 7.10: SPICE BSIM (level 4) parameters.

Name	Description	Units	Default	L/W
A0	drain saturation current for $V_{GS} = 0$ (LEVEL=4) (VERSION: SOMEVERSIONSOFSPICE) (A_0)	A	0.1	
A1	coefficient of V_1 (primary transconductance parameter)			
CGBO	gate-bulk overlap capacitance per meter channel length (PARASITIC) (C_{GBO})	F/m	REQUIRED	
CGDO	gate-drain overlap capacitance per meter channel width (PARASITIC) (C_{GDO})	F/m	REQUIRED	
CGSO	gate-source overlap capacitance per meter channel width (PARASITIC) (C_{GSO})	F/m	REQUIRED	
CJ	source-drain junction capacitance per unit area (PARASITIC) (C_J)	F/m^2	REQUIRED	
CJSW	source-drain junction sidewall capacitance per unit length (PARASITIC) ($C_{J,SW}$)	F/m	REQUIRED	
DL	shortening of channel (Δ_L)	μm	REQUIRED	
DW	narrowing of channel (Δ_W)	μm	REQUIRED	
DELL	source-drain junction length reduction ($DELL$)	m	REQUIRED	
ETA	zero-bias drain-induced barrier lowering coefficient (η)	-	REQUIRED	*
JS	source-drain junction current density (J_S)	A/m^2	REQUIRED	
K1	body effect coefficient (K_1)	$V^{\frac{1}{2}}$	REQUIRED	*
K2	drain/source depletion charge sharing coefficient (K_2)	-	REQUIRED	*
MJ	grading coefficient of source-drain junction (M_J)	-	REQUIRED	
MJSW	grading coefficient of source-drain junction sidewall ($M_{J,SW}$)	-	REQUIRED	
MUS	mobility at zero substrate bias and at $V_{DS} = V_{DD}$ (μ_S)	cm^2/V^2s	REQUIRED	*
MUZ	zero-bias mobility (μ_Z)	cm^2/Vs	REQUIRED	
NO	zero-bias subthreshold slope coefficient (N-zero) (N_0)	-	REQUIRED	*
NB	sensitivity of subthreshold slope to substrate bias (PARASITIC) (N_B)	-	REQUIRED	*
ND	sensitivity of subthreshold slope to drain bias (N_D)	-	REQUIRED	*

Continued on next page

Table 7.10: SPICE BSIM (level 4) parameters.

Name	Description	Units	Default	L/W
PB	built in potential of source/ drain junction (ϕ_J)	V	REQUIRED	
PBSW	built in potential of source/ drain juntion sidewall (PARASITIC) ($\phi_{J,\mathrm{SW}}$)	V	REQUIRED	
PHI	surface inversion potential ($2\phi_B$)	V	REQUIRED	*
RSH	drain and source diffusion sheet resistance (PARASITIC) (R_{SH})	Ω-square	REQUIRED	
TEMP	temperature at which parameters were measured (T)	C	REQUIRED	
TOX	gate oxide thickness (T_{OX})	μm	REQUIRED	
U0	zero-bias transverse-field mobility degradation coefficient (U-zero (U_0)	V^{-1}	REQUIRED	*
U1	zero-bias velocity saturation coefficient (U_1)	μm/V	REQUIRED	*
VDD	measurement bias range (V_{DD})	V	REQUIRED	
WDF	source-drain junction default width (W_{DF})	m	REQUIRED	
VFB	flat-band voltage (V_{FB})	V	REQUIRED	*
X2E	sensitivity of drain-induced barrier lowering effect to substrate bias (X_{2E})	V^{-1}	REQUIRED	*
X2MS	sensitivity of mobility to substrate bias at $V_{DS} = V_{DD}$ ($X_{2\mathrm{MS}}$)	$\mathrm{cm}^2/\mathrm{V}^2\mathrm{s}$	REQUIRED	*
X2MZ	sensitivity of mobility to substrate bias at $V_{DS} = 0$ ($X_{2\mathrm{MZ}}$)	$\mathrm{cm}^2/\mathrm{V}^2\mathrm{s}$	REQUIRED	*
X2U0	sensitivity of transverse field mobility degradation effect to substrate bias (X2U-zero) ($X_{2\mathrm{U0}}$)	V^{-2}	REQUIRED	*
X2U1	sensitivity of velocity saturation effect to substrate bias ($X_{2\mathrm{U1}}$)	$\mu\mathrm{m}\mathrm{V}^{-2}$	REQUIRED	*
X3E	sensitivity of drain-induced barrier lowering effect to drain bias at $V_{DS} = V_{DD}$ (X_{3E})	V^{-1}	REQUIRED	*
X3MS	sensitivity of mobility to drain bias at V_{DS}= V_{DD} ($X_{3\mathrm{MS}}$)	$\mathrm{cm}^2/\mathrm{V}^2\mathrm{s}$	REQUIRED	*
X3U1	sensitivity of velocity saturation effect on drain bias at V_{DS} = V_{DD} ($X_{3\mathrm{U1}}$)	$\mu\mathrm{V}^{-2}$	REQUIRED	*
XPART	gate oxide capacitance charge partition model flag (X_{PART}) XPART = 0: selects a 40:60 drain:source charge partition XPART = 1: selects a 0:100 drain:source charge partition	-	REQUIRED	

*Table 7.11 continued: BSIM (*LEVEL*) 4 model keywords, extensions.*

Table 7.11: SPICE BSIM (level 4) parameters, extensions.

Name	Description	Units	Default
AF	flicker noise exponent (A_F)	-	1
CBD	zero-bias B-D junction capacitance (C'_{BD})	F	0
CBS	zero-bias B-S junction capacitance (C'_{BS})	F	0
FC	coefficient for forward-bias depletion capacitance formula (F_C)	-	0.5
IS	bulk junction saturation current (I_S)	A	1E-14
KF	flicker noise coefficient (K_F)	-	0
L	channel length (L)	m	DEFL
N	bulk p-n emission coefficient (PARASITIC) (N)	-	0
PBSW	bulk p-n sidewall potential (PARASITIC) $(\phi_{J,\mathrm{SW}})$	V	PB
JSSW	bulk junction sidewall current per unit length (PARASITIC) $(J_{S,\mathrm{SW}})$	A/m	0
RB	bulk ohmic resistance (PARASITIC) (R_B)	Ω	0
RD	drain ohmic resistance (PARASITIC) (R_D)	Ω	0
RDS	drain-source shunt resistance (R_{DS})	Ω	0
RG	gate ohmic resistance (PARASITIC) (R_B)	Ω	0
RS	source ohmic resistance (PARASITIC) (R_S)	Ω	0
TT	bulk p-n transit time (τ_T)	s	0
W	channel width (W)	m	DEFL

AC Analysis

The AC analysis uses the model of figure 7.46 with the capacitor values evaluated at the DC operating point with

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \tag{7.455}$$

and

$$R_{DS} = \frac{\partial I_{DS}}{\partial V_{DS}} \tag{7.456}$$

Noise Analysis

The MOSFET noise model accounts for thermal noise generated in the parasitic resistances and shot and flicker noise generated in the drain source current generator. The rms (root-mean-square) values of thermal noise current generators shunting the four parasitic resistance R_B, R_D, R_G and R_S are

$$I_{n,B} = \sqrt{4kT/R_B}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.457}$$

$$I_{n,D} = \sqrt{4kT/R_D}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.458}$$

$$I_{n,G} = \sqrt{4kT/R_G}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.459}$$

$$I_{n,S} = \sqrt{4kT/R_S}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.460}$$

The rms value of noise current generators in series with the drain-source current generator

$$I_{n,DS} = \left(I^2_{\mathrm{SHOT},DS} + I^2_{\mathrm{FLICKER},DS}\right)^{1/2} \tag{7.461}$$

$$I_{\mathrm{SHOT},DS} = \sqrt{4kTg_m\frac{2}{3}} \quad \mathrm{A}/\sqrt{\mathrm{Hz}}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.462}$$

$$I_{\mathrm{FLICKER},DS} = \sqrt{\frac{K_F I_D^{A_F}}{fK_{\mathrm{CHANNEL}}}} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.463}$$

where the transconductance

$$g_m = \frac{\partial I_D}{\partial V_{GS}} \tag{7.464}$$

is evaluated at the `DC` operating point, and

$$K_{\mathrm{CHANNEL}} = \frac{\partial L^2_{\mathrm{EFF}}\epsilon_{\mathrm{Si}}}{\partial T_{OX}} \tag{7.465}$$

N Digital Input Interface

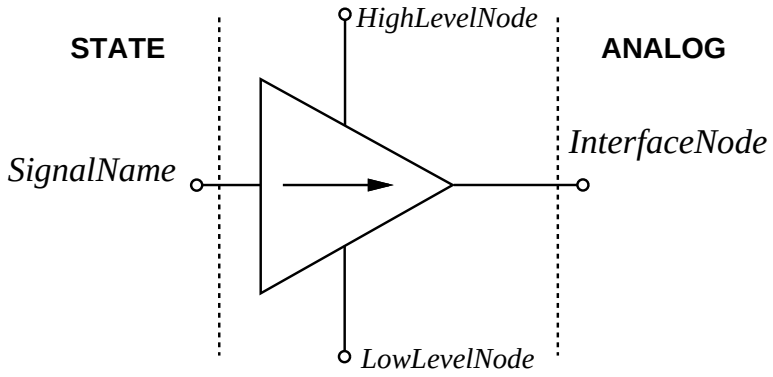


Figure 7.57: N — Digital input interface element. Converts from a digital (state) signal to an analog signal.

Form

N*name InterfaceNode LowLevelNode HighLevelNode ModelName* [SIGNAME = *DigitalSignalName*] [IS = *InitialState*]

InterfaceNode Identifier of node interfacing between digital signal and continuous time circuit.

LowLevelNode Identifier of low level reference node. Normally this is the logic "zero" voltage.

HighLevelNode Identifier of high level reference node. Normally this is the logic "one" voltage.

ModelName Name of the model specifying transitions times and resistances and capacitances of each logic state

SIGNAME Keyword for digital signal name. (optional)

DigitalSignalName Digital signal name. *DigitalSignalName* is the name of the signal specified in the input file specified in the element model. If it is omitted then *DigitalSignalName* defaults element name N*name* stripped of the prefix N (i.e. *name*).

IS Keyword for initial state. (optional)

InitialState Integer speciftying the initial state. If specified, it must be 0, 1, ..., or 19. This over rides the state specified at TIME=0 in the digital input file (see the model specification). The state of the digital interface input (N) element remains as the *InitialState* state until a state (other than the state at TIME=0 is input from the specified file.

Example

```
N100 1 0 2 INTERFACE_FROM_REGISTER SIGNAME=REG1 IS=0
NCONTROL 1 0 2 CONTROL
```

Model Type

DINPUT

The digital input interface is modeled by time variable resistances between the *Interface Node* and the *Low Level Node* and between the the *Interface Node* and the *High Level Node*. The variable resistances are shunted by fixed capacitances. The resistance are controlled by parameters specified in the model *ModelName*. The resistance varies exponentially from the old state to the new state over the time period indicated for the new state. This approximates the output of a digital gate.

DINPUT Model — Digital Input Interface Model

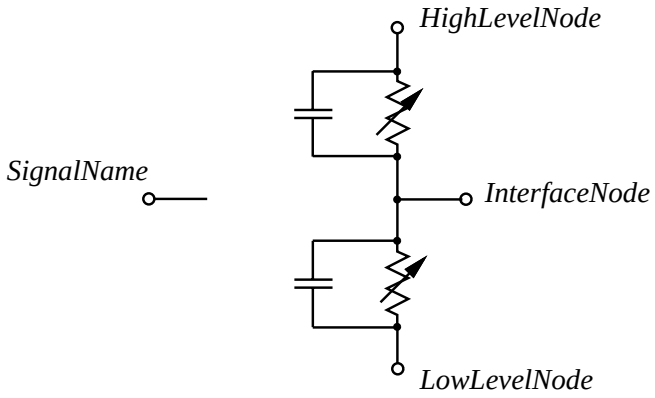


Figure 7.58: Digital input interface model.

The digital input interface is modeled by time variable resistances between the *Interface Node* and the *Low Level Node* and between the the *Interface Node* and the *High Level Node*. The variable resistances are shunted by fixed capacitances. The parameters are controlled by parameters specified in the model. Upon a state transition the two resistances vary exponentially from the old state to the new state over the time period indicated for the new state. This approximates the output of a digital gate. The sequence of states and the state change times are specified in the file specified by the `FILE` = (*InputFileName*) keyword in the model. The initial state at `TIME` 0 is taken from this file unless the `IS` keyword is specified on the element line. In the I_S (= *InitialState*) keyword is specified then the state of the digital input interface is *InitialState* until a state transition at `TIME` > 0 is specified in the file *InputFileName*.

Keywords:

Name	Description	Units	Default
FILE	digital input filename. If more than one model refers to the same file then the filenames specified must be identical and not logicly equivalent. This ensures that the file is opened only once.	-	REQUIRED
FORMAT	digital input file format	-	1
TIMESTEP	digital input file time step	s	1NS
CLO	capacitance to low level node	F	0
CHI	capacitance to high level node	F	0
SnNAME	state "n" character abreviation n = 0, 2, ..., or 19	-	REQUIRED
SnTSW	state "n" switching time n = 0, 2, ..., or 19	s	REQUIRED
SnRLO	state "n" resistance to low level node n = 0, 2, ..., or 19	Ω	REQUIRED
SnRHI	state "n" resistance to high level node n = 0, 2, ..., or 19	Ω	REQUIRED

O Digital Output Interface

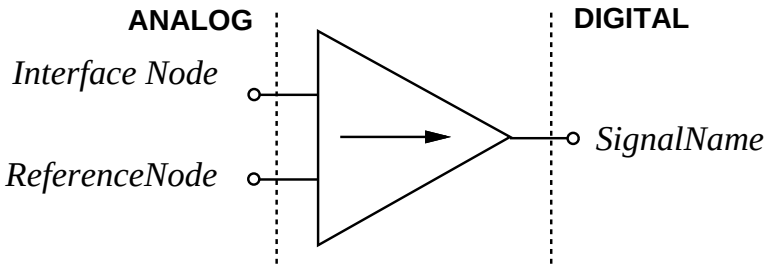


Figure 7.59: O — Digital output interface element.

Form

O*name InterfaceNode ReferenceNode ModelName* [SIGNAME = *DigitalSignalName*]

InterfaceNode Identifier of node interfacing between digital signal and continuous time circuit.

ReferenceNode Identifier of reference node. Normally this is ground

ModelName Name of the model specifying transistions times and resistances and capacitances of each logic state.

SIGNAME Keyword for digital signal name. (optional)

DigitalSignalName Digital signal name.

Example

```
O100 1 0 INTERFACE_TO_MEMORY SIGNAME=MEM1
OADD1 1 0 2 ADD1
```

Model Type

DOUTPUT

The digital output interface is modeled by time variable resistances between the *Interface Node* and the *Low Level Node* and between the the *Interface Node* and the *High Level Node*. The variable resistances are shunted by fixed capacitances. The parameters are controlled by parameters specified in the model. The resistance varies exponentially from the old state to the new state over the time period indicated for the new state. This approximates the output of a digital gate.

DOUTPUT Model Digital Output Interface Model

Keywords:

Name	Description	Units	Default
FILE	digital output filename. If more than one model refers to the same file then the filenames specified must be identical and not logicly equivalent. This ensures that the file is opened only once.	-	REQUIRED
FORMAT	digital output file format	-	1
TIMESTEP	digital output file time step	s	1NS
TIMESCALE	digital output file time scale	s	1
CHGONLY	Output type flag: = 0 → output at each TIMESTEP = 1 → output only on state change	-	0
CLOAD	capacitance	F	0
RLOAD	resistance	Ω	1000
S*n*NAME	state "n" character abreviation n = 0, 2, ..., or 19	-	REQUIRED
S*n*VLO	state "n" low level voltage n = 0, 2, ..., or 19	s	REQUIRED
S*n*VHI	state "n" high level voltage n = 0, 2, ..., or 19	s	REQUIRED

The digital output interface is modeled by a resistance *RLoad* and capacitance *CLoad* between the *InterfaceNode* and the *Reference Node* . The values of *Rload* and *CLoad* are specified in the model *ModelName*.

A state transistion from state n (n = one of 0, 1, 2, ... 19) is indicated if the interface voltage $V_{InterfaceNode} - V_{ReferenceNode}$ between the *InterfaceNode* and the *ReferenceNode* node is outside the range S*n*VHI − S*n*VLO. If there is a state transistion then the valid voltage range of each state k is considered in order from state k = 0 to state 19 to determine which voltage range S*k*VHI − S*k*VLO brackets the current interface voltage $V_{InterfaceNode} - V_{ReferenceNode}$. The first valid state becomes the new state. If there is no valid state then the new state is indeterminate and designated by "?". At each TIME being a multiple integer of TIMESTEP a line is written to the digital output file *OutputFileName*. If the new state at the time $t_i = i \cdot$ TIMESTEP is n then the i th line is:

$$int(i \cdot \texttt{TIMESCALE})n$$

where int() is the integer operation. An example of the first few lines of *OutputFileName* with a TIMESTEP of 1 ns and TIMESCALE of 2 is: 0.012042638?101120141

P Port Element

Some versions of spice only.

