

SPICE Short Course

By Dr. Muhammad Elrabaa

- What is in the name?

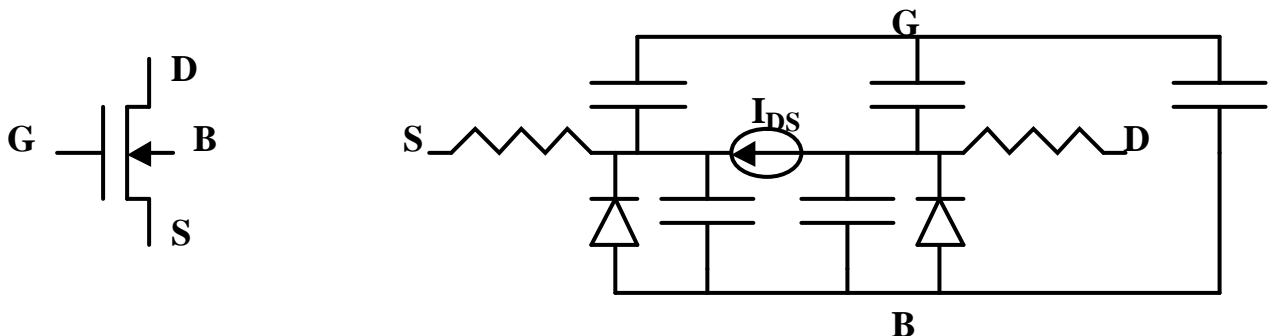
Simulation Program with Integrated Circuits Emphasis = SPICE

- What does it do?

SPICE is used to simulate (i.e. solve for currents and voltages) electronic circuits and is especially suited for integrated circuits.

- How does it do that?

SPICE replaces each element (MOSFET transistors, BJTs, Diodes) in the circuit with a simpler model that only contains resistors, capacitors, dependent current and voltage sources and then solve for currents and voltages using Kirchoff's current and voltage laws. The values of these resistors, capacitors, and dependent sources are calculated using the device parameters supplied by the user. The figure below shows the SPICE model of a MOSFET transistor:



The equivalent circuit model of a MOSFET Transistor in SPICE

- **What is the input to SPICE?**

The input to SPICE is a text file containing the circuit description, the required analysis, and the required output. The first line in the input file **MUST contain the circuit name**. SPICE always takes the first line to be the circuit's name, so no other thing should be there. **Also the last line must have the .END** command. All the other things go between these two. Comments can be placed in any line that starts with *****. Comments should be used to enhance the readability of the SPICE file.

- I. The **circuit description** section, also called the *netlist*, contains the circuit elements (transistors, resistors, capacitors, diodes ...etc.) as well as the power supplies and input stimuli. The syntax for different circuit elements is shown below. Note that in SPICE3, node names can be strings and not necessarily numbers as in old SPICE. This should be utilized for greater readability of the SPICE netlist (node names should have meanings). If a line gets too long it can be continued on the next line with a **+** placed at the beginning of the 2nd line.

Resistors: Rxxxx 1st_node 2nd_node *value*

Capacitors: Cxxxx 1st_node 2nd_node *value*

Inductors: Lxxxx 1st_node 2nd_node *value*

Mutual Inductors: Kxxxx Lyyyy Lzzzzz *value*

Diodes: **Dxxxx** Anode_Node Cathod_node *model_name*



The *model_name* is the name of model specifying the parameters for SPICE to calculate the diode characteristics.

The **syntax** for that is:

.model *model_name* **D** (Type of model: Diode
model parameters ...
+ *continued model parameters ...*)
Line continuation

Bipolar Transistors (BJTs):

Qxxx Collector_node Base_node Emitter_node *model_name*

Again, the syntax for the model is:

.model *model_name* **NPN** (Type of BJT: **NPN** or **PNP**
BF=75 RB=100 CJE=1PF
+ *CJC=3PF ... continued model parameters ...*)
Line continuation

MOSFET Transistors:

Mxx D_node G_node S_node B_node *mod_name* L=1u W=10u

L and W are the channel length and width, respectively.

Additional size parameters can be added in the statement above

such as AD=... (drain area), AS=... (Source area), PD=... (periphery of drain). The model is again defined as:

Type of MOSFET: **NMOS** or **PMOS**

.model *model_name* **nmos** (LEVEL=3 TPG=+1
 + GAMMA=1.140501 THETA=0.8109787
 + KAPPA=0.1579183 ... *continued model parameters* ...)

