

Brief Introduction to HSPICE Simulation

Wojciech Giziewicz

1 Introduction

This document is based on one written by Ihsan Djomehri, Spring 1999.

Originally developed at Berkeley in the late 60s and early 70s, SPICE has evolved into one of the tools of choice for circuit simulation. SPICE reads in a list of circuit nodes and the elements between them (called the SPICE deck since in the early days each element was entered on its own punch card, and a whole circuit simulation was thus a deck of cards) and solves for the voltages at all nodes. Relevant to our needs in 6.012, the simulation modes include nonlinear DC analysis (bias or operating point characteristics), linear AC analysis, and transient analysis.

2 The SPICE Deck

To run SPICE on Athena, make a new directory for your SPICE work, then run `add hspice` to attach the HSPICE locker. The program takes an input file (the deck) and outputs its results to the terminal. Sometimes there can be a lot of output, so it is convenient to redirect it into a file, for example
`% hspice mydeck.sp > mydeck.out.`

Use your favourite text editor to create your spice deck. Each line of the file defines an element of the circuit. The first line in the file is always treated as a comment. A `*` at the beginning of any line causes it to be treated as a comment. The general format for entering a circuit element is a one-letter identifier merged with a name, the nodes that connect to it, and the properties of the element. Numerical property values are always in SI units (meters, seconds, volts, etc...) but can be modified by a suffix letter that acts as a multiplier.

x	k	m	u	n	p	f
10 ⁶	10 ³	10 ⁻³	10 ⁻⁶	10 ⁻⁹	10 ⁻¹²	10 ⁻¹⁵

Begin by sketching out your circuit on paper, labeling each node with a number or name (only numbers were allowed in the old days), and naming each circuit element. Node 0 is always ground and floating nodes (a node from which there is no connection to ground through circuit elements) will generate an error. Your spice deck will consist of a list of all of your circuit elements (order is not important) and analysis commands. The following elements may be of interest:

Voltage:	<i>Vname N+ N- <qualifier> value</i>
Current:	<i>Iname N+ N- <qualifier> value</i>
Resistor:	<i>Rname N+ N- value</i>
Capacitor:	<i>Cname N+ N- value</i>
Inductor:	<i>Lname N+ N- value</i>
Diode:	<i>Dname N+ N- model</i>
BJT:	<i>Qname Nc Nb Ne model</i>
MOSFET:	<i>Mname Nd Ng Ns Nb model</i> <i>+ L=value W=value</i> <i>+ < AD=value AS=value</i> <i>+ PD=value PS=value></i>

Everything appearing in italics is mandatory, parameters within `<>` are optional. As shown above, a statement may be broken up into two or more lines by placing a “+” at the beginning of the second and each subsequent line.

Sources and passive devices are defined with a positive and a negative node (N+, N-). Current flows through one of these elements from the positive to negative node. This is particularly important to remember for a current source, because the specified current flows slightly counter-intuitively *through* the element from the positive to the negative node. Typically, one begins by specifying all voltage and current sources, followed by passives and active devices. For the sources, the *qualifier* input is **DC**, **AC**, or a transient function as applicable to the respective analysis. A source may be specified with all three, and the analysis will choose the appropriate parameter. At a minimum, the DC value is usually specified (e.g. **V1 in gnd DC 1**). For an AC signal, one specifies the AC magnitude (e.g. **V1 in gnd AC 1**). The

transient functions of interest are:

```
Pulse Train:      PULSE(v1 v2 td tr tf pw per)
Piecewise Linear: PWL(t1 v1 t2 v2 ...)
Sine Wave:       SIN(vo va freq <td>)
```

In the parameter lists above for the pulse train (or square wave if appropriate parameters are used), *v1* and *v2* are low and high voltages, *td* is the time delay before starting, *tr* and *tf* are the pulse rise and fall times, *pw* is the pulse width (time spent at *v2*), *per* is the period at which the pulse is repeated. For the PWL, a list of time and level pairs is provided. For the SIN function, *vo* is the offset voltage, *va* is the sinusoidal amplitude, *freq* is the frequency of the signal, and *td* is the delay. These functions may be used with current sources, specifying currents instead of voltages. Also, note that the SIN function is used with a transient analysis and does not in any way imply an AC analysis.