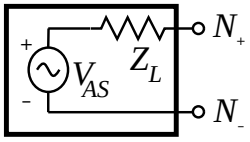


Figure 7.60: P — port element.

Form

P*name* N_+ N_- `PNR=` *PortNumber* [`ZL=` *ReferenceImpedance*]

N_+ is the positive element node,

N_- is the negative element node, and

PNR is the integer index of the port. The port index must be numbered sequentially beginning at 1. That is, the first occurrence of a `P` element in the in input netlist must have `PNR=1`, the second occurrence `PNR=2`, etc. (Units: none; Required; Symbol: *PortNumber*;)

ZL is the reference impedance of port (Units: Ω; Optional; Default: 50 Ω; Symbol: Z_L;)

Example

```
PORT1 1 0 PNR=1 ZL=75
```

Note

1. V_{AS} in Fig. 7.60 is not visible to the user and is used by the program to test for the S parameters. As an example of using the port specification with a source consider the partial circuit in Fig. 7.61

. The spice code defining this is *Example*

P*name* N_+ N_- `PNR=` *PortNumber* [`ZL=` *ReferenceImpedance*]
[`VIN` N_- `0 PULSE (Pulse Specification)`]

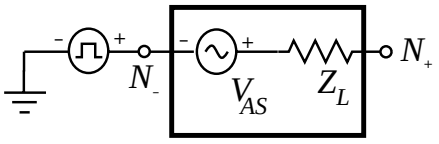


Figure 7.61: Example of the usage of a P element with a pulse voltage source.

Q Bipolar Junction Transistor

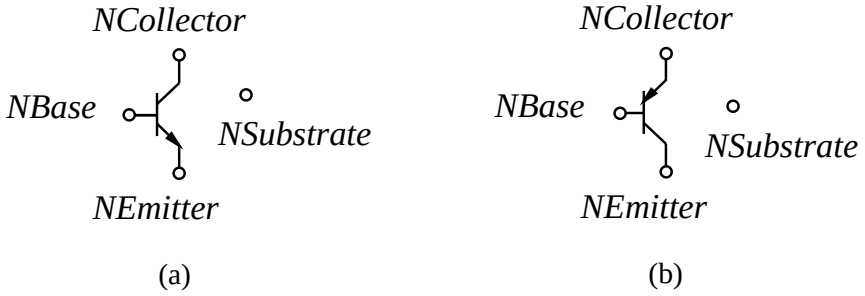


Figure 7.62: Q — bipolar junction transistor element: (a) NPN transistor; (b) PNP transistor.

Form

Q*name NCollector NBase NEmitter* [*NSubstrate*] *ModelName* [*Area*] [OFF]
+ [IC=*Vbe,Vce*]

NCollector is the collector node.

NBase is the base node.

NEmitter is the emitter node.

NSubstrate is the optional substrate node. If not specified, ground is used as the substrate node. If *NSubstrate* is a name as allowed in PSpice) it must be enclosed in square brackets, e.g. [*NSubstrate*], to distinguish it from *ModelName*.

ModelName is the model name.

Area is the area factor
If the area factor is omitted, a value of 1.0 is assumed. (Units: none; Optional; Default: 1; Symbol: *Area*)

OFF indicates an (optional) initial condition on the device for the DC analysis. If specified the DC operating point is calculated with the terminal voltages set to zero. Once convergence is obtained, the program continues to iterate to obtain the exact value of the terminal voltages. The OFF option is used to enforce the solution to correspond to a desired state if the circuit has more than one stable state.

IC is the optional initial condition specification using IC=V_{BE}, V_{CE} is intended for use with the UIC option on the .TRAN line, when a transient analysis is desired starting from other than the quiescent operating point. See the *.IC* line description for a better way to set transient initial conditions.

Example

```
Q20 10 50 0 QFAST IC=0.65,15.0
Q5PUSH 10 29 14 200 MODEL1
```

NPN Model		NPN Si Bipolar Transistor Model
PNP Model		PNP Si Bipolar Transistor Model
LPNP Model	PSPICE Only	Lateral PNP Si Bipolar Transistor Model

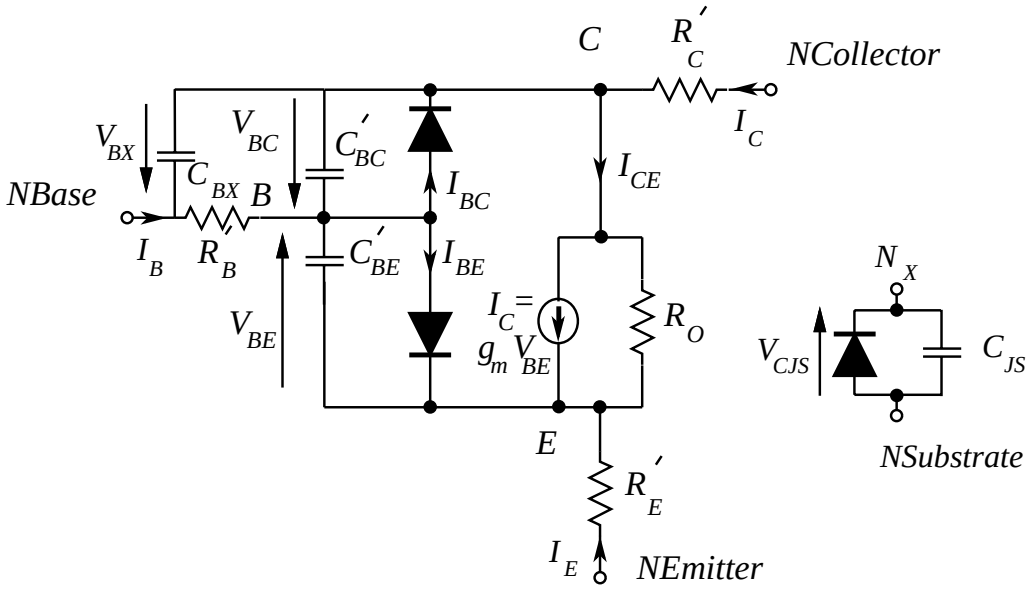


Figure 7.63: Schematic of the NPN bipolar junction transistor model. In the NPN and PNP models node N_X is connected to node C In the LPNP model node N_X is connected to node B.

The NPN and PNP BJT models are identical but with the positive sense of currents and voltages opposite so that the model parameters are always positive. The LPNP model is used for a lateral PNP IC transistor structure. In the NPN and PNP models the node N_X in figure 7.63 is connected to node C — the internal collector node. In the LPNP model node N_X is connected to node B — the internal base node. Only the model type designated on the element line distinguishes which schematic is used.

The bipolar junction transistor model in SPICE is based on the charge control model of Gummel and Poon. Extensions in the SPICE implementation deal with effects at high bias levels. The model reduces to the simpler Ebers-Moll model with the ommission of appropriate model parameters.

Table 7.12: BJT model parameters.

Name	Description	Units	Default	Are
AF	flicker noise exponent (A_F)	-	1	
BF	ideal maximum forward beta (β_F)	-	100	

Continued on next page

Table 7.12: BJT model parameters.

Name	Description	Units	Default	Are
BR	ideal maximum reverse beta (β_R)	-	1	
C2	alternative keyword for ISE PSPICE only.			
C4	alternative keyword for ISC PSPICE only.			
CCS	alternative keyword for CJS PSPICE only.			
CJC	base-collector zero-bias depletion capacitance (C_{JC})	F	0	*
CJE	base-emitter zero-bias depletion capacitance (C_{JE})	F	0	*
CJS	zero-bias collector-substrate capacitance (C_{JS})	F	0	*
EG	energy gap voltage (barrier height) (E_G)	eV	1.11	
FC	coefficient for forward-bias depletion capacitance formula (F_C)	-	0.5	
IK	alternative keyword for IKF PSPICE only.			
IKF	corner of forward beta high current roll-off (I_{KF})	A	∞	*
IKR	corner of reverse beta high current roll-off (I_KF)	A	∞	*
IRB	current where base resistance falls halfway to its minimum value (I_{RB})	-	∞	*
IS	transport saturation current (I_S)	A	1.0E-16	*
ISC	base-collector leakage saturation current (I_{SC}) If ISC is greater than 1 it is treated as a multiplier. In this case $I_{SC} =$ ISC I_S	A	0	*
ISE	base-emitter leakage saturation current (I_{SE}) If ISE is greater than 1 it is treated as a multiplier. In this case $I_{SE} =$ ISE I_S	A	0	*
ISS	substrate p-n junction saturation current (I_{SS}) PSPICE only.	A	0	*
ITF	high-current parameter for effect on TF ($I_{\tau F}$)	A	0	*
KF	flicker-noise coefficient (K_F)	-	0	
MC	alternative keyword for MJC PSPICE only.			
ME	alternative keyword for MJE PSPICE only.			
MJC	base-collector junction exponential factor (M_{JC})	-	0.33	
MJE	base-emitter junction exponential factor (M_{JE})	-	0.33	
MJS	substrate junction exponential factor (M_{JS})	-	0	

Continued on next page

Table 7.12: BJT model parameters.

Name	Description	Units	Default	Are
MS	alternative keyword for MJS PSPICE only.			
NC	base-collector leakage emission coefficient (N_C)	-	2	
NE	base-emitter leakage emission coefficient (N_E)	-	1.5	
NF	forward current emission coefficient (N_F)	-	1.0	
NR	reverse current emission coefficient (N_R)	-	1	
NS	substrate p-n emission coefficient (N_S) PSPICE only.	-	1	*
PC	alternative keyword for VJC PSPICE only.			
PE	alternative keyword for VJE PSPICE only.			
PS	alternative keyword for VJS PSPICE only.			
PT	alternative keyword for XTI PSPICE only.			
PTF	excess phase at frequency=1.0/(TF 2π) Hz ($P_{\tau F}$)	degree	0	
RB	zero bias base resistance (R_B)	Ω	0	*
RBM	minimum base resistance at high currents (R_{BM})	Ω	RB	*
RC	collector resistance (R_C)	Ω	0	*
RE	emitter resistance (R_E)	Ω	0	*
TF	ideal forward transit time (τ_F)	s	0	
TR	ideal reverse trarsit time (τ_R)	s	0	
TRB1	RB linear temperature coefficient (T_{RB1}) PSPICE only.	$^\circ C^{-1}$	1	*
TRB2	RB quadratic temperature coefficient (T_{RB2}) PSPICE only.	$^\circ C^{-2}$	1	*
TRC1	RC linear temperature coefficient (T_{RC1}) PSPICE only.	$^\circ C^{-1}$	1	*
TRC2	RC quadratic temperature coefficient (T_{RC2}) PSPICE only.	$^\circ C^{-2}$	1	*
TRE1	RE linear temperature coefficient (T_{RE1}) PSPICE only.	$^\circ C^{-1}$	1	*
TRE2	RE quadratic temperature coefficient (T_{RE2}) PSPICE only.	$^\circ C^{-2}$	1	*
TRM1	RBM linear temperature coefficient (T_{RM1}) PSPICE only.	$^\circ C^{-1}$	1	*
TRM2	RBM quadratic temperature coefficient (T_{RM2}) PSPICE only.	$^\circ C^{-2}$	1	*
VA	alternative keyword for VAF PSPICE only.			
VB	alternative keyword for VAR PSPICE only.			
VAF	forward Early voltage (V_{AF})	V	∞	
VAR	reverse Early voltage (V_{AR})	V	∞	

Continued on next page

Table 7.12: BJT model parameters.

Name	Description	Units	Default	Are
VJC	base-collector built-in potential (V_{JC})	V	0.75	
VJE	base-emitter built-in potential (V_{JE})	V	0.75	
VJS	substrate junction built-in potential (V_{JS})	V		
VTF	voltage describing V_{BC} dependence of TF ($V_{\tau F}$)	V	∞	
XCJC	fraction of B-C depletion capacitance connected to internal base node (X_{CJC})	-	1	
XTB	forward and reverse beta temperature exponent (X_{TB})	-		
XTI	temperature exponent for effect on IS (X_{TI})	-	3	
XTF	coefficient for bias dependence of TF ($X_{\tau F}$)	-		

Standard Calculations

The physical constants used in the model evaluation are

k	Boltzman's constant	$1.3806226\,10^{-23}$ J/K
q	electronic charge	$1.6021918\,10^{-19}$ C

Absolute temperatures (in kelvins, K) are used. The thermal voltage

$$V_{\mathrm{TH}}(T_{\mathrm{NOM}}) = \frac{kT_{\mathrm{NOM}}}{q}. \tag{7.466}$$

Temperature Dependence

Temperature effects are incorporated as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K).

$$V_{\rm TH} = \frac{kT}{q} \tag{7.467}$$

$$I_S(T) = I_S e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/V_{\rm TH}} + \left(\frac{T}{T_{\rm NOM}}\right)^{X_{TI}/N_F} \tag{7.468}$$

$$I_{SE}(T) = I_{SE} e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/V_{\rm TH}} + \left(\frac{T}{T_{\rm NOM}}\right)^{X_{TI}/N_E} \tag{7.469}$$

$$I_{SC}(T) = I_{SC} e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/V_{\rm TH}} + \left(\frac{T}{T_{\rm NOM}}\right)^{X_{TI}/N_C} \tag{7.470}$$

$$I_{SS}(T) = I_{SS} e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/V_{\rm TH}} + \left(\frac{T}{T_{\rm NOM}}\right)^{X_{TI}/N_S} \tag{7.471}$$

$$V_{JE}(T) = V_{JE}(T_{\rm NOM})(T - T_{\rm NOM}) - 3V_{\rm TH}\ln\left(\frac{T}{T_{\rm NOM}}\right) E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.472}$$

$$V_{JC}(T) = V_{JC}(T_{\rm NOM})(T - T_{\rm NOM}) - 3V_{\rm TH}\ln\left(\frac{T}{T_{\rm NOM}}\right) E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.473}$$

$$V_{JS}(T) = V_{JS}(T_{\rm NOM})(T - T_{\rm NOM}) - 3V_{\rm TH}\ln\left(\frac{T}{T_{\rm NOM}}\right) E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.474}$$

$$C_{JC}(T) = C_{JC}\{1 + M_{JC}[0.0004(T - T_{\rm NOM}) + (1 - V_{JC}(T)/V_{JC}(T_{\rm NOM}))]\} \tag{7.475}$$

$$C_{JE}(T) = C_{JE}\{1 + M_{JE}[0.0004(T - T_{\rm NOM}) + (1 - V_{JE}(T)/V_{JE}(T_{\rm NOM}))]\} \tag{7.476}$$

$$C_{JS}(T) = C_{JS}\{1 + M_{JS}[0.0004(T - T_{\rm NOM}) + (1 - V_{JS}(T)/V_{JS}(T_{\rm NOM}))]\} \tag{7.477}$$

$$\beta_F(T) = \beta_F(T_{\rm NOM})^{X_{TB}} \tag{7.478}$$

$$\beta_R(T) = \beta_R(T_{\rm NOM})^{X_{TB}} \tag{7.479}$$

$$E_G(T) = E_G(T_{\rm NOM}) - 0.000702\frac{T^2}{T + 1108} \tag{7.480}$$

$$R_B(T) = R_B(T_{\rm NOM})\left[1 + T_{RB1}(T - T_{\rm NOM}) + T_{RB2}(T - T_{\rm NOM})^2\right] \tag{7.481}$$

$$R_{BM}(T) = R_{BM}(T_{\rm NOM})\left[1 + T_{RM1}(T - T_{\rm NOM}) + T_{RM2}(T - T_{\rm NOM})^2\right] \tag{7.482}$$

$$R_C(T) = R_C(T_{\rm NOM})\left[1 + T_{RC1}(T - T_{\rm NOM}) + T_{RC2}(T - T_{\rm NOM})^2\right] \tag{7.483}$$

$$R_E(T) = R_E(T_{\rm NOM})\left[1 + T_{RE1}(T - T_{\rm NOM}) + T_{RE2}(T - T_{\rm NOM})^2\right] \tag{7.484}$$