SPICE also supports other devices such as JFETs, Transmission lines, MESFETs, and voltage and current controlled switches (refer to the manuals for these).

Independent Sources:

There are many possible independent sources in SPICE. We are going to focus on the ones mostly used.

DC (constant source):

Vxx	+ve_node	-ve_node	[DC] value
Ixx	+ve_node	-ve_node	[DC] value

For transient analysis we have many sources. The most used ones are described below:

Pulsed source (e.g. a clock):

Vxx +ve_n -ve_n pulse(low_value high_value delay rise_time fall_time pulse_width period)

Example: Vclk clk 0 pulse(0 5 0ns 0.1ns 0.1ns 5ns 10ns)

Piece-wise linear source (for arbitrary-shaped transient):

Vxx +ve_n -ve_n pwl(T1 V1 T2 V2 T3 V3)

Ti and Vi are pairs of time and voltage values.

Sinusoidal Source (for transient analysis of analog circuits):

**Vxx +ve_n -ve_n sin(V_offset V_amplitude Frequency
T_delay Damping_factor)**

There are also other transient sources like exponential and frequency modulated sinusoidal.

For AC analysis:

**Vxx +ve_n -ve_n <DC/Tran value> AC <magnitude>
<phase>**

Dependent Sources:

These are used to model linear circuit behavior (macro modeling).

Linear Voltage-Controlled Current Sources $I=value*V$:

**GXX +ve_n -ve_n +ve_controlling_n -ve_controlling_n
VALUE**

Linear Voltage-Controlled Voltage Sources $V=value*V$:

**EXX +ve_n -ve_n +ve_controlling_n -ve_controlling_n
VALUE**

Linear Current-Controlled Current Sources $V=value*I$:

FXX +ve_n -ve_n +ve_controlling_n -ve_controlling_n
VALUE

Linear Current-Controlled Voltage Sources $V=value*I$:

HXX +ve_n -ve_n +ve_controlling_n -ve_controlling_n
VALUE

In addition SPICE has non-linear dependent sources (refer to the manuals).

Subcircuits Definition in SPICE:

When a circuit is used in multiple places it is worth making it a sub-circuit and just call it wherever it is needed. A subcircuit description is enclosed between a **.SUBCKT** and an **.ENDS** commands as shown below:

.SUBCKT subckt_name node1 node2 node3 ...

netlist

.ENDS

The nodes used within the subckt definition are local to that section, i.e. the same names can be used in other subcircuit definitions. The node order in the subcircuit call must match that in the SUBCKT definition. To call a subcircuit:

Xxxxx node1 node2 node3 ... subckt_name

.INCLUDE command

General form:

.INCLUDE filename

.INCLUDE "filename with spaces.cir" If the path contains spaces.

Examples:

```
.INCLUDE \users\spice\common\wattmeter.cir
```

```
.INCLUDE "\users\spice files\wattmeter.cir"
```

Frequently, portions of circuit descriptions will be reused in several input files, particularly with common models and subcircuits. In any SPICE input file, the `.include` line may be used to copy some other file as if that second file appeared in place of the `".include"` line in the original file. There is no restriction on the file name imposed by SPICE beyond those imposed by the local operating system.

.LIB Command

General form:

```
.LIB filename
```

```
.LIB "filename with spaces"
```

Examples:

```
.LIB \users\spice\common\bipolar.lib
```

The `.LIB` line is similar to the `.INCLUDE` line except that the specified file is assumed to contain `.MODEL` and `.SUBCKT` definitions. WinSpice3 searches for any undefined models or subcircuits in the specified file and extracts the required definitions and pastes them into the circuit. The main difference

is that because it only extracts parts of the specified file and does not include the whole file in your circuit, the **.LIB** line uses far less memory. The input file can have any extension, but by convention has the extension **.lib**.

.OPTIONS Command

Various parameters of the simulations available in Spice can be altered to control the accuracy, speed, or default values for some devices. These parameters may be changed via the **.OPTIONS** line:

.OPTIONS OPT1 OPT2 ...

Example: The following suppress the printing of the device model in the output of the simulations.