The active device on which we will spend the most time is the MOSFET. For this device, four connections are specified: drain, gate, source, and body. The device is referenced to an appropriate model that defines its parameters and is included as a card in the deck in the following form.

```
.model name PMOS/NMOS level=num <params>
```

The model is generally provided either in the assignment (in the case of this course) or by the manufacturer in the case of an IC foundry. A MOSFET card must also include gate length and width (parameters *L* and *W*), and drain and source areas and perimeters (*AD*, *AS*, *PD*, *PS*) for accurate transient and AC analyses.

In practice, a design will include a limited number of MOSFET geometries. At the same time, in a design with more than a handful of transistors it can be tedious (as well as an important source of errors) to manually calculate and enter the parameters every time a geometry changes. SPICE provides a facility to define variables and use expressions of these variables in other cards. The scope of the variables is the entire deck. The following is a simple example:

```
.PARAM Lnmos=1.5u
.PARAM Wnmos=3u
.PARAM Ldiff=3u
```

```
M1 out in gnd gnd Nmodel L=Lnmos W=Wnmos
+ AD='Ldiff*Wnmos' PD='2*Ldiff+Wnmos
+ AS='Ldiff*Wnmos' PS='2*Ldiff+Wnmos
```

The gate length and width, along with source/drain diffuse lengths (all in microns, hence the **u** suffix) are defined as variables, and a MOSFET as defined in the model **Nmodel** is declared (source and body connected to ground, gate to node **in**, drain to node **out**). This method may seem to lead to an unnecessary amount of typing in a trivial case, but is extremely helpful in a larger design and its use is very strongly encouraged.

3 Performing an Analysis

When using HSPICE, the deck should always include the card

```
.options post
```

Additionally, appropriate cards are required for different types of analysis. The analyses themselves are saved in separate data files. The following analyses are only a small subset of what may be performed, but they are all that will be required in this course.

3.1 DC Analysis

```
.DC Vname1/IName1 start1 stop1 step1
+ < Vname2/IName2 start2 stop2 step2>
```

This command performs a DC sweep from the specified start to end values, with the specified step. It may optionally perform a nested sweep of a second variable. The nested sweep may be useful, for example, to generate a MOSFET I_D vs. V_{DS} curve for different values of V_{GS} .

3.2 AC Analysis

```
.AC DEC pointsperinterval fstart fstop
```

This command will produce magnitude and phase information for each node of the circuit for the given

frequency range, taking data points per DECADE (logarithm base 10). Recall that in the voltage and current source cards it is possible to specify an AC qualifier followed by a magnitude. The output data will therefore be in reference to the source specified with an AC qualifier and relative to the magnitude specified. It is important to remember that in an AC analysis, spice uses only a linearised model of the circuit. Therefore, an AC input magnitude of 1000 V will give the same result as an input magnitude of 1 V multiplied by 1000. In other words, the AC analysis does not check as to whether the specified parameters violate the DC operating point conditions.

3.3 Transient Analysis

```
.TRAN tstep tstop
```

This command performs a time-dependent analysis starting from time=0 sec to time=tstop sec. recording a value every tstep seconds.

4 Viewing Analysis Results

HSPICE saves its analysis results in a set of files in the directory in which it was run. For the first assignment, we would like you to use the program **awaves** to view the analysis data. To plot the appropriate data, open your HSPICE design and choose the analysis and nodes of interest from the results browser. Using awaves you may do things such as plotting different node voltages together on the same axis and taking measurements at various points on the curves. You should take time to explore all of the options that this program provides.

5 Example

Show in Figure 1 is an example circuit, an NMOS inverter. The first step of a SPICE simulation is always to draw the circuit and name all nodes and circuit elements. Traditionally nodes were numbered, with 0 reserved for ground, but in modern SPICE implementations nodes may also have text labels.