Capacitances

The base-emitter capacitance, $C_{BE} = Area(C_{BE\tau} + C_{BEJ})$ (7.485)
where the base-emitter transit time or diffusion capacitance

$$C_{BE\tau} = \tau_{F,\mathrm{EFF}} \frac{\partial I_{BF}}{\partial V_{BE}} \tag{7.486}$$

the effective base transit time is empirically modified to account for base puchout, space-charge limited current flow, quasi-saturation and lateral spreading which tend to increase τ_F

$$\tau_{F,\mathrm{EFF}} = \tau_F \left[1 + X_{TF}(3x^2 - 2x^3) e^{(V_{BC}/(1.44 V_{TF})}\right] \tag{7.487}$$

and $x = I_{BF}/(I_{BF} + AreaI_{TF})$. The base-emitter junction (depletion) capacitance

$$C_{BEJ} = \begin{cases} C_{JE}\left(1 - \frac{V_{BE}}{V_{JE}}\right)^{-M_{JE}} & V_{BE} \le F_C V_{JE} \\ C_{JE}\left(1 - F_C\right)^{-(1+M_{JE})}\left(1 - F_C(1+M_{JE}) + M_{JE}\frac{V_{BE}}{V_{JE}}\right) & V_{BE} > F_C V_{JE} \end{cases} \tag{7.488}$$

The base-collector capacitance, $C_{BC} = Area(C_{BC\tau} + X_{CJC} C_{BCJ})$ (7.489)
where the base-collector transit time or diffusion capacitance

$$C_{BC\tau} = \tau_R \frac{\partial I_{BR}}{\partial V_{BC}} \tag{7.490}$$

The base-collector junction (depletion) capacitance

$$C_{BCJ} = \begin{cases} C_{JC}\left(1 - \frac{V_{BC}}{V_{JC}}\right)^{-M_{JC}} & V_{BC} \le F_C V_{JC} \\ C_{JC}\left(1 - F_C\right)^{-(1+M_{JC})}\left(1 - F_C(1+M_{JC}) + M_{JC}\frac{V_{BC}}{V_{JC}}\right) & V_{BC} > F_C V_{JC} \end{cases} \tag{7.491}$$

The capacitance between the extrinsic base and the intrinsic collector

$$C_{BX} = \begin{cases} Area(1 - X_{CJC}) C_{JC}\left(1 - \frac{V_{BX}}{V_{JC}}\right)^{-M_{JC}} & V_{BX} \le F_C V_{JC} \\ (1 - X_{CJC}) C_{JC}\left(1 - F_C\right)^{-(1+M_{JC})} & V_{BX} > F_C V_{JC} \\ \quad \times \left(1 - F_C(1+M_{JC}) + M_{JC}\frac{V_{BX}}{V_{JC}}\right) & \end{cases} \tag{7.492}$$

The substrate junction capacitance

$$C_{JS} = \begin{cases} AreaC_{JS}\left(1 - \frac{V_{CJS}}{V_{JS}}\right)^{-M_{JS}} & V_{CJS} \le 0 \\ AreaC_{JS}\left(1 + M_{JS}\frac{V_{CJS}}{V_{JS}}\right) & V_{CJS} > 0 \end{cases} \tag{7.493}$$

I/V Characteristics

The base-emitter current, $I_{BE} = \frac{I_{BF}}{\beta_F} + I_{LE}$ (7.494)

the base-collector current, $I_{BC} = \frac{I_{BR}}{\beta_R} + I_{LC}$ (7.495)

and the collector-emitter current, $I_{CE} = \frac{I_{BF} - I_{BR}}{K_{QB}}$ (7.496)

where the forward diffusion current, $I_{BF} = I_S \left(e^{V_{BE}/(N_F V_{\rm TH})} - 1 \right)$ (7.497)

the nonideal base-emitter current, $I_{LE} = I_{SE} \left(e^{V_{BE}/(N_E V_{\rm TH})} - 1 \right)$ (7.498)

the reverse diffusion current, $I_{BR} = I_S \left(e^{V_{BC}/(N_R V_{\rm TH})} - 1 \right)$ (7.499)

the nonideal base-collector current, $I_{LC} = I_{SC} \left(e^{V_{BC}/(N_C V_{\rm TH})} - 1 \right)$ (7.500)

and the base charge factor, $K_{QB} = \frac{1}{2} \left[1 - \frac{V_{BC}}{V_{AF}} - \frac{V_{BE}}{V_{AB}} \right]^{-1} \left(1 + \sqrt{1 + 4 \left(\frac{I_{BF}}{I_{KF}} + \frac{I_{BR}}{I_{KR}} \right)} \right)$ (7.501)

Thus the conductive current flowing into the base, $I_B = I_{BE} + I_{BC}$ (7.502)

the conductive current flowing into the collector, $I_C = I_{CE} - I_{BC}$ (7.503)

and the conductive current flowing into the emitter, $I_C = I_{BE} + I_{CE}$ (7.504)

Parasitic Resistances

The resistive parasitics R_B, R_E, are R_C are scaled by the area factor, *Area*, specified on the element line. This enables the model parameters `RB`, `RE` and `RC` to be absolute quantities if *Area* is omitted as it defaults to 1, or as sheet resistivities.

$$R'_B = R_B/Area \quad (7.505)$$

$$R'_C = R_C/Area \quad (7.506)$$

$$R'_E = R_E/Area \quad (7.507)$$

$$R'_B = \begin{cases} R_{BM} + \dfrac{R_B - R_{BM}}{K_{QB}} & I_{RB} \text{ omitted} \\ R_{BM} + 3(R_B - R_{BM}) \dfrac{\tan x - x}{x \tan^2(x)} & I_{RB} \text{ defined} \end{cases} \quad (7.508)$$

where $x = \left(\sqrt{1 + \frac{144 I_B}{I_{RB} \pi^2}} - 1 \right) \left(\frac{24}{\pi^2} \sqrt{\frac{I_B}{I_{RB}}} \right)^{-1}$ (7.509)

AC Analysis

The AC analysis uses the model of figure 7.46 with the capacitor values evaluated at the `DC` operating point with

$$g_m = \frac{\partial I_{CE}}{\partial V_{BE}} \quad (7.510)$$

and

$$R_O = \frac{\partial I_{CE}}{\partial V_{CE}} \quad (7.511)$$

Noise Analysis

The BJT noise model accounts for thermal noise generated in the parasitic resistances and shot and flicker noise generated in the base-emitter and base-collector junction regions. The rms (root-mean-square) values of thermal noise current generators shunting the three parasitic resistance R_B, R_C, and R_E are

$$I_{n,B} = \sqrt{4kT/R_B}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.512}$$

$$I_{n,C} = \sqrt{4kT/R_C}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.513}$$

$$I_{n,E} = \sqrt{4kT/R_E}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.514}$$

The rms value of the base noise current generator is

$$I_{n,B} = \left(I_{\mathrm{SHOT},B}^2 + I_{\mathrm{FLICKER},B}^2\right)^{1/2} \tag{7.515}$$

where

$$I_{\mathrm{SHOT},B} = \sqrt{2qI_B} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.516}$$

$$I_{\mathrm{FLICKER},B} = \sqrt{K_F I_B^{A_F}/f} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.517}$$

and f is frequency. The rms value of the collector noise current generator is

$$I_{n,C} = \left(I_{\mathrm{SHOT},C}^2 + I_{\mathrm{FLICKER},C}^2\right)^{1/2} \tag{7.518}$$

where

$$I_{\mathrm{SHOT},C} = \sqrt{2qI_C} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.519}$$

$$I_{\mathrm{FLICKER},C} = \sqrt{K_F I_C^{A_F}/f} \quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.520}$$

R Resistor

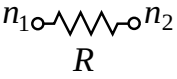


Figure 7.64: R — resistor element.

Form

R*name* N_1 N_2 *ResistorValue* IC=V_R]

SPICE3*Form*

R*name* N_1 N_2 [*ModelName*] *ResistorValue* [IC=V_R] [L= *Length*] [W= *Width*]

PSPICE*Form*

R*name* N_1 N_2 [*ModelName*] *ResistorValue* []

N_1 is the positive element node,

N_2 is the negative element node, and

ModelName is the optional model name.

ResistorValue is the resistance. (Units: F; Required; Symbol: $ResistorValue$;

IC is the optional initial condition specification Using IC=V_C is intended for use with the UIC option on the .TRAN line, when a transient analysis is desired with an initial voltage V_C across the capacitor rather than the quiescent operating point. Specification of the transient initial conditions using the .IC statement (see page 66) is preferred and is more convenient.

Example

```
R1 1 2 12.3MEG
```

Model Type

RES

RES Model SPICE3 Only Resistor Model

Form

.MODEL *ModelName* RES([[*keyword* = *value*] ...])

Example

```
.MODEL SMALLRES RES( )
```

Model Keywords

Name	Description	Units	Default
DEFW	default device width (W_{DEF})	meters	1e-6
NARROW	narrowing due to side etching (X_{NARROW})	meters	0.0

The SPICE3 resistor model is a process model for a monolithicly fabricated resistor enabling the capacitance to be determined from geometric information. If the parameter W is not specified on the element line then $Width$ defaults to DEFW $= W_{\text{DEF}}$. The effective dimensions are reduced by etching so that the effective length of the capacitor is

$$L_{\text{EFF}} = Length - X_{\text{NARROW}} \tag{7.521}$$

and the effective width is

$$W_{\text{EFF}} = Width - X_{\text{NARROW}} \tag{7.522}$$

The value of the capacitance

$$CapacitorValue = C_J L_{\text{EFF}} W_{\text{EFF}} + C_{J,\text{SW}} L_{\text{EFF}} + W_{\text{EFF}} + L_{\text{EFF}} + W_{\text{EFF}}) \tag{7.523}$$

RES Model — PSPICE Only — Resistor Model

The resistor model consists of process-related device data that allow the resistance to be calculated from geometric information and to be corrected for temperature. The parameters available are:
Keywords:

Name	Description	Units	Default
R	resistance multiplier ($R_{\text{MULTIPLIER}}$)	-	1
TC1	first order temperature coefficient (T_{C1})	$^\circ\text{C}^{-1}$	0
TC2	second order temperature coefficient (T_{C2})	$^\circ\text{C}^{-2}$	0
TCE	exponential temperature coefficient (T_{CE})	$\%/^\circ\text{C}^{-2}$	0

The sheet resistance is used with the narrowing parameter and L and W from the resistor line to determine the nominal resistance by the formula

$$R = \begin{cases} ResistorValue & \text{no model specified} \\ ResistorValue\, R_{\text{MULTIPLIER}} \times \left[1 + T_{C1}(T - T_{\text{NOM}}) + T_{C2}(T - T_{\text{NOM}})^2\right] & \text{model specified and TCE not specified} \\ ResistorValue\, R_{\text{MULTIPLIER}} 1.01^{[T_{CE}(T - T_{\text{NOM}})]} & \text{model and TCE specified} \end{cases} \tag{7.524}$$

Noise Analysis

The resistor noise model accounts for thermal noise generated in the resistance. The rms (root-mean-square) values of thermal noise current generators shunting the resistance is

$$I_n = \sqrt{4kT/R}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.525}$$

S Voltage Controlled Switch

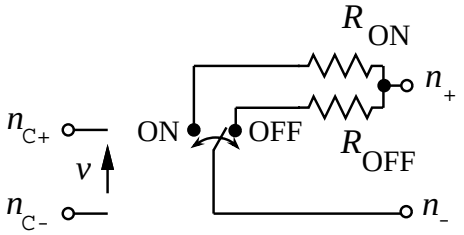


Figure 7.65: S — voltage controlled switch element.

Form

`S`*name* N_+ N_- N_{C+} N_{C-} *ModelName* [`ON`] [`OFF`]

PSPICE*Form*

`S`*name* N_+ N_- N_{C+} N_{C-} *ModelName*

Example

```
S1 1 2 3 4 SWITCH1
S2 5 6 3 0 SM2
SWITCH1 1 2 10 0 SMODEL1
```

N_+ is the positive node of the switch.

N_- is the negative node of the switch.

N_{C+} is the positive controlling node of the switch.

N_{C-} is the negative controlling node of the switch.

ModelName is the model name and is required.

`ON` is the optional initial condition. It is intended for use with the `UIC` option on the `.TRAN` line, when a transient analysis is desired starting from other than the quiescent operating point. It is also the initial condition on the device for `DC` analysis.

`OFF` is the optional initial condition. If specified the `DC` operating point is calculated with the terminal voltages set to zero. Once convergence is obtained, the program continues to iterate to obtain the exact value of the terminal voltages. The OFF option is used to enforce the solution to correspond to a desired state if the circuit has more than one stable state.

The initial conditions are optional. For the voltage controlled switch, nodes N_{C+} and NC- are the positive and negative controlling nodes respectively. For the current controlled switch, the controlling current is that through the specified voltage source. The direction of positive controlling current flow is from the positive node, through the source, to the negative node.

Model Type

`VSWITCH`

VSWITCH Model — Voltage-Controlled Switch Model

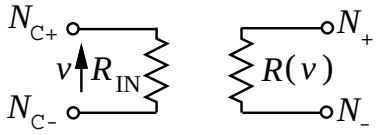


Figure 7.66: VSWITCH — voltage controlled switch model.

The voltage-controlled switch model is supported by both SPICE3 and PSPICE. However the model keywords differ slightly.

SPICE3 *keywords:*

Name	Description		Units	Default
`VT`	threshold voltage	(V_{ON})	V	0.0
`VH`	hysteresis voltage	(V_{OFF})	V	0.0
`RON`	on resistance	(R_{ON})	Ω	1.0
`ROFF`	off resistance	(R_{OFF})	Ω	1/GMIN

PSPICE *keywords:*

Name	Description		Units	Default
`VON`	threshold voltage	(V_{ON})	V	0.0
`VOFF`	hysteresis voltage	(V_{OFF})	V	0.0
`RON`	on resistance	(R_{ON})	Ω	1.0
`ROFF`	off resistance	(R_{OFF})	Ω	1/GMIN

Care must be exercised in using the switch. An instantaneous switch is highly nonlinear and will very likely lead to convergence problems. This problem is alleviated in the `VSWITCH` model by ramping the resistance of the switch from its off value to its on value. For this ramping action to be effective the difference between V_{ON} and V_{OFF} must not be too small. Also the values of R_{ON} and R_{OFF} should not be extreme. The ration R_{ON}/R_{OFF} should be be as small as possible.

If R_{ON}/R_{OFF} is large, e.g. $R_{ON}/R_{OFF} > 10^{12}$, then the default error tolerances `TRTOL` and `CHGTOL`, specified in a `.OPTIONS` statement (see page 83) may need to be changed.

`TRTOL` Change to 1.0 from 7.0 idf there are convergence problems during transient analysis.

`CHGTOL` If a switch is across a capacitor then `CHGTOL` should be reduced to 10^{-16} if there are convergence problems during transient analysis.

Switch Model

The switch is modeled by a voltage variable resistor R and an input input resistance $R_{\rm IN}$, see figure 7.66. $R_{\rm IN} = 1/G_{\rm MIN}$ to ensure that the controlling nodes are not floating and that the voltage v between the controlling nodes can not change instantaneously. $G_{\rm MIN}$ = `GMIN` is described on page **??**.

Standard Calculations

$$R_{\rm MEAN} = \sqrt{R_{\rm ON} + R_{\rm OFF}} \tag{7.526}$$

$$R_{\rm RATIO} = R_{\rm ON}/R_{\rm OFF} \tag{7.527}$$

$$V_{\rm MEAN} = \sqrt{V_{\rm ON} + V_{\rm OFF}} \tag{7.528}$$

$$V_\Delta = \left(\frac{v - V_{\rm MEAN}}{V_{\rm ON} - V_{\rm OFF}}\right) \tag{7.529}$$

If $V_{\rm ON} > V_{\rm OFF}$ the switch resistance

$$R = \begin{cases} R_{\rm ON} & v \geq V_{\rm ON} \\ R_{\rm OFF} & v \leq V_{\rm OFF} \\ R_{\rm MEAN}\, R_{\rm RATIO}^{1.5V_\Delta}\, R_{\rm RATIO}^{1.5V_\Delta^3} & V_{\rm OFF} < v < V_{\rm ON} \end{cases} \tag{7.530}$$

If $V_{\rm ON} < V_{\rm OFF}$ the switch resistance

$$R = \begin{cases} R_{\rm ON} & v \leq V_{\rm ON} \\ R_{\rm OFF} & v \geq V_{\rm OFF} \\ R_{\rm MEAN}\, R_{\rm RATIO}^{1.5V_\Delta}\, R_{\rm RATIO}^{1.5V_\Delta^3} & V_{\rm OFF} < v < V_{\rm ON} \end{cases} \tag{7.531}$$

Noise Analysis

The voltage controlled switch noise model accounts for thermal noise generated in the switch resistance. The rms (root-mean-square) values of thermal noise current generators shunting the switch resistance is

$$I_n = \sqrt{4kT/R}\ {\rm A}/\sqrt{\rm Hz} \tag{7.532}$$

where T is the analysis temperature in kelvin (K), and k ($= 1.3806226\,10^{-23}$ J/K) is Boltzmans constant.