.OPTIONS NOMOD

- II. The **required analysis** is one of several types that can be performed by SPICE. These include:
1. **Operating point analysis:** In this type of analysis, SPICE would calculate the voltages and currents everywhere for a specific value of input (single point). This is useful in finding the bias point of a circuit.

Syntax: .op

2. **DC sweep analysis:** In this type of analysis, SPICE would calculate the voltages and currents everywhere for a range

of input. For each value of input, SPICE assumes that the circuit has reached steady state. This analysis is useful in obtaining DC transfer characteristics (e.g. output voltage versus input voltage of a gate or an amplifier) as well as the I-V characteristics of transistors. More than one source can be swept.

Syntax:

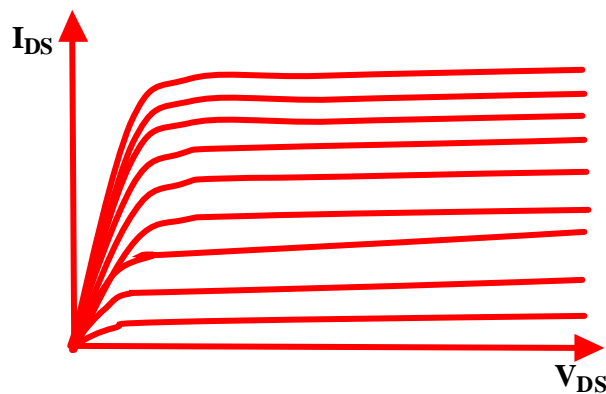
**.DC Source1_name START1_value STOP1_value STEP1_Value
< Source2_name START2_value STOP2_value STEP2_Value >**

For each value of source 2, source 1 is swept for its entire range.

Example: The line below can be used to generate the I-V characteristics of a MOSFET transistor (I_{DS} versus V_{DS} for several values of V_{GS}):

.DC VDS 0 5 0.1 VGS 0 5 0.5

This will produce something like this:



- 3. Transient analysis:** This is the time-dependent simulation. It is used to find the circuit transient behavior in response to an input. This useful in finding circuit's delay, shape of output, settling time, output over and under shoot, and output rise and fall times. Of course for transient analysis to be meaningful, at least one source in the circuit must have a transient behavior (pulse, pwl, sin ...etc.). A **.tran** statement will cause SPICE to generate the node voltages and branch currents as a function of time.

Syntax:

.tran time_step end_time <start_time <TMAX>><UIC>

Time_step is the printing or plotting. End_time is the final time, and start_time the initial time (if it is omitted, it is assumed to be zero). UIC (use initial conditions) direct SPICE not to solve for the quiescent operating point before beginning the transient analysis. If this keyword is specified, WinSpice3 uses the values specified using IC=... on the various elements as the initial transient condition and proceeds with the analysis. If the .IC control line has been specified, then the node voltages on the .IC line are used to compute the initial conditions for the devices. This is very useful when finding out the transients of circuits with feedback (such as oscillators). The syntax for the .IC statement is:

.IC V(NOD1_NAME)=VAL V(NOD2_NAME)=VAL ...

Example:

.IC V(11)=5 V(4)=-5 V(2)=2.2

Hence @ t=0 V(11) is set to 5V, V(4) to -5V ..etc.

4. **AC (small signal) Analysis:** This analysis for linear, steady-state analysis of circuits, usually linear circuits (i.e. analog circuits such as amplifiers). Again at least one source in the circuit must be specified as an AC source. For AC analysis SPICE would calculate the voltages and currents versus frequency (both magnitude and phase). The frequency points can be displayed on a log scale (decades), octave, or linear. The syntax for AC analysis is:

.AC DEC No_of_points_per_decade START_freq End_freq

.AC OCT No_of_points_per_octave START_freq End_freq

.AC LIN No_of_points START_freq End_freq

SPICE also offer other types of analysis such as distortion analysis, sensitivity analysis, transfer-function analysis (refer to the manual for details on these types of analysis).

- III. The **required output:** SPICE can output the results of simulation in many ways using the **.PRINT**, **.PLOT**, and **.FOUR** statements.