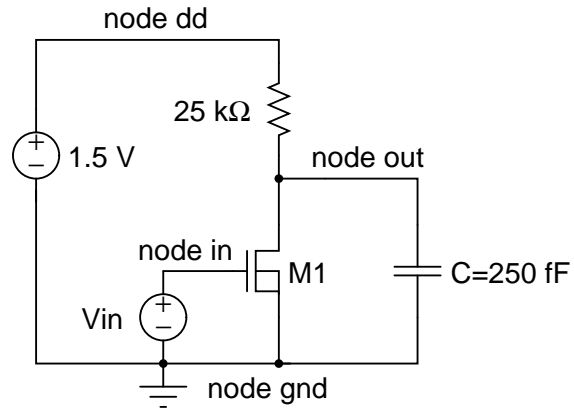


Figure 1: Schematic of an example NMOS inverter showing all circuit elements and node names.

Once the drawing is complete, a deck may be written.

NMOS Inverter Example

```
* Define Voltage Sources
Vin in gnd DC 1 PULSE(0 1.5 0n 1n 1n 18n 40n)
Vdd dd gnd DC 1.5
```

```
* Define Load Capacitor
CG out gnd 250f
```

```
* Define Load Resistor
Rload dd out 25k
```

```
* Define MOSFET parameters
.PARAM Ldiff=6u
.PARAM Wnmos=3u
.PARAM Wpmos=6u
.PARAM Lnmos=1.5u
.PARAM Lpmos=1.5u
```

```
* Define MOSFETs
```

```

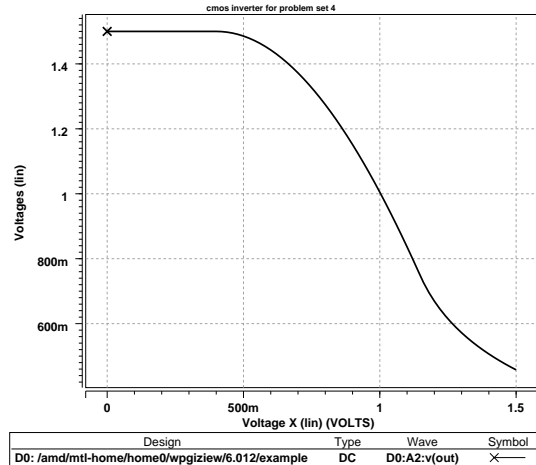
M1 out in gnd gnd N1 L=Lnmos W=Wnmos
+ AD='Ldiff*Wnmos' AS='Ldiff*Wnmos'
+ PD='2*Ldiff+Wnmos' PS='2*Ldiff+Wnmos'

* MOSFET Model Definitions
.MODEL N1 NMOS LEVEL=1 VT0=0.4 TOX=1.5e-8 KP=5e-5
+ LAMBDA=0.1 CJ=1e-4 CJSW=5e-10 PB=0.8

* Analysis
.options post
.DC Vin 0 1.5 .02
.TRAN .1n 25n

.end

```



The deck is simulated with the command

```

% add hspice
% hspice example.sp > example.out

```

The example.out file contains (as readable text) important information about the circuit, including the DC status of all of the nodes and circuit elements.

Once the simulation is complete, we can open the design in **awaves** and plot data for both of the analyses that were run. The command to open awaves is:

```

% awaves example &

```

Figure 2: Voltage at the output for Vin swept from 0 to 1.5 V

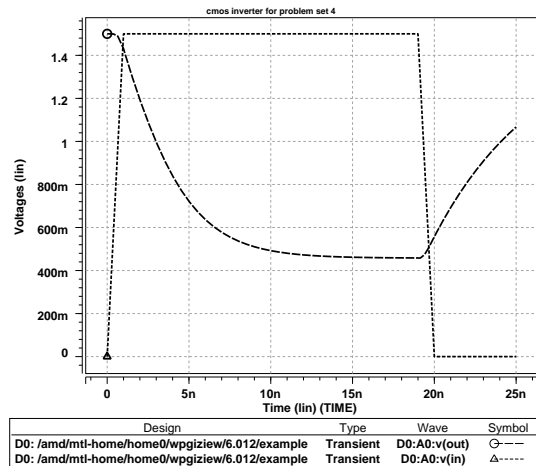


Figure 3: Transient analysis of the voltage at the output due to steps at the input.