T Transmission Line

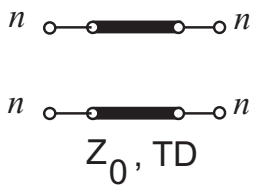


Figure 7.67: T — transmission line element.

Form

T*name* n_1 n_2 n_3 n_4 Z0=*CharacteristicImpedance* TD=*TimeDelay* [IC=V_1, I_1, V_2, I_2]

T*name* n_1 n_2 n_3 n_4 Z0=*CharacteristicImpedance* F=*ReferenceFrequency* + [NL=*NormalizedElectricalLength*] [IC=V_1, I_1, V_2, I_2]

SPICE3*Form*

T*name* n_1 n_2 n_3 n_4 [*ModelName*] Z0=*CharacteristicImpedance* TD=*TimeDelay* [IC=V_1, I_1, V_2, I_2]

T*name* n_1 n_2 n_3 n_4 [*ModelName*] Z0=*CharacteristicImpedance* + F=*ReferenceFrequency* + [NL=*NormalizedElectricalLength*] [IC=V_1, I_1, V_2, I_2]

PSPICE*Form*

T*name* n_1 n_2 n_3 n_4 Z0=*CharacteristicImpedance* TD=*TimeDelay*

T*name* n_1 n_2 n_3 n_4 Z0=*CharacteristicImpedance* +F=*Frequency* [NL=*NormalizedElectricalLength*]

n_1 positive node at port 1.

n_2 negative node at port 2.

n_3 positive node at port 2.

n_4 negative node at port 2.

ModelName is the model name.

Z0 is the characteristic impedance. (Z-zero)
(Units: Ω; Required; Symbol: Z_0; Default: none)

TD transmission line delay. (Units: s; Either TD or F Required; Symbol: T_D; Default: none)

F reference frequency. (Units: Hz; Either TD or F Required; Symbol: F; Default: none)

`NL` normalized electrical length. Normalization is with respect to the wavelength in free space at the reference frequency `F`.
(Units: none; Optional; Symbol: $L_{\text{NORMALIZED}}$; Default: 0.25)

`IC` is the optional initial condition specification using `IC=` V_1, I_1, V_2, I_2 is intended for use with the `UIC` option on the `.TRAN` line, when a transient analysis is desired starting from other than the quiescent operating point. Specification of the transient initial conditions using the `.IC` statement (see page 66) is preferred and is more convenient.

Example

```
T1 1 0 2 0 Z0=50 TD=10NS

TLONG 1 0 2 0 Z0=50 F=1G NL=10

TLONG 1 0 2 0 Z0=50 F=1G
```

Note

1. The length of the line may be expressed in either of two forms. The transmission delay, *TD*, may be specified directly (as TD=10ns, for example). Alternatively, a frequency F may be given, together with NL, the normalized electrical length of the transmission line with respect to the wavelength in the line at the frequency F. If a frequency is specified but NL is omitted, 0.25 is assumed (that is, the frequency is assumed to be the quarter-wave frequency). Note that although both forms for expressing the line length are indicated as optional, one of the two must be specified.

2. Note that only 3 distinct nodes should be specified as this element describes a single propagating mode. With four distinct nodes specified, two propagating modes may exist on the actual line. If there are four distinct nodes then two lines are required.

3. The transmission line `T` element is modeled as a bidirectional ideal delay element. The maximum time step in SPICE is limited to half of the time delay along the line. Thus short transmission lines can result in many time steps in a transient analysis. Unnecessary short transmission lines should be avoided.

Model Type

`URC`

URC Model — Lossy RC Transmission Line Model

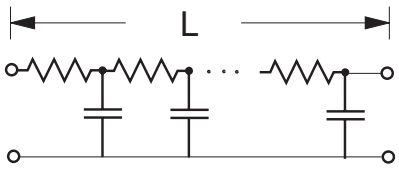


Figure 7.68: URC — lossy RC transmission line model: (a) linear RC transmission line model; (b) nonlinear transmission line model.

Form

`.MODEL` *ModelName* `URC(` [[*keyword* = *value*] ...] `)`

Example

```
.MODEL LONGLINE URC( )
```

Table 7.13: URC model parameters.

Name	Description	Units	Default
K	Propagation Constant	-	2.0
FMAX	Maximum Frequency of interest	Hz	1.0G
RPERL	Resistance per unit length ($I_{S,\text{PERL}}$)	Ω/m	1000
CPERL	Capacitance per unit length ($I_{S,\text{PERL}}$)	F/m	1.0E-15
ISPERL	Saturation current per unit length ($I_{S,\text{PERL}}$)	A/m	
RSPERL	Diode Resistance per unit length ($I_{S,\text{PERL}}$)	Ω/m	0

The URC model was originally proposed by Gertzberrg [**?**] In this model a transmission line is represented by the cascade of a number of transmission line segments each of which is modeled by an RC or R-Diode subcircuit. The lengths of the line segments increases in a geometric progression towards the middle of the line. The number of line segments is

$$N = \tag{7.533}$$

and the length of the ith line segment is

$$l_i = \tag{7.534}$$

If `ISPERL` is not specified then a linear transmission line is modeled, see figure 7.68, with

$$R_i = R_{\text{PERL}} l_i \tag{7.535}$$

$$C_i = C_{\text{PERL}} l_i \tag{7.536}$$

If `ISPERL` is not then a diode loaded nonlinear transmission line is modeled, see figure 7.68, with

$$R_i = R_{\text{PERL}} l_i \tag{7.537}$$

$$R_{S,i} = R_{S,\text{PERL}} l_i \tag{7.538}$$

$$C_i = C_{J,i} \left(1 - \frac{\phi}{V_{J,i}} \right)^{-\frac{1}{2}} \tag{7.539}$$

$$I_S = I_{S,i} \left(e^{\frac{V_{J,i}}{V_{\text{TH}}}} - 1 \right) \tag{7.540}$$

where the zero-bias capacitance of the ith diode is

$$C_{J,i} = C_{\text{PERL}} l_i \tag{7.541}$$

its reverse saturation current is

$$I_{S,i} = I_{S,\text{PERL}} \tag{7.542}$$

U Universal Element

The U element is a universal element for extending the number of elements that SPICE can handle. The element is determined by the model type. In some cases model parameters indicate further refinement.

Since the SPICE U element is used to specify many different kinds of physical elements there is no common form for it. Each physical element is distinguished by the type of the model it refers to. The format and description of each element is given in the model description as indicated in the following table. The index is used in this manual to uniquely identify the various elements.

Model Type

Model	Index	Description	Page
STRUC	STRUC	Geometric coupled, lossy planar transmission line	??
U	U311	Geometric coupled planar transmission line (up to 5 lines) Identifying Model Parameters: `LEVEL=3 EVEL=1 PVEL=1`	??
U	U312	Geometric coaxial cable Identifying Model Parameters: `LEVEL=3 EVEL=1 PVEL=2`	??
U	32	General transmission line (up to 5 lines) defined by precomputed parameters. Identifying Model Parameters: `LEVEL=3 EVEL=2`	??
U	U34	General transmission line (up to 5 lines) defined by measurements. Identifying Model Parameters: `LEVEL=3 EVEL=3`	??
U	U4	Digital output element Identifying Model Parameter: `LEVEL=4`	??
U	U5	Digital input element Identifying Model Parameter: `LEVEL=5`	??

Model	INdex	Description	Page
UDLY	UDLY	Delay Line	??
UEFF	UEFF	Edge-Triggered Flip-Flop	??
UGATE	UGATE	Standard Gate Model Form	??
UGFF	UGFF	Gated Flip-Flop	??
UIO	UIO	IO Model	??
URC	URC	Lossy RC transmission line	243
USUHD	USUHD	Setup and Hold Checker	??
UTGATE	UTGATE	Tri-State Gate	??
UWDTH	UWDTH	Pulse-Width Checker	??

Note

1. One of the problems with SPICE is that the first letter of an element's name is used to determine the type of an element. One consequence of this is that there can only be 26 elements — one for each letter of the alphabet. SPICE2G6The original version of SPICE from which all subsequent versions of SPICE have been developed had less than 26 elements and so this was not a problem. With the addition of new element types several of the originally unised letters were used and a universal element, the U element, introduced to handle even more. The U element is used to represent many different type of elements such as lossy transmission lines and digital devices. All of the U elements refer to models and the model name, and sometimes model parameters, used to indicate the actual element referred to.

V Independent Voltage Source

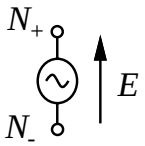


Figure 7.69: V — Independent voltage source.

Form

V*name* N_+ N_- [[*DC*] [*DCvalue*]
+ [AC [*ACmagnitude* [*ACphase*]]]
+ [DISTOF1 [*F1Magnitude* [*F1Phase*]]] + [DISTOF2 [*F2Magnitude* [*F2Phase*]]
]

SPICE3*Form*

V*name* N_+ N_- [[*DC*] [*DCvalue*]
+ [AC [*ACmagnitude* [*ACphase*]]]
+ [*TransientSpecification*]
+ [DISTOF1 [*F1Magnitude* [*F1Phase*]]]
+ [DISTOF2 [*F2Magnitude* [*F2Phase*]]]

PSPICE*Form*

V*name* N_+ N_- [[*DC*] [*DCvalue*] [AC [*ACmagnitude* [*ACphase*]]]
+ [*TransientSpecification*] [SNR *InputVoltageSNR*] [RS *SourceResistanceValue*]
[RL *LoadResistanceValue*]

Example

```
VBIAS 1 0 5.0
VCLOCK 20 5 PULSE(0 5 1N 2N 1.5N 21.9N 5N 20N)
VSSIGNAL AC 1U 90
```

N_+ is the positive voltage source node.

N_- is the negative voltage source node.

`DC` is the optional keyword for the `DC` value of the source.

DCvalue is the `DC` voltage value of the source.
(Units: V; Optional; Default: 0; Symbol: V_{DC})

`AC` is the keyword for the `AC` value of the source.

ACmagnitude is the `AC` magnitude of the source used during `AC` analysis. That is, it is the peak `AC` voltage so that the `AC` signal is $ACmagnitude \sin(\omega t + \text{ACphase})$. *ACmagnitude* is ignored for other types of analyses.
(Units: V; Optional; Default: 1; Symbol: V_{AC})

ACphase is the ac phase of the source. It is used only in `AC` analysis.
(Units: Degrees; Optional; Default: 0; Symbol: ϕ_{AC})

`DISTOF1` is the distortion keyword for distortion component 1 which hass frequency `F1`. (see the description of the `.DISTO` statement on page 58).

F1magnitude is the magnitude of the distortion component at `F1`. See `.DISTOF1` keyword above.
(Units: V; Optional; Default: 1; Symbol: V_{F1})

F1phase is the phase of the distortion component at `F1`. See `.DISTOF1` keyword above.
(Units: Degrees; Optional; Default: 0; Symbol: ϕ_{F1})

`DISTOF2` is the distortion keyword for distortion component 2 which hass frequency `F2`. (see the description of the `.DISTO` statement on page 58).

F2magnitude is the magnitude of the distortion component at `F2`. See `.DISTOF2` keyword above.
(Units: V; Optional; Default: 1; Symbol: V_{F2})

F2phase is the phase of the distortion component at `F2`. See `.DISTOF2` keyword above.
(Units: Degrees; Optional; Default: 0; Symbol: ϕ_{F2})

`SNR` is the input signal-to-noise ratio keyword. SomeVersionsOfSpice

InputVoltageSNR is the value of the signal-to-noise ratio at the input.
(Units: None; Optional; Default: use thermal noise of R_S; Symbol: SNR_I) SomeVersionsOfSpice

`RS` is the source resistance keyword. SomeVersionsOfSpice

SourceResistanceValue is the value of the source resistance.
(Units: Ohms; Optional; Default: 50Ω; Symbol: R_S)
Note: if port 1 is specified then the resistance specified for the port takes precedence. SomeVersionsOfSpice

`RL` is the source resistance keyword. SomeVersionsOfSpice

LoadResistanceValue is the value of the load resistance.
(Units: Ohms; Optional; Default: 50Ω; Symbol: R_L)
Note: if port 2 is specified then the resistance specified for the port takes precedence. SomeVersionsOfSpice

TransientSpecification is the optional transient specification described more fully below.

Note

1. The independent voltage source has three different sets of parameters to describe the source for DC analysis (see `.DC` on page 55), AC analysis (see `.AC` on page 53), and transient analysis (see `.TRAN` on page 109). The `DC` value of the source is used during bias point evaluation and `DC` analysis is *DCValue*. It is also the constant value of the voltage source if no *TransientSpecification* is supplied. It may also be used in conjunction with the `PWL` transient specification if a time zero value is not provided as part of the transient specification. The `AC` specification, indicated by the keyword `AC` is independent of the `DC` parameters and the *Transient Specification*.

2. See the `.NOISE` statement description for a discussion of how SNR$_I$ R_S, R_L are used in noise calculations.

3. The original documentation distributed with SPICE2G6 and SPICE3 incorrectly stated that if a *TransientSpecification* was supplied then the time-zero transient voltage was used in `DC` analysis and in determiniong the operating point.

Transient Specification

Five transient specification forms are supported: pulse (`PULSE`), exponential (`EXP`), sinusoidal (`SIN`), piece-wise linear (`PWL`), and single-frequency FM (`SFFM`). The default values of some of the parameters of these transient specifications include `TSTEP` which is the printing increment and `TSTOP` which is the final time (see the `.TRAN` statement on page 109 for further explanation of these quantities). In the following t is the transient analysis time.

Exponential:

Form

`EXP(` V_1 V_2 [T_{D1}] [τ_1] [T_{D2}] [τ_2] `)`

Name	Description	Units	Default
V_1	initial voltage	A	REQUIRED
V_2	pulsed voltage	A	REQUIRED
T_{D1}	rise delay time	s	0.0
τ_1	rise time constant	s	TSTEP
T_{D2}	fall delay time	s	T_{D1} + TSTEP
τ_2	fall time constant	s	TSTEP

The exponential transient is a single-shot event specifying two exponentials. The voltage is V_1 for the first T_{D1} seconds at which it begins increasing exponentially towards V_2 with a time constant of τ_1 seconds. At time T_{D2} the voltage exponentially decays towards V_1 with a time constant of τ_2. That is,

$$v = \begin{cases} V_1 & t \le T_{D1} \\ V_1 + (V_2 - V_1)(1 - e^{(-(t - T_{D1})/\tau_1)}) & T_{D1} < t \le T_{D2} \\ V_1 + (V_2 - V_1)(1 - e^{(-(t - T_{D1})/\tau_1)}) + (V_1 - V_2)(1 - e^{(-(t - T_{D2})/\tau_2)}) & t > T_{D2} \end{cases} \tag{7.543}$$

Single-Frequency FM:

Form

`SFFM(` V_O V_A F_C μ F_S `)`

Name	Description	Units	Default
V_O	offset voltage	A	
V_A	peak amplitude of AC voltage	A	
F_C	carrier frequency	Hz	1/TSTOP
μ	modulation index	-	0
F_S	signal frequency	Hz	1/TSTOP

The single frequency frequency modulated transient response is described by

$$v = V_O + V_A \sin\left(2\pi\, F_C t + \mu \sin\left(2\pi F_S t\right)\right) \tag{7.544}$$

Pulse:

Form

PULSE(V_1 V_2 [T_D] [T_R] [T_F] [W] [T])

Name	Description	Units	Default
V_1	initial voltage	A	REQUIRED
V_2	pulsed voltage	A	REQUIRED
T_D	delay time	s	0.0
T_R	rise time	s	TSTEP
T_F	fall time	s	TSTEP
W	pulse width	s	TSTOP
T	period	s	TSTOP

The pulse transient waveform is defined by

$$v = \begin{cases} V_1 & t \le T_D \\ V_1 + \frac{t'}{T_R}(V_2 - V_1) & 0 < t' \le T_R \\ V_2 & T_R < t' < (T_R + W) \\ V_2 - \frac{t' - W}{T_F}(V_1 - V_2) & (T_R + W) < t' < (T_R + W + T_F) \\ V_1 & (T_R + W + T_F) < t' < T \end{cases} \tag{7.545}$$

where

$$t' = t - T_D - (n-1)T \tag{7.546}$$

and t is the voltage analysis time and n is the cycle index. The effect of this is that after an initial time delay T_D the transient waveform repeats itself every cycle.