1. The **.PRINT** statement:

General form:

.PRINT PRTYPE OV1 <OV2 ... OV8>

PRTYPE is the type of the analysis (DC, AC, TRAN ...etc.).
OVi is output variable i. Output variables can be specified in many forms:

V(N1<,N2>)

specifies the voltage difference between nodes N1 and N2. If N2 (and the preceding comma) is omitted, ground (0) is assumed. For AC analysis, V(N1<,N2>) gives the magnitude of the complex voltage. For compatibility with SPICE2, the following five additional values can be accessed for the AC analysis by replacing the "V" in V(N1,N2) with:

- V** magnitude (same as VM below)
- VR** real part
- VI** imaginary part
- VM** magnitude
- VP** phase (in radians or degrees - see the units variable description)
- VDB** $20 \log_{10}(\text{magnitude})$ i.e. in dBs

I(VXX)

specifies the current flowing in the independent voltage source named VXX. Positive current flows from the positive node, through the source, to the negative node. For the AC analysis, the corresponding replacements for the letter I may be made in the same way as described for voltage outputs i.e.

- I** magnitude (same as IM below)
- IR** real part
- II** imaginary part
- IM** magnitude

IP phase (in radians or degrees - see
the units variable description)

IDB $20 \log_{10}(\text{magnitude})$ i.e. in dBs

Examples:

.PRINT TRAN V(4) I(VIN)

.PRINT DC V(2) I(VSRC) V(23, 17)

.PRINT AC VM(4, 2) VR(7) VP(8, 3)

2. The **.PLOT** statement:

General form:

**.PLOT PLTYPE OV1 <(PLO1, PHI1)> <OV2 <(PLO2,
PHI2)> ... OV8>**

The Plot line defines the contents of one plot of from one to eight output variables. PLTYPE is the type of analysis (DC, AC, TRAN, NOISE, or DISTO) for which the specified outputs are desired. The syntax for the OV1 is identical to that for the .PRINT line.

The letter X indicates the overlap of two or more traces on any plot.

When more than one output variable appears on the same plot, the first variable specified is printed as well as plotted. If a printout of all variables is desired, then a companion .PRINT line should be included.

Examples:

.PLOT DC V(4) V(5) V(1)

.PLOT TRAN V(17, 5) (2,5) I(VIN) V(17) (1,9)

.PLOT AC VM(5) VM(31, 24) VDB(5) VP(5)

.PLOT DISTO HD2 HD3(R) SIM2

.PLOT TRAN V(5,3) V(4) (0,5) V(7) (0,10)

3. The .FOUR statement:

This is Fourier Analysis of Transient Analysis Output. The

General form:

.FOUR FREQ OV1 <OV2 OV3 ...>

Examples:

.FOUR 100K V(5)

The Four (or Fourier) line controls whether WinSpice3 performs a Fourier analysis as a part of the transient analysis. FREQ is the fundamental frequency, and OV1 the desired output vector. The Fourier analysis is performed over the interval <TSTOP-period, TSTOP>, where TSTOP is the final time specified for the transient analysis, and period is one period of the fundamental frequency. The DC component and the first nine harmonics are determined. For maximum accuracy, TMAX (see the .TRAN line) should be set to period/100.0 (or less for very high-Q circuits).

WinSpice3 INTERACTIVE INTERPRETER

WinSpice3 consists of a simulator and a front-end for data analysis and plotting. The command line interface has most of the capabilities of the UNIX C-shell. WinSpice3 can plot data from a simulation on a graphics terminal or a workstation display. Note that the raw output file is different from the data that SPICE2 writes to the standard output.

WinSpice3 commands can be included in the SPICE file enclosed between a **.control** and **.endc** lines. Commands can include loops, if-else-statements, variable assignments, analysis commands, run command, and plotting commands. This is a very powerful simulation tool that can enable the automatic design of electronic circuits. Any device parameter (e.g. W of a MOSFET or BF of a BJT) can be automatically altered (swept) to determine the required value. Check the following example:

```
.control
DC VGS 0 5 0.1           *the required analysis
destroy all             * destroy all previous results
let wn=5                 * define a new variable wn
while wn<51             * A while loop
alter @M1[w]=wn*1u      * Alter W of a transistor called M1
run                     * run WinSpice3
let wn=wn+5             * define a new value of wn
end                     * repeat the loop as long as wn <51
*then plot the results of the 10 DC runs
plot dc1.V(1) dc2.V(1) dc3.V(1) dc4.V(1) dc5.V(1) dc6.V(1)
dc7.V(1) dc8.V(1) dc9.V(1) dc10.V(1)
.endc                   * end of the WinSpice3 commands
```