Piece-Wise Linear:

Form

PWL(T_1 V_1 [T_2 V_2 ... T_i V_i ... T_N V_N])

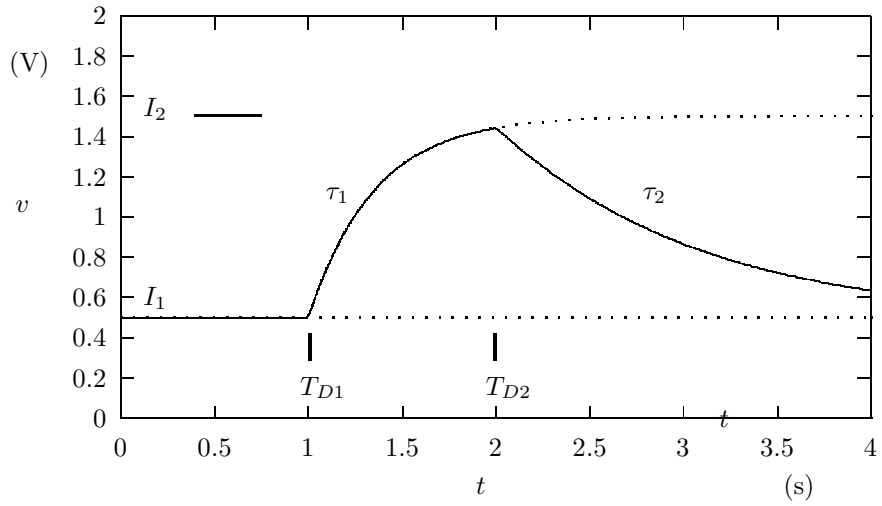


Figure 7.70: Voltage source exponential (`EXP`) waveform for `EXP(0.1 0.8 1 0.35 2 1)`

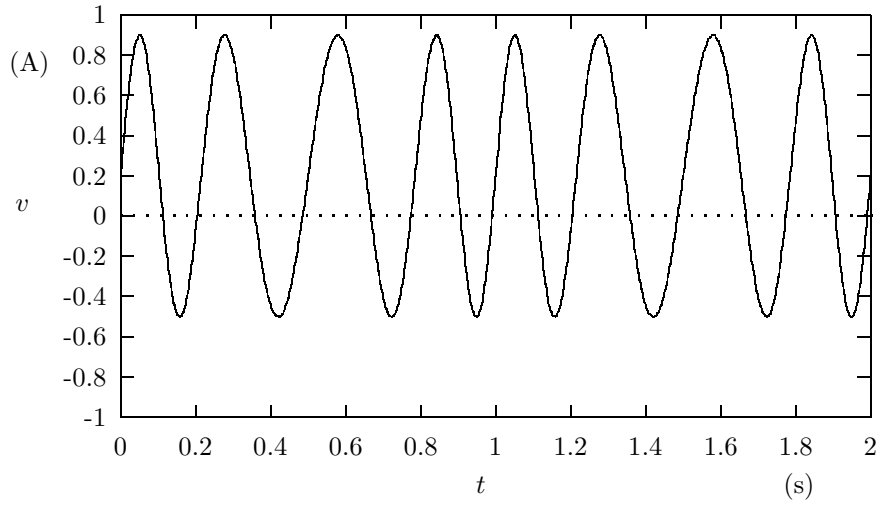


Figure 7.71: Voltage source single frequency frequency modulation (`SFFM`) waveform for `SFFM(0.2 0.7 4 0.9 1)`

Each pair of values (T_i, V_i) specifies that the value of the source is V_i at time = T_i. At times between T_i and T_{i+1} the values are linearly interpolated. If $T_1 > 0$ then the voltage is constant at *DCValue* (specified on the element line) until time T_1.

$$v = \begin{cases} DCvalue & t < T_1 \\ V_i & t = T_i \\ V_{i+1} & t = T_{i+1} \\ V_i + \left(\frac{t-T_i}{T_{i+1}-T_i}\right)(V_{i+1} - V_i) & T_i < t \le T_{i+1} \\ V_N & t > T_N \end{cases} \tag{7.547}$$

Sinusoidal:

Form

SIN(V_O V_A [F] [T_D] [θ])

PSPICE*Form*

SIN(V_O V_A [F] [T_D] [θ ϕ])

PSPICE*Form*

SIN(V_O V_A [F] [T_D] [θ ϕ])

Name	Description	Units	Default
V_O	voltage offset	A	REQUIRED
V_A	voltage amplitude	A	REQUIRED
F	frequency	Hz	1/TSTOP
T_D	time delay	s	0
Θ	damping factor	1/s	0
ϕ	phase	degree	0

The sinusoidal transient waveform is defined by

$$v = \begin{cases} V_0 & t \le T_D \\ V_0 + V_1 e^{-[(t-T_D)\Theta]} \sin 2\pi[F(t-T_D) + \phi/360] & t > T_D \end{cases} \tag{7.548}$$

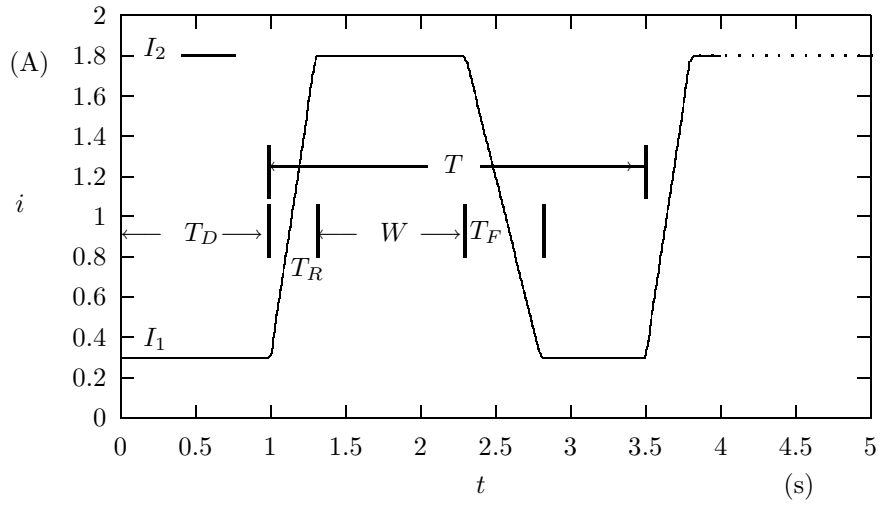


Figure 7.72: Voltage source transient pulse (`PULSE`) waveform for `PULSE(0.3 1.8 1 2.5 0.3 1 0.7)`

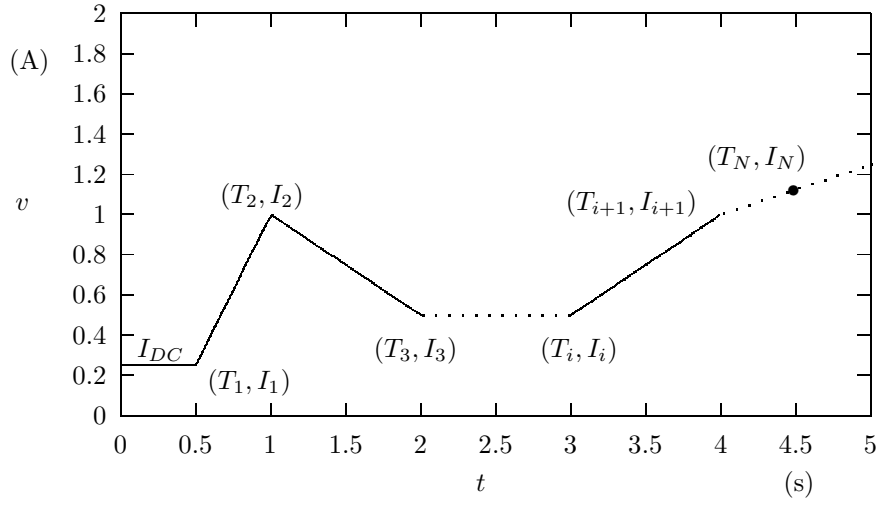


Figure 7.73: Voltage source transient piece-wise linear (`PWL`) waveform for
`PWL(1 0.25 1 1 2 0.5 ... 3 0.5 4 1 ... 4.5 1.25 ...)` with *DCValue = 0.25*.

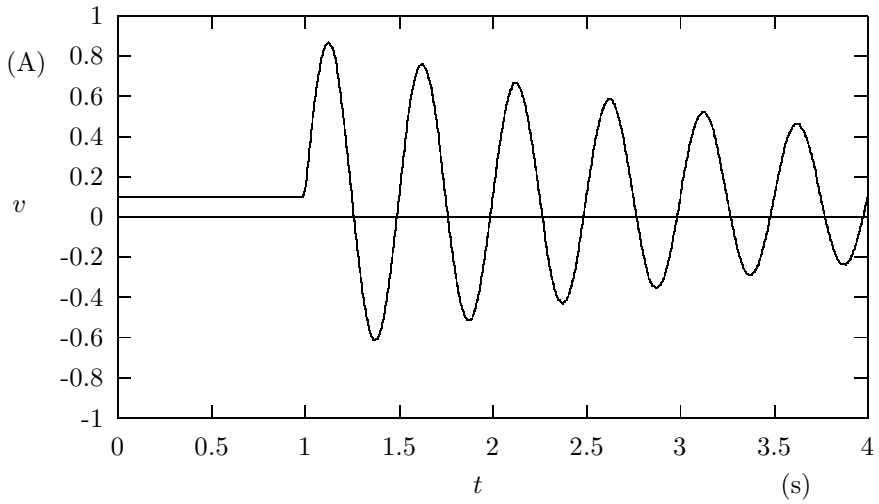


Figure 7.74: Voltage source transient sine (`SIN`) waveform for `SIN(0.1 0.8 2 1 0.3 )`.

W Current Controlled Switch

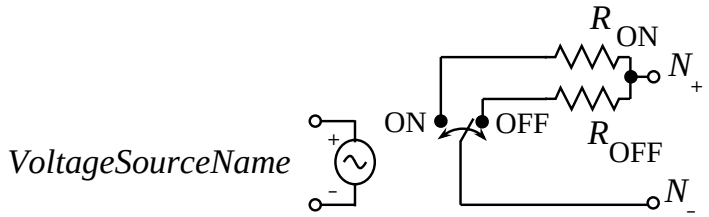


Figure 7.75: W — current controlled switch.

Form

`W`*name* N_1 N_2 *VoltageSourceName ModelName* [*ON*] [*OFF*]

PSPICE*Form*

`W`*name* N_1 N_2 *VoltageSourceName ModelName*

N_+ is the positive node of the switch.

N_- is the negative node of the switch.

VoltageSourceName is the name of the voltage source the current through which is the controlling current. The voltage source must be a `V` element.

`ON` is the optional initial condition. It is intended for use with the `UIC` option on the `.TRAN` line, when a transient analysis is desired starting from other than the quiescent operating point. It is also the initial condition on the device for `DC` analysis.

`OFF` is the optional initial condition. If specified the `DC` operating point is calculated with the terminal voltages set to zero. Once convergence is obtained, the program continues to iterate to obtain the exact value of the terminal voltages. The OFF option is used to enforce the solution to correspond to a desired state if the circuit has more than one stable state.

Model Type

`ISWITCH`

ISWITCH Model Current-Controlled Switch Model

The current-controlled switch model is supported by both SPICE3 and PSPICE. However the model keywords differ slightly.

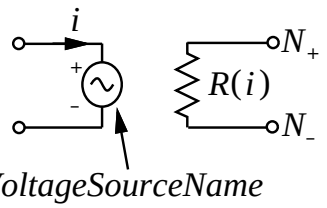


Figure 7.76: ISWITCH — current controlled switch model.

SPICE3 *keywords:*

Name	Description		Units	Default
IT	threshold current	(I_{ON})	A	0.0
IH	hysteresis current	(I_{OFF})	A	0.0
RON	on resistance	(R_{ON})	Ω	1.0
ROFF	off resistance	(R_{OFF})	Ω	1/GMIN

PSPICE *keywords:*

Name	Description		Units	Default
ION	threshold current	(I_{ON})	A	0.0
IOFF	hysteresis current	(I_{OFF})	A	0.0
RON	on resistance	(R_{ON})	Ω	1.0
ROFF	off resistance	(R_{OFF})	Ω	1/GMIN

Care must be exercised in using the switch. An instantaneous switch is highly nonlinear and will very likely lead to convergence problems. This problem is alleviated in the `ISWITCH` model by ramping the resistance of the switch from its off value to its on value. For this ramping action to be effective the difference between I_{ON} and I_{OFF} must not be too small. Also the values of R_{ON} and R_{OFF} should not be extreme. The ration R_{ON}/R_{OFF} should be be as small as possible.

If R_{ON}/R_{OFF} is large, e.g. $R_{ON}/R_{OFF} > 10^{12}$, then the default error tolerances `TRTOL` and `CHGTOL`, specified in a `.OPTIONS` statement (see page 83) may need to be changed.

`TRTOL` Change to 1.0 from 7.0 idf there are convergence problems during transient analysis.

`CHGTOL` If a switch is across a capacitor then `CHGTOL` should be reduced to 10^{-16} if there are convergence problems during transient analysis.

Switch Model

The switch is modeled by a current variable resistor R, see figure 7.76.

Standard Calculations

$$R_{\rm MEAN} = \sqrt{R_{\rm ON} + R_{\rm OFF}} \tag{7.549}$$
$$R_{\rm RATIO} = R_{\rm ON}/R_{\rm OFF} \tag{7.550}$$
$$I_{\rm MEAN} = \sqrt{I_{\rm ON} + I_{\rm OFF}} \tag{7.551}$$
$$I_{\Delta} = \left(\frac{i - I_{\rm MEAN}}{I_{\rm ON} - I_{\rm OFF}}\right) \tag{7.552}$$

If $I_{\rm ON} > I_{\rm OFF}$ the switch resistance

$$R = \begin{cases} R_{\rm ON} & i \geq I_{\rm ON} \\ R_{\rm OFF} & i \leq I_{\rm OFF} \\ R_{\rm MEAN}\, R_{\rm RATIO}^{1.5 I_{\Delta}}\, R_{\rm RATIO}^{1.5 I_{\Delta}^3} & I_{\rm OFF} < i < I_{\rm ON} \end{cases} \tag{7.553}$$

If $I_{\rm ON} < I_{\rm OFF}$ the switch resistance

$$R = \begin{cases} R_{\rm ON} & i \leq I_{\rm ON} \\ R_{\rm OFF} & i \geq I_{\rm OFF} \\ R_{\rm MEAN}\, R_{\rm RATIO}^{1.5 I_{\Delta}}\, R_{\rm RATIO}^{1.5 I_{\Delta}^3} & I_{\rm OFF} < i < I_{\rm ON} \end{cases} \tag{7.554}$$

Noise Analysis

The current controlled switch noise model accounts for thermal noise generated in the switch resistance. The rms (root-mean-square) values of thermal noise current generators shunting the switch resistance is

$$I_n = \sqrt{4kT/R}\ \text{A}/\sqrt{\text{Hz}} \tag{7.555}$$

where T is the analysis temperature in kelvin (K), and k ($= 1.3806226\,10^{-23}$ J/K) is Boltzmanns constant.

X Subcircuit Call

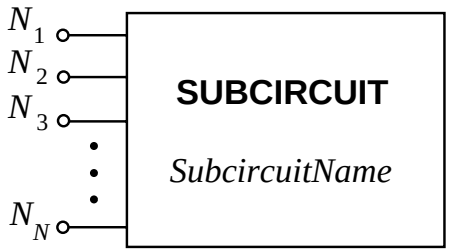


Figure 7.77: X — subcircuit call element.

Form

`X`*name* N_1 [N_2 N_3 ... N_N] *SubcircuitName*

PSPICE*Form*

`X`*name* N_1 [N_2 N_3 ... N_N] *SubcircuitName* [*PARAMS:* [*keyword* = { *Expression* } ...] [*Keyword* = *Value* ...]]

N_1 is the first node of the subcircuit.

N_N is the Nth node of the subcircuit.

SubcircuitName is the name of the subcircuit.

`PARAMS:` indicates that parameters are to be passed to the subcircuit.

`keyword:` is keyword corresponding to the keywords defined in the `.SUBCKT` statement. (See page 103).

`value:` is numeric value.

`Expression:` is an algebraic expression which evaluates to a numeric value. `.SUBCKT` statement. (See section **??** on page **??**).

Example

```
X1 2 4 17 3 1 MULTI
```

Subcircuits are incorporated by using the "`X`" element. The number of nodes of the "`X`" element must correspond to the number of nodes in the definition of the subcircuit (i.e. is on the `.SUBCKT` statement (see page 103).

Z Distributed Discontinuity

Only a few versions of SPICE support this. *Form*

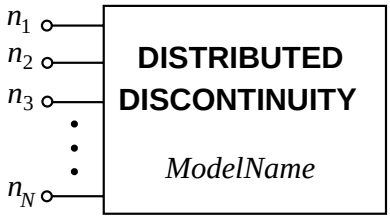


Figure 7.78: Z — distributed discontinuity.

Z*name N1 ... Nn Mname*

The Z element defines a subcircuit of L's and C's describing a distributed element. The L and C subcircuit is calculated using field theoretic based models and are optimized for maximum accuracy at 3GHz. In a contrast a quasistatic model is evaluated at `DC` and its accuracy will degrade as frequencies increase. The upper frequency of validity is selectable by the model parameter `N`. `N` is the harmonic of 3 GHz at which the model is valid. The default for `N` is 5 so that by default the models are valid to a frequency of 15 GHz.

Mname is the name of a model which can be one of several types:

`ZSTEP`	Microstrip impedance step
`LBEND`	Microstrip right-angle bend
`MBEND`	Microstrip right-angle bend
`TJUNC`	Microstrip T-junction (three way junction)
`XJUNC`	Microstrip X-junction (four way bend)

Example

```
Z1 1 5 7 Tjunc1
```

LBEND Model — Microstrip Right-Angle Bend Model

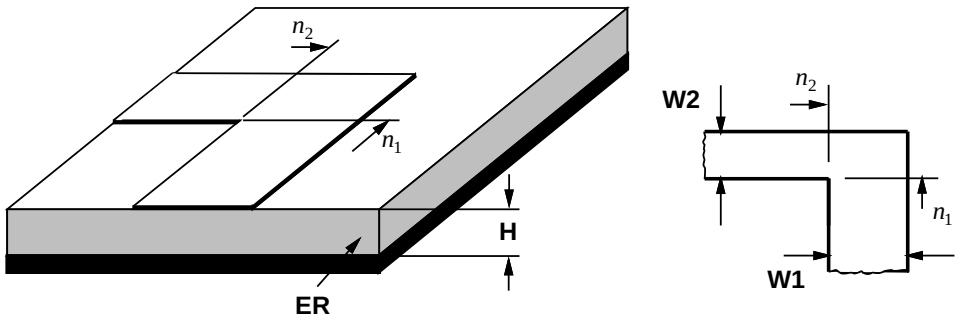


Figure 7.79: LBEND — microstrip right-angle bend model.

Form

`.MODEL` *Mname* `LBEND` [`ER`=*value*] `H`=*xvalue* `W1`=*xvalue* `W2`=*xvalue* [`N`=*ivalue*]

Keywords:

Name	Description	Units	Default
ER	Permittivity of dielectric layer	-	1.0
H	Height of dielectric layer	mm	REQUIRED
W1	First line width	mm	REQUIRED
W2	Second line width	mm	required
N	Number of calculated harmonics (0-15)	-	5

Example

```
.MODEL MB1 LBEND ER=9.8 H=0.635 W1=1.2 W2=3.2
```

MBEND Model — Microstrip Mitered Right-Angle Bend Model

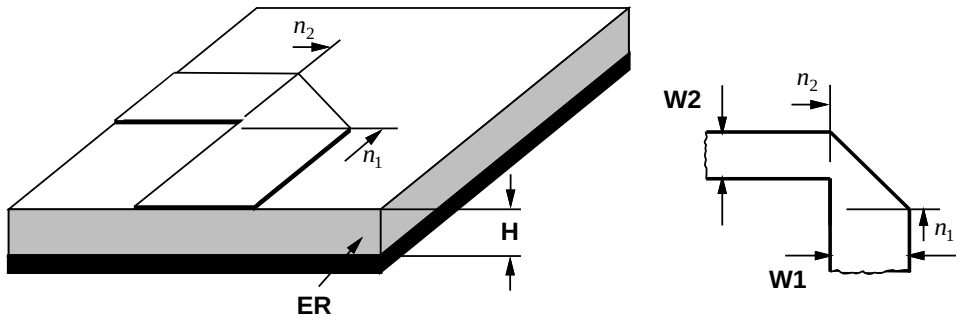


Figure 7.80: MBEND — microstrip mitered right-angle bend model.

Form

`.MODEL` *Mname* `MBEND` [`ER`=*value*] `H`=*xvalue* `W1`=*xvalue* `W2`=*xvalue* [`N`=*ivalue*]

Keywords:

Name	Description	Units	Default
ER	Permittivity of dielectric layer	-	1.0
H	Height of dielectric layer	mm	REQUIRED
W1	First line width	mm	REQUIRED
W2	Second line width	mm	REQUIRED
N	Number of calculated harmonics (0-15)	-	5

Example

```
.MODEL MB1 MBEND ER=9.8 H=0.635 W1=1.2 W2=3.2
```

TJUNC Model — Microstrip T-Junction Model

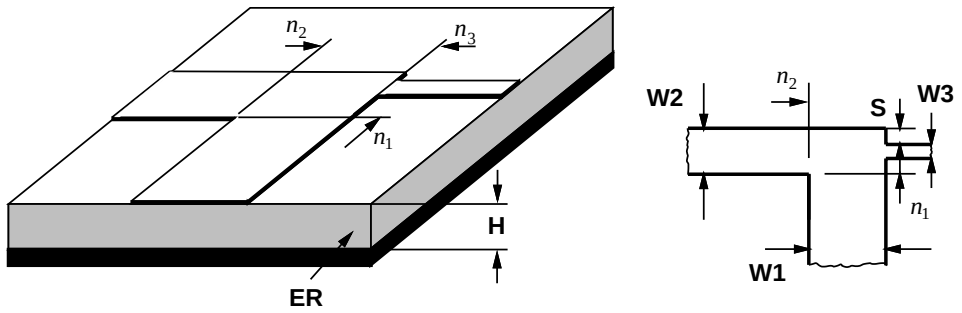


Figure 7.81: TJUNC — microstrip T-junction model.

Form

`.MODEL` *Mname* `TJUNC` [`ER`=*value*] `H`=*xvalue* `W1`=*xvalue* `W2`=*xvalue* `W3`=*xvalue* [`S`=*xvalue*] [`N`=*ivalue*]

Keywords:

Name	Description	Units	Default
ER	Permittivity of dielectric layer	-	1.0
H	Height of dielectric layer	mm	REQUIRED
W1	First line width	mm	REQUIRED
W2	Second line width	mm	REQUIRED
W3	Third line width	mm	REQUIRED
S	Edge offset of line 3	mm	0
N	Number of calculated harmonics (0-15)	-	5

Example

```
.MODEL TJ1 TJUNC ER=9.8 H=0.635 W1=1.2 W2=3.2 W3=1.1 W4=1
```

XJUNC Model — Microstrip X-junction

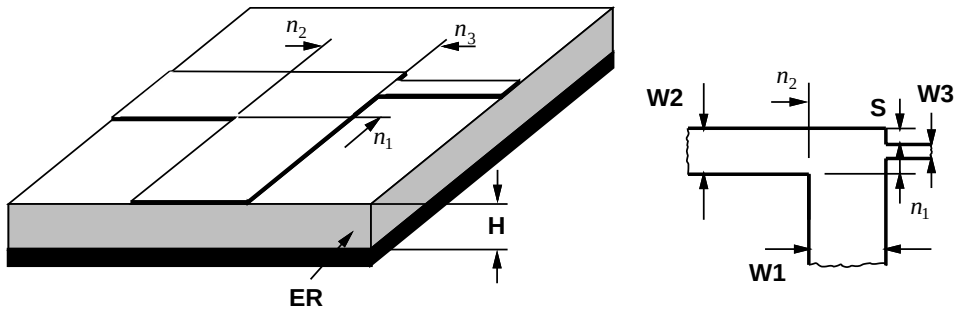


Figure 7.82: XJUNC — microstrip X-junction.

Form

`.MODEL` *Mname* `XJUNC` [`ER`=*value*] `H`=*xvalue* `W1`=*xvalue* `W2`=*xvalue* `W3`=*xvalue* `W4`=*xvalue* [`S`=*xvalue*] [`N`=*ivalue*]

Keywords:

Name	Description	Units	Default
ER	Permittivity of dielectric layer	-	1.0
H	Height of dielectric layer	mm	REQUIRED
W1	First line width	mm	REQUIRED
W2	Second line width	mm	REQUIRED
W3	Third line width	mm	REQUIRED
S	Edge offset of line 3	mm	0
N	Number of calculated harmonics (0-15)	-	5

Example

```
.MODEL TJ1 TJUNC ER=9.8 H=0.635 W1=1.2 W2=3.2 W3=1.1 W4=1
```

ZSTEP Model — Microstrip Impedance Step Model

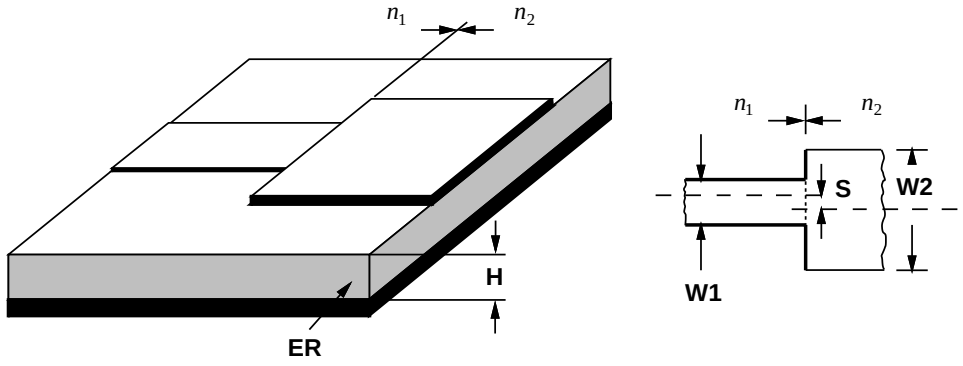


Figure 7.83: ZSTEP — microstrip impedance step model.

Form

`.MODEL` *Mname* `ZSTEP` [`ER`=*value*] `H`=*xvalue* `W1`=*xvalue* `W2`=*xvalue* [`S`=*xvalue*] [`N`=*ivalue*]

Keywords:

Name	Description	Units	Default
ER	Permittivity of dielectric layer	-	1.0
H	Height of dielectric layer	mm	required
W1	First line width	mm	required
W2	Second line width	mm	required
S	Displacement (see figure)	mm	0
N	Number of calculated harmonics (0-15)	-	5

Example

```
.MODEL STEP1 ZSTEP ER=9.8 H=0.635 W1=1.2 W2=3.2
```

Z MESFET

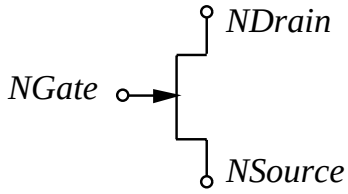


Figure 7.84: Z — GASFET element.

Form

Z*name NDrain NGate NSource ModelName* [*AREA*] [*OFF*] [*IC=VDS,VGS*]

Example

```
Z1 7 2 3 ZM1 OFF
```

NDrain is the drain node.

NGate is the gate node.

NSource is the source node.

`ModelName` is the model name.

`OFF` indicates an (optional) initial condition on the device for `DC` analysis. If specified the `DC` operating point is calculated with the terminal voltages set to zero. Once convergence is obtained, the program continues to iterate to obtain the exact value of the terminal voltages. The OFF option is used to enforce the solution to correspond to a desired state if the circuit has more than one stable state.

`IC` is the optional initial condition specification. Using `IC=`V_{DS}, V_{GS}, V_{BS} is intended for use with the `UIC` option on the `.TRAN` line, when a transient analysis is desired starting from other than the quiescent operating point. Specification of the transient initial conditions using the `.IC` statement (see page 66) is preferred and is more convenient.

Model Type

`GASFET`

GASFET Model GaAs MESFET Model

Form

`.MODEL` *ModelName* `GASFET(` [[*keyword* `=` *value*] ...] `)`

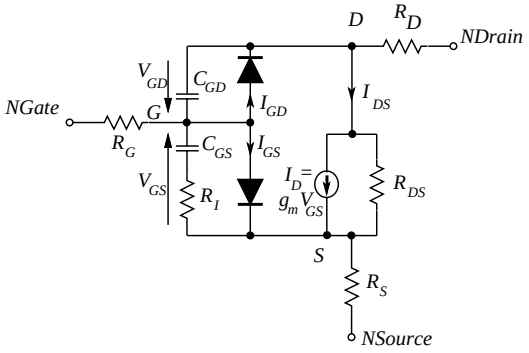


Figure 7.85: Schematic of the SPICE3GASFET model. V_{GS}, V_{DS}, and V_{GD} are intrinsic gate-source, drain-source and gate-drain voltages between the internal gate, drain, and source terminals designated G, D, and S respectively.

Example

```
.MODEL GAAS12 GASFET()
```

Raytheon model

This model is also known as the Statz model and model was developed at Raytheon for the modeling of GaAs MESFETs used in digital circuits. It is based on empirical fits to measured data [23].

The parameters of the `GASFET` model for PSPICEare given in table 7.14.

It is assumed that the model parameters were determined or measured at the nominal temperature $T_{\rm NOM}$ (default 27°C) specified in the most recent `.OPTIONS` statement preceeding the `.MODEL` statement. The physical constants used in the model evaluation are

Table 7.14: SPICE3GASFET model keywords.

Keywords:

Name	Description	Units	Default	Area
VTO	pinch-off voltage V_{T0} $(T_{C,VT0})$	V	-2.0	
BETA	transconductance parameter (β)	A/V^2	1.0E-4	*
B	doping tail extending parameter (B)	1/V	0.3	*
ALPHA	saturation voltage parameter $(alpha)$	1/V	2	*
LAMBDA	channel length modulation parameter (λ)	1/V	0	
RD	drain ohmic resistance (R_D)	Ω	0	*
RS	source ohmic resistance (R_S)	Ω	0	*
CGS	zero-bias G-S junction capacitance (C'_{GS})	F	0	*
CGD	zero-bias G-D junction capacitance (C'_{GD})	F	0	*
PB	gate junction potential $(V_{\rm BI})$	V	1	
KF	flicker noise coefficient (K_F)	-	0	
AF	flicker noise exponent (A_F)	-	1	
FC	coefficient for forward-bias depletion capacitance formula	-	0.5	

k	Boltzman's constant	$1.3806226\,10^{-23}$ J/K
q	electronic charge	$1.6021918\,10^{-19}$ C

Standard Calculations

Absolute temperatures (in kelvins, K) are used. The thermal voltage

$$V_{\rm TH} = \frac{kT_{\rm NOM}}{q}. \tag{7.556}$$

and the band gap energy at the nominal temperature is

$$E_G(T_{\rm NOM}) = E_G(0) - 0.000702\frac{4T_{\rm NOM}^2}{T_{\rm NOM} + 1108}. \tag{7.557}$$

Here $E_G(0)$ is the parameter `EG` — the band gap energy at 0 K.

Temperature Dependence

Temperature effects are incorporated as follows where T and $T_{\rm NOM}$ are absolute temperatures in Kelvins (K).

$$\beta(T) = \beta 1.01^{(T_{C,\beta}(T - T_{\rm NOM}} \tag{7.558}$$

$$I_S(T) = I_S e^{\left(E_g(T)\frac{T}{T_{\rm NOM}} - E_G(T)\right)/(nV_{\rm TH})} \left(\frac{T}{T_{\rm NOM}}\right)^{(X_{TI}/n)} \tag{7.559}$$

$$C'_{GS}(T) = C_{GS}\left\{1 + M\left[0.0004(T - T_{\rm NOM}) + \left(1 - \frac{V_{BI}(T)}{V_{BI}}\right)\right]\right\} \tag{7.560}$$

$$C'_{GD}(T) = C_{GD}\left\{1 + M\left[0.0004(T - T_{\rm NOM}) + \left(1 - \frac{V_{BI}(T)}{V_{BI}}\right)\right]\right\} \tag{7.561}$$

$$E_G(T) = E_G(0) - 0.000702\frac{4T_{\rm NOM}^2}{T_{\rm NOM} + 1108} \tag{7.562}$$

$$V_{BI}(T) = V_{BI}\frac{T}{T_{\rm NOM}} - 3V_{\rm TH}\ln\left(\frac{T}{T_{\rm NOM}}\right) + E_G(T_{\rm NOM})\frac{T}{T_{\rm NOM}} - E_G(T) \tag{7.563}$$

$$V_{T0}(T) = V_{T0} + T_{C,VT0}(T - T_{NOM}) \tag{7.564}$$

$$V_{\rm TH} = \frac{kT}{q} \tag{7.565}$$

Parasitic Resistances

The resistive parasitics R_S, R_G and R_D are are calculated from the sheet resistivities `RS` ($= R'_S$), `RG` ($= R'_G$) and `RD` ($= R'_D$), and the *Area* specified on the element line.

$$R_D = R'_D Area \tag{7.566}$$

$$R_G = R'_G Area \tag{7.567}$$

$$R_S = R'_S Area \tag{7.568}$$

The parasitic resistance parameter dependencies are summarized in figure 7.86.

PROCESS PARAMETERS	
`RD`	R'_D
`RG`	R'_G
`RS`	R'_S

+

GEOMETRY PARAMETERS
Area

→

DEVICE PARAMETERS
$R_D = f(Area, R'D)$
$R_G = f(Area, R'G)$
$R_S = f(Area, R'S)$

Figure 7.86: MOSFET parasitic resistance parameter relationships.

Leakage Currents

Current flows across the normally reverse biased gate-source and gate-drain junctions. The gate-source leakage current

$$I_{GS} = Area\, I_S e^{(V_{GS}/V_{\rm TH} - 1)} \tag{7.569}$$

and the gate-drain leakage current

$$I_{GD} = Area\, I_S e^{(V_{GD}/V_{\rm TH} - 1)} \tag{7.570}$$

The dependencies of the parameters describing the leakage current are summarized in figure 7.87.

I/V Characteristics

The current/voltage characteristics are evaluated after first determining the mode (normal: $V_{DS} \geq 0$ or inverted: $V_{DS} < 0$) and the region (cutoff, linear or saturation) of the current (V_{DS}, V_{GS}) operating point.

Normal Mode: ($V_{DS} \geq 0$)

The regions are as follows:

$$\begin{array}{ll} \text{cutoff region:} & V_{GS}(t-\tau) < V_{T0} \\ \text{linear region:} & V_{GS}(t-\tau) > V_{T0} \text{ and } V_{DS} \leq 3/\alpha \\ \text{saturation region:} & V_{GS}(t-\tau) > V_{T0} \text{ and } V_{DS} > 3/\alpha \end{array}$$

Then

$$I_{DS} = \begin{cases} 0 & \text{cutoff region} \\ Area\,\beta\,(1+\lambda V_{DS})\,\dfrac{[V_{GS}(t-\tau)-V_{T0}]^2}{1+B[V_{GS}(t-\tau)-V_{T0}]}\text{Ktanh} & \text{linear and saturation regions} \end{cases} \tag{7.571}$$

where

$$\text{Ktanh} = \begin{cases} 1-\left(1-V_{DS}\frac{\alpha}{3}\right)^3 & \text{linear region} \\ 1 & \text{saturation regions} \end{cases} \tag{7.572}$$

is a taylor series approximation to the tanh function. *Inverted Mode: ($V_{DS} < 0$)*

In the inverted mode the MOSFET I/V characteristics are evaluated as in the normal mode (7.571) but with the drain and source subscripts exchanged.

The relationships of the parameters describing the I/V characteristics of the model are summarized in figure 7.88.

Capacitances

PROCESS PARAMETERS		+	GEOMETRY PARAMETERS	→	DEVICE PARAMETERS
IS	I_S		*Area*		$I_{GS} = f(I_S, Area)$ $I_{GD} = f(I_S, Area)$

Figure 7.87: GASFET leakage current parameter dependencies.

The drain-source capacitance

$$C_{DS} = Area\, C'_{DS} \tag{7.573}$$

The gate-source capacitance

$$C_{GS} = Area \left[C'_{GS} F_1 F_2 \left(1 - \frac{V_{\rm new}}{V_{BI}} \right)^{-\frac{1}{2}} + C'_{GD} F_3 \right] \tag{7.574}$$

The gate-source capacitance

$$C_{GD} = Area \left[C'_{GS} F_1 F_3 \left(1 - \frac{V_{\rm new}}{V_{BI}} \right)^{-\frac{1}{2}} + C'GD F_2 \right] \tag{7.575}$$

where

$$F_1 = \frac{1}{2} \left\{ 1 + \frac{V_{\rm eff} - V_{T0}}{\sqrt{(V_e - V_{T0})^2 + \delta^2}} \right\} \tag{7.576}$$

$$F_2 = \frac{1}{2} \left\{ 1 + \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.577}$$

$$F_3 = \frac{1}{2} \left\{ 1 - \frac{V_{GS} - V_{GD}}{\sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}}} \right\} \tag{7.578}$$

$$V_{\rm eff} = \frac{1}{2} \left\{ V_{GS} + V_{GD} + \sqrt{(V_{GS} - V_{GD})^2 + \alpha^{-2}} \right\} \tag{7.579}$$

$$\tag{7.580}$$

$$V_{\rm new} = \begin{cases} A_1 & A_1 < V_{\rm MAX} \\ V_{\rm MAX} & A_1 \geq V_{\rm MAX} \end{cases} \tag{7.581}$$

and

$$A_1 = \frac{1}{2} \left[V_e + V_{T0} + \sqrt{(V_e + V_{T0})^2 + \delta^2} \right] \tag{7.582}$$

In the model δ and $V_{\rm MAX}$ are not settable by the user. Empirically they were determined to be

$$V_{\rm MAX} = 0.5 \qquad\qquad delta = 0.2$$

PROCESS PARAMETERS	
ALPHA	α
B	B
BETA	β
LAMBDA	λ
VTO	V_{T0}

+

GEOMETRY PARAMETERS
$Area$

→

DEVICE PARAMETERS
$\{I_{DS} = f(Area, \alpha, B, \beta, \lambda, V_{T0})\}$

Figure 7.88: LEVEL 2 (Raytheon model) I/V dependencies.

The capacitance parameter dependencies are summarized in figure 7.89.

AC Analysis

The AC analysis uses the model of figure **??** with the capacitor values evaluated at the `DC` operating point with

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \tag{7.583}$$

and

$$R_{DS} = \frac{\partial I_{DS}}{\partial V_{DS}} \tag{7.584}$$

Noise Analysis

The MOSFET noise model accounts for thermal noise generated in the parasitic resistamces and shot and flicker noise generated in the drain source current generator. The rms (root-mean-square) values of thermal noise current generators shunting the four parasitic resistance R_D, R_G and R_S are

PROCESS PARAMETERS	
`ALPHA`	α
`CGD`	C'_{GD}
`CGS`	C'_{GS}
`CDS`	C'_{DS}
`VBI`	V_{BI}
`VTO`	V_{T0}
`M`	B

+

GEOMETRY PARAMETERS
$Area$

→

DEVICE PARAMETERS
$\{C_{DS} = f(Area, C'_{DS})\}$
$\{C_{GD} = f(Area, C'_{GD}, \alpha, B, F_C, V_{BI}, V_{T0})\}$
$\{C_{GS} = f(Area, C'_{GS}, \alpha, B, F_C, V_{BI}, V_{T0})\}$

Figure 7.89: Capacitance dependencies.

$$I_{n,D} = \sqrt{4kT/R_D}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.585}$$
$$I_{n,G} = \sqrt{4kT/R_G}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.586}$$
$$I_{n,S} = \sqrt{4kT/R_S}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.587}$$

Shot and flicker noise are modeled by a noise current generator in series with the drain-source current generator. The rms value of this noise generator is

$$I_{n,DS} = \sqrt{I^2_{\mathrm{SHOT},DS} + I^2_{\mathrm{FLICKER},DS}} \tag{7.588}$$

$$I_{\mathrm{SHOT},DS} = \sqrt{4kTg_m\frac{2}{3}}\quad \mathrm{A}/\sqrt{\mathrm{Hz}}\ \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.589}$$
$$I_{\mathrm{FLICKER},DS} = \sqrt{\frac{K_F I_{DS}^{A_F}}{f}}\quad \mathrm{A}/\sqrt{\mathrm{Hz}} \tag{7.590}$$

where the transconductance

$$g_m = \frac{\partial I_{DS}}{\partial V_{GS}} \tag{7.591}$$

is evaluated at the `DC` operating point and f is the analysis frequency.

Chapter 8

Examples

8.1 Simple Differential Pair

The following circuit determines the dc operating point of a simple differential pair. In addition, the ac small-signal response is computed over the frequency range 1Hz to 100MEGHz.

```
SIMPLE DIFFERENTIAL PAIR
VCC 7 0 12
VEE 8 0 -12
VIN 1 0 AC 1
RS1 1 2 1K
RS2 6 0 1K
Q1 3 2 4 MOD1
Q2 5 6 4 MOD1
RC1 7 3 10K
RC2 7 5 10K
RE 4 8 10K
.MODEL MOD1 NPN BF=50 VAF=50 IS=1.E-12 RB=100 CJC=.5PF TF=.6NS
.AC DEC 10 1 100MEG
.END
```

8.2 MOS Output Characteristics

The following file computes the output characteristics of a MOSFET device over the range 0-10V for VDS and 0-5V for VGS.

```
MOS OUTPUT CHARACTERISTICS
VDS 3 0
VGS 2 0
M1 1 2 0 0 MOD1 L=4U W=6U AD=10P AS=10P
.MODEL MOD1 NMOS VTO=-2 NSUB=1.0E15 UO=550
* VIDS MEASURES ID, WE COULD HAVE USED VDS, BUT ID WOULD BE NEGATIVE
VIDS 3 1
.DC VDS 0 10 .5 VGS 0 5 1
.END
```

8.3 Simple RTL Inverter

The following file determines the dc transfer curve and the transient pulse response of a simple RTL inverter. The input is a pulse from 0 to 5 Volts with delay, rise, and fall times of 2ns and a pulse width of 30ns. The transient interval is 0 to 100ns, with printing to be done every nanosecond.

```
SIMPLE RTL INVERTER
VCC 4 0 5
VIN 1 0 PULSE 0 5 2NS 2NS 2NS 30NS
RB 1 2 10K
Q1 3 2 0 Q1
RC 3 4 1K
.MODEL Q1 NPN BF 20 RB 100 TF .1NS CJC 2PF
.DC VIN 0 5 0.1
.TRAN 1NS 100NS
.END
```

8.4 Adder

The following file simulates a four-bit binary adder, using several subcircuits to describe various pieces of the overall circuit.

```
ADDER - 4 BIT ALL-NAND-GATE BINARY ADDER
*** SUBCIRCUIT DEFINITIONS
.SUBCKT NAND 1 2 3 4
*   NODES:  INPUT(2), OUTPUT, VCC
Q1 9 5 1 QMOD
D1CLAMP 0 1 DMOD
Q2 9 5 2 QMOD
D2CLAMP 0 2 DMOD
RB 4 5 4K
R1 4 6 1.6K
Q3 6 9 8 QMOD
R2 8 0 1K
RC 4 7 130
Q4 7 6 10 QMOD
DVBEDROP 10 3 DMOD
Q5 3 8 0 QMOD
.ENDS NAND
.SUBCKT ONEBIT 1 2 3 4 5 6
*   NODES:  INPUT(2), CARRY-IN, OUTPUT, CARRY-OUT, VCC
X1 1 2 7 6 NAND
X2 1 7 8 6 NAND
X3 2 7 9 6 NAND
X4 8 9 10 6 NAND
X5 3 10 11 6 NAND
X6 3 11 12 6 NAND
X7 10 11 13 6 NAND
X8 12 13 4 6 NAND
X9 11 7 5 6 NAND
.ENDS ONEBIT
.SUBCKT TWOBIT 1 2 3 4 5 6 7 8 9
*   NODES:  INPUT - BIT0(2) / BIT1(2), OUTPUT - BIT0 / BIT1,
*           CARRY-IN, CARRY-OUT, VCC
X1 1 2 7 5 10 9 ONEBIT
X2 3 4 10 6 8 9 ONEBIT
.ENDS TWOBIT
.SUBCKT FOURBIT 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
*   NODES:  INPUT - BIT0(2) / BIT1(2) / BIT2(2) / BIT3(2),
*           OUTPUT - BIT0 / BIT1 / BIT2 / BIT3, CARRY-IN, CARRY-OUT,
VCC
X1 1 2 3 4 9 10 13 16 15 TWOBIT
X2 5 6 7 8 11 12 16 14 15 TWOBIT
.ENDS FOURBIT
*** DEFINE NOMINAL CIRCUIT
.MODEL DMOD D
.MODEL QMOD NPN(BF=75 RB=100 CJE=1PF CJC=3PF)
VCC 99 0 DC 5V
VIN1A 1 0 PULSE(0 3 0 10NS 10NS   10NS   50NS)
VIN1B 2 0 PULSE(0 3 0 10NS 10NS   20NS  100NS)
VIN2A 3 0 PULSE(0 3 0 10NS 10NS   40NS  200NS)


VIN2B 4 0 PULSE(0 3 0 10NS 10NS   80NS  400NS)
VIN3A 5 0 PULSE(0 3 0 10NS 10NS  160NS  800NS)
```

Descriptions of the basic algorithms of SPICE.

Bibliography

[1] McCalla

Descriptions of MOSFET models.

[2] P. Antognetti and G. Massobrio (editors), *Semiconductor Device Modeling with SPICE*, McGraw-Hill: New York, 1988.

[3] H. Shichman and D. A. Hodges, "Modeling and simulation of insulated-gate field-effect transistor switching circuits," *IEEE Journal of Solid-State Circuits*, Vol. 3, September 1968, Page 285.

[4] J. E. Meyer, "MOS models and circuit simulation," *RCA Review*, Vol 32., 1971.

[5] D. E. Ward and R. W. Dutton, "A channel-oriented model for MOS transistors capacitances," *IEEE J. Solid-State Circuits*, Vol. 13, 1978.

[6] S. Y. Oh, D. E. Ward and R. W. Dutton, "Transisent analysis of MOS transistors," *IEEE Trans. Electron Devices*, Vol. 27, No. 8, 1980.

[7] A. Vladimirescu and S. Liu, *The simulation of MOS transistor integrated circuits using SPICE2*, University of California, Berkeley, Memorandum No. M80/7, February 1980.

[8] K. Lee, M. Shur, T. Fjeldly and T. Ytterdal, *Semiconductor Device Modeling for VLSI*, Prentice Hall: Englewood Cliffs, New Jersey, 1993.

[9] B. J. Sheu, D.L. Scharfetter, P.-K. Ko, and M.-C. Jeng, "BSIM: Berkeley Short-Channel IGFET model for MOS transistors," *IEEE Journal of Solid-State Circuits*, Vol 22, August 1987, pp. 558-566.

[10] J. R. Pierret, " *A MOS Parameter extraction program for the BSIM model*, University of California, Berkeley, Memorandum No. M84/99, November 1984.

Descriptions of MESFET models.

[11] W. R. Curtice, "A MESFET model for use in the design of GaAs integrated circuits," *IEEE Transactions on Microwave Theory and Techniques,* Vol. 28, pp. 448-456, 1980.

[12] W.R. Curtice and M. Ettenberg, "A nonlinear GaAs FET model for use in the design of output circuits for power amplifiers," *IEEE Trans. on Microwave Theory and Tech.*, Vol. MTT-33, Dec. 1985, pp. 1383-1393.

[13] T. Kacprzak and A. Materka, "Compact dc model of GaAs FET's for large-signal computer calculation," *IEEE Journal of Solid-State Circuits*, Vol. SC-18, April 1983, pp. 211-213.

[14] S.E. Sussman-Fort, "On the basic algorithms of SPICE with application to microwave circuit simulation," *Presented at Nonlinear CAD and Modeling Workshop*, IEEE International Microwave Symp. 1987

[15] S.E. Sussman-Fort, S. Narasimhan, and K. Mayaram, "A complete GaAs MESFET computer model for SPICE," *IEEE Trans. Microwave Theory and Tech.*, Vol. MTT-32, April 1984, pp. 471-473.

[16] S.E. Sussman-Fort, J.C. Hantgan, and F.L. Huang, "A SPICE model for enhancement- and depletion-mode GaAs FET's," *IEEE Trans. Microwave Theory Tech.*, Vol. MTT-34, Nov. 1986, pp. 1115-1118.

[17] J.M. Golio, J.R. Hauser, and P.A. Blakey, "A large-signal GaAs MESFET model implemented on SPICE," *IEEE Circuits and Devices Magazine*, Sep. 1985, pp. 21-30.

[18] J.M. Golio, P.A. Blakey, and R.O. Grondin, "A general CAD tool for large-signal GaAs MESFET circuit design," *IEEE MTT-S International Microwave Symposium Digest*, June 1985, pp. 417-420.

[19] Angelov:Zirath

[20] J.M. Golio, J.R. Hauser and P.A. Blakey "A large-signal GaAs MESFET model implemented on SPICE," *IEEE Circuits and Devices Magazine*, Sept. 1985, pp. 21-30.

[21] J.M. Golio, P.A. Blakey, and R.O. Grondin, "A general CAD tool for large-signal GaAs MESFET circuit design," *IEEE MTT-S Digest*, 417-420,.

[22] P.H. Ladbrooke, "Reverse modelling of GaAs MESFETs and HEMTs," *GEC Journal of Research*, Vol. 6, 1988, pp. 1-9.

[23] H. Statz, P. Newman, I. .W. Smith, R. A. Pucel and H. A. Haus, "GaAs FET Device and Circuit Simulation in SPICE," *IEEE Transactions on Electron Devices*, Vol. 34, pp 160–169, 1987.

[24] A. J. McCamant, G. D. McCormack, and D. H. Smith, "An improved GaAs MESFET Model for SPICE," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 38, pp. 822-824, June 1990.

Descriptions of transformer model.

[25] D. C. Jiles and D. L. Atherton, "Theory of ferromagnetic hysteresis," *Journal of Magnetism and Magnetic Materials*, Vol. 61, 1986. pp. 48

[26] D. Divekar, "Comments no 'GaAs FET devices and circuit simulation ni SPICE,' *IEEE Trans. on Electron Devices*, Vol. 34, December 1987, p. 2564.

[27] I.W. Smith, H. Statz, H.A. Haus, and R.A. Pucel, "On charge nonconservation in FETs," *IEEE Trans. on Electron Devices*, Vol. 34, December 1987, pp.2565-2568.

Miscellaneous

[28] D. Knuth, *The Art of Computer programming*, Vol. 2.

[29] S. M. Sze, *Physics of Semiconductor Devices*, 2nd Edition, John Wiley and Sons: New York, 1981

[30] S. M. Sze, *High-Speed Semiconductor Devices*, John Wiley and Sons: New York, 1990.

Appendix A

Error

A.1 Introduction

SPICE errors can result from errors in syntax, errors in the wiring of the original circuit resulting in the circuit not being solvable, and convergence errors during analysis that prevent continuation of analysis.
then we have not specified our circuit as drawn. In this case, we also leave one terminal of the resistor unconnected to anything else and SPICE detects the error and reports it in the output file:

```
0*ERROR*:  LESS THAN 2 CONNECTIONS AT NODE      2
```

In a complex circuit it is always easy to get one node number wrong on one element but leave all of the nodes connected to two or more elements. In this case SPICE might detect no errors. If the output looks 'wrong' for any reason, the first thing to do is to draw your circuit by looking at the SPICE file as written and check that against your intended circuit.

Another important thing to remember about error messages is that SPICE is not very good at drawing attention to them.

More expensive commercial versions of SPICE are much more user friendly but still maintain full SPICE upwards compatibility by reporting all errors in the traditional SPICE way as well. SPICE output files tend to be long and are cryptic looking. Error and Warning messages can be found almost anywhere within them. If an error has occured it may be necessary to examine the entire output file to determine exactly what caused the error. Mainstream commercial versions of SPICE have somewhat better error reporting than less expensive commercial versions and the public domain versions.

In the following we list errors common to most versions of SPICEas they derive from SPICE2G6. Common errors for some of the commercial versions of SPICE are also included. The errors are arranged in alphabetic order. Each commercial version of SPICE supports additional elements that are particular to tha particular version but generally are selfexplanatory. For errors messages indicating a particular element the syntax of the element described in the element catalog chapter (chapter 7).

A.2 List of Errors

`at least two numeric values required`
Not enough numeric values are provided. See the element form for the element.

`cannot use LIST with DEC or OCT sweeps`
See statement for correct form.

`conflicting length`
With a transmission line element either the time delay `TD` and the reference frequency `F` are both specified or the time delay `TD` and the normalized electrical length `NL` are both specified. if the parameter pairs are specified then there are two possible electrical lengths. See the input forms of the `T` element on page 242.

`conflicting specifications`
With an independent current source `I` or independent voltage source`V` element two or more transient types are specified. See the allowable element line forms on page 168 for the `I` element and on page 246 for the `V` element.

`contrary parameters`
See page **??** for description of this element.

`coupling coefficient out of range.`
The coupling coefficient for a K element must be between 0 amd 1.

`digital files option not present`
The digital files (registered) option must be purchased separately. SPICE version dependent.

`ERROR -- ... does not match nodes of ...`
Commonly this is because the number of nodes of a subcircuit does not match the number of nodes of a subcircuit call (`X` element).

`ERROR: CPU Time limit exceeded`
The CPU time specified by the `.OPTIONS` parameter `CPTIME`, or its default, has been exceeded.

`ERROR: less than 2 connections at node nnnn`
Every node must have at least two connections or elese a node is left floating and the voltage at the node can not be determined. This may be either because the node is floating and the voltage at the node is indeterminate or else because the numerical techniques used require it.

`ERROR: model ... referenced by ... is undefined`
A model was referenced but the actual model was never specified via a `.MODEL` statement.

`ERROR: Node is floating`
This is either because this is only one connection between this node or there is no `DC` path from this node to ground as required in determining the `DC` voltage at the node.

`ERROR: subcircuit ... is undefined`
A subcircuit was referenced but the actual subcircuit was never specified via a `.SUBCKT` statement.

`ERROR: transient analysis iterations`
The number of transient iterations specified by the `.OPTIONS` parameters `ITL4` or `ITL5`, or their defaults, has been exceeded.

`ERROR: voltage loop`
Voltage sources and/or elements such as inductors or transmission lines that are modeled using controlled voltage sources, are arranged in a loop. This results in a modified nodeal admittance matrix that can not be solved.

`ERROR: voltage source ... which controls switch ... is undefined`
A voltage source was referenced but the actual element was never specified.

`expand: parameter syntax error for ...`
Error occurred during subcircuit expansion in handling parameters. With some version of SPICE parameters are supported. Parameters must be in the form *Keyword = Value* where *Value* may be a numeric value or an expression. Either the Keyword is missing or is not an alphanumeric quantity, or Value is missing or is neither a numeric quantity nor an expression that evaluates to a numeric quantity. The error is either on the `X` element line or on the `.SUBCKT` statement.

`Expression evaluation error: function syntax error`
Error in expression evaluation or input prevents continuation.

`Expression evaluation error: syntax error`
Error in expression evaluation or input prevents continuation.

`Expression evaluation error: undefined parameter`
Error in expression evaluation or input prevents continuation.

`Error in expression.`
Error in expression evaluation or input prevents continuation.

`Expression syntax error.`
some versions of SPICE support expressions. The expression is syntactically incorrect or other error that prevents evaluation of the expression.

`extra fields`
Extra quantities on element line or statement that were not used.

`function syntax error in expression.`
Error in expression evaluation or input prevents continuation.

`incomplete range`
A range was indicated but is incomplete.

`I(node) is not valid`
To specify currents a voltage source must be indicated. Specifying the current at a node is meaningless.

`incorrect number of parameters`
Not enough parameters specified for this element.

`inductor: mutual coupling requires two (or more) inductors`
A K element must comprise two or more inductors. This element is SPICE version dependent.

`inductor part of another CORE device`
An inductor is specified as a component of two different K elements An inductor can only be part of one K element. This element is SPICE version dependent.

`inductor part of another K element`
An inductor is specified as a component of two different K elements An inductor can only be part of one K element. This element is SPICE version dependent.

`inductor part of another mutual coupling device`
An inductor is specified as a component of two different K elements An inductor can only be part of one K element. This element is SPICE version dependent.

`inductor part of this K device`
A K element contains two inductors of the same name. An inductor can only be specified once in a K element inductor list. This element is SPICE version dependent.

`invalid .WIDTH card`
The parameters on the `.WIDTH` card are incorrectly specified or a parameter is not supported by this version of SPICE.

`invalid analysis type`
Analysis type specified on a .FOUR .TF .NOISE .SENS .MC .PLOT .PRINT .PROBE statement is incorrect. Supported analysis types include AC, DC, TRAN and NOISE although this list is SPICE version dependent. See the full description of the allowable analysis types for the statement.

`invalid card`
The statement may be spelled incorrectly or this version of SPICE does not recognize this statement.

`invalid device`
The first letter of an element card indicates the particular device being referred to. Not all versions of SPICE support the same set of elements and here an non-supported element is being invoked.

`invalid device in subcircuit`
Generally any element may be used with a subcircuit. (between `.SUBCKT` and `.ENDS`. Some versions of SPICE support special elements or forms of elements that can only be used at the top level circuit. See the description of the element.

`invalid dimension`
The degree (dimension or order) of a polynomial must be more than 0 and less than the max polynomial order supported by the current version of SPICE.

`invalid function`
Some commercial versions of SPICE support function evaluations. An unsupported function is being used.

`invalid increment`
Usually caused by specifying an increment of zero on a .DC statement

`invalid node number in .SUBCKT statement`
The invalid node number was specified in .SUBCKT statement. This is usually because the ground node (either 0 or, in some versions of SPICE "`GND`") was specified in the list of nodes in the `.SUBCKT` statement.

`invalid number`
Due to a number being of the wrong sign or too small or zero where this is not valid. Possibly an integer was expected and a floating point number was supplied.

`invalid option`
The value of the parameter is invalid.

`invalid outside of .SUBCKT`
.ENDS is used out of context and does not terminate a subcircuit.

`invalid parameter`
Invalid parameter on element line. Probably misspelled or not supported by this version of SPICE.

`invalid parameter in model`
Either the parameter is misspelled ot it is not supported by this version of SPICE.

`invalid port name for transmission line`
Port name in output list is either missing or invalid.

`invalid print interval`
See the analysis statement for the requirements on specifying the output reporting interval.

`invalid run number`
A negative number of runs was specified in a Monte Carlo analysis.

`invalid specification`
Error in specifying element. See the form for this element.

`invalid step size`
Step size must be positive.

`invalid sweep type`
Sweep type not supported by this version of SPICE or sweep type incorrectly specified.

`invalid value`
Due to a value being of the wrong sign or too small or zero where this is not valid. Possibly an integer was expected and a floating point number was supplied.

`last PWL pair incomplete`
The piecewise linear characteristic of an `I` or `V` element must be specified in time,value pairs.

`list of runs`
No runs were specified in a Monte Carlo analysis

`missing .ENDS in .SUBCKT`
A subcircuit must end with a `.ENDS` statement. This was missing.

`missing analysis type`
Analysis type must be specified in a .FOUR .TF .NOISE .SENS .MC .PLOT .PRINT .PROBE statement but is missing.

`missing component value`
Component value for R L or C element missing

`missing control node`
A control node is missing for an `E` element.

`missing controlling source`
The name of a controlling voltage source is missing for a `G` element.

`missing device or node`
device or node expected but missing.

`missing dimension`
The degree (dimension or order) of a polynomial not specified. The degree of a polynomial should be specified in the form "`POLY(`n`)`" or "`POLY` n ".

`missing file name`
File name expected in a `.INCL` but missing.

`missing frequency`
Frequency missing on a .FOUR card.

`missing gain`
The gain must be specified for an `E` or `F` element.

`missing inductor`
In reading K element line expected to read in the name of an inductor but it was missing.

`missing INOISE or ONOISE`
.NOISE statement and either INOISE or ONOISE parameters required or illegal parameter.

`missing model`
Model Name expected on element line but was missing

`missing model name`
Model Name expected on element line but was missing

`missing name`
Subcircuit name expected on `.SUBCKT` (or similar) statement but not found.

`missing node`
A node number expected but not provided. Possibly not enough nodes specified. See the form for this element in the element catalog (chapter 7.

`missing node list`
No nodes are specified.

`missing or invalid model name or type`
Either the model name is missing or it is not a valid type for this element.

`missing or invalid value`
Due to a value not being supplied where it is expected, being of the wrong sign or too small or zero where this is not valid. Possibly an integer was expected and a floating point number was supplied.

`missing output variables`
No output variables specified for a .FOUR .TF .NOISE .SENS .MC .PLOT .PRINT .PROBE statement.

`missing parameter`
Parameter expected but not found.

`missing pnr`
In a .AS statement the port number is missing.

`missing polynomial`
The `POLY` keyword was specified but no polynomial coefficients were found.

`missing run count`
The number of Monte Carlo runs is incorrectly specified.

`missing second port in S(pnr1,pnr2)`
There must be a second port in a .AS output specification.

`missing second node in V(node1,node2)`
In this syntax two nodes must be specified.

`missing seed`
A random number seed was expected for this element but it was missing

`missing source`
Name of source expected but it was not supplied.

`missing subcircuit name`
Subcircuit name expected on `.SUBCKT` statement but not found.

`missing sweep type`
Sweep type not specified for `.AC analysis`.

`missing transconductance`
The transconductance must be specified for a `G` element.

`missing transresistance`
The transresistance must be specified for an `H` element.

`missing value`
Value (or expression) expected but not found. Either a non-numeric quantity (i.e. not a number) is in a location where a numeric value value is expected or a quantity is missing. If an expression was specified it was either incorrectly delimited or its evaluation is not supported by this version of SPICE.

`must be >= 1`
An positive integer value was expected upon input but the value was not 1 or more.

`must be a two terminal device`
To specify a current a two terminal device must be indicated. More elements with more than three terminals the edge current cannot be uniquely identified by specifying the nodes.

`must be a voltage source name`
Name of voltage source expected on element line but not supplied A voltage source name must begin with V.

`must be an inductor`
In reading K element line expected to read in the name of an inductor but the name did not begin with 'L'

`must be I or V`
In a .AC, .FOUR .TF .SENS .MC .PLOT .PRINT .PROBE statement. Only I or V can be specified. Something else is in output list. Generally this is because only `I` or `V` elements can be swept.

`must be independent source (I or V)`
An element was specified but it was not an I or V element.

`must be monotonically increasing or decreasing`
Quantities in list must be monotonically increasing or decreasing.

`must be S`
The keyword "S" expected.

`must be V`
A list of voltage sources is required but either a numeric value was provided an element other than a voltage source specified. (The voltage sources in list of must begin with `V`.) In the case of a polynomial specification the number of controlling voltage sources must be equal to the polynomial degree previously specified on the element line. Either not enough voltage sources are specified or a non-voltage source is specified.

`name on .ENDS does not match .SUBCKT`
An optional name may be included on a `.ENDS` statement. If specified it must match the name specified on the matching `.SUBCKT` statement.

`nesting level exceeded`
The number of files that can be included using `.INCL` is limited. The limit is SPICE version specific but is typically around 5.

`node's voltage already set`
Two attempts have been made to set the initial voltage at a node. The initial value may be specified using either a `.NODESET` statement or using the initial condition `IC` parameter on some elements.

`not a valid parameter for model type`
parameter not recognized for this model.

`not unique`
Some versions of SPICEallow abbreviated forms of statements. The minimum allowable abbreviation must be unique. The abbreviation used in the netlist is not unique and is an abbreviation of two or more statements.

`only .AC .DC and .TRAN valid`
The type of analysis in a Monte Carlo run must be either `.DC`

, `.AC` or `.TRAN`.

`only .MODEL valid in subcircuit`
`.MODEL` is the only statement allowed within a subcircuit description (between `/SUBCKT` and `.ENDS`.

`only one .TEMP and .DC TEMP allowed`
One .TEMP statement and one .DC TEMP statement allowed but not both. This is SPICE version dependent.

`Parameter syntax error`
With some version of SPICE parameters are supported. The "`PARAMS:`" keyword indicates that parameters are to be specified in the form *Keyword* = *Value* where *Value* may be a numeric value or an expression. Either the Keyword is missing or is not an alphanumeric quantity, or Value is missing or is neither a numeric quantity nor an expression that evaluates to a numeric quantity..

`PNR already defined`
The port number can only be defined once.

`PNR missing or invalid`
The port number was either missing or not correctly specified.

`run count`
The number of Monte Carlo runs is incorrectly specified.

`run count must be > 1`
The number of Monte Carlo runs is incorrectly specified.

`syntax error.`
Input is in error. Often due to an non-numeric value in input where a non-numeric value expected or vice-versa.

`syntax error in expression.`
Error in expression evaluation or input prevents continuation.

`TD or F must be specified`
With a transmission line element either the time delay `TD` or reference frequency `F` must be specified. See the `T` element on page 242.

`temperature`
A 0 K (Kelvin) temperature is not valid. The usual problem is that a 0 Celsius temperature was specified but a temperature specified by a `.temp` statement must be an absolute temperature (in Kelvin)Q

`time must be increasing`
In specifying the transient behavior of an `I` or `V` element times must be increasing.

`time must not be negative`
In specifying the transient behavior of an `I` or `V` element a negative time was specified.

`too many coefficients`
The number of polynomial coefficients specified exceeds that supported in this version of SPICE.

`too many inductors`
There is a limit on the number of inductors per K element. This limit has been exceeded.

`too many tolerances`
The number of tolerances that may be specified is limited. This limit is SPICE version dependent.

`TooMany`
Too many parameters, values or nodes on element line or `.SUBCKT` statement.

`unable to open file`
File specified in a `.INCL` or `.LIB` does not exist in the current directory or default directories.

`undefined parameter`
An unsupported parameter keyword specified. Either this version of SPICE does not support this parameter for this element or statement or the parameter is misspelled.

`unknown parameter`
A parameter was used on an element line or in a `.MODEL` statement which was not recognized. Either this version of SPICE does not support this parameter or the parameter is misspelled.

`value may not be 0`
A non-zero value expected.

`voltage source name`
Name of voltage source expected on element line but not supplied

`WIDTH must be 80 or 132`
A SPICE supports two output log formats that are either 80 columns or 132 columns wide. A width other than 80 or 132 was specified.

`Z0 must be specified`
With a transmission line element the `Z0` parameter must always be input using the syntax `Z0=`*CharacteristicImpedance*.

Index