Instituto Tecnológico y de Estudios Superiores de Occidente
Departamento de Electrónica, Sistemas e Informática

CIRCUIT SIMULATION USING WINSPICE: A QUICK INTRODUCTION

José F. Enríquez-de-la-O and José E. Rayas-Sánchez

Tlaquepaque, Jal., México
Septiembre 25, 2002

CIRCUIT SIMULATION USING WINSPICE: A QUICK INTRODUCTION

José F. Enríquez-de-la-O and José E. Rayas-Sánchez

Keywords WinSpice tutorial, WinSpice manual, NMOS model, PMOS model, analysis using WinSpice, introduction to WinSpice, circuit simulation, SPICE.

Abstract WinSpice is a general-purpose circuit simulation program for nonlinear DC, nonlinear transient, and linear AC analysis. In this paper, the creation of a specific CMOS application is shown. This tutorial is intended to help the first-time WinSpice user, by illustrating how to create the input file and how to perform different types of circuit analysis.

I. INTRODUCTION

WinSpice is a general-purpose circuit simulation program for nonlinear DC, nonlinear transient, and linear AC analysis. WinSpice is a public domain port of Spice3F4 to Win32 systems, developed at the University of California in Berkeley [1]. Circuits may contain resistors, capacitors, inductors, mutual inductors, independent voltage and current sources, four types of dependent sources, lossless and lossy transmission lines (two separate implementations), switches, uniform distributed RC lines, and the five most common semiconductor devices: diodes, BJTs, JFETs, MESFETs, and MOSFETs [2].

In this tutorial, we illustrate the simulation of a specific CMOS application. This tutorial is intended to help the first-time WinSpice user. Circuit creation (description of the input file) is illustrated. Different types of circuit analysis are also realized and illustrated.

WinSpice has built-in models for the semiconductor devices, and we need to specify only the pertinent model parameter values. The model for the BJT is based on the integral-charge model of Gummel and Poon; however, if the Gummel-Poon parameters are not specified, the

This work was supported by CONACYT (Consejo Nacional de Ciencia y Tecnología, Mexico) under Grants 010581 and I39341-A.

J.E. Rayas-Sánchez is with the Department of Electronics, Systems and Informatics, ITESO University, Tlaquepaque, Jalisco, 45090 Mexico (http://iteso.mx/~erayas). J.F. Enríquez-de-la-O is supported by a Research Assistanship granted by ITESO.

model reduces to the simpler Ebers-Moll model. In either case, charge-storage effects, ohmic resistances, and a current-dependent output conductance may be included. The diode model can be used for either junction diodes or Schottky barrier diodes. The JFET model is based on the FET model of Shichmanand Hodges. Six MOSFET models are implemented: MOS1 is described by a square-law I-V characteristic, MOS2 is an analytical model, while MOS3 is a semi-empirical model; MOS6 is a simple analytic model accurate in the short-channel region (this level is used in the example); MOS4 and MOS5 are the BSIM (Berkeley Short-channel IGFET T Model) and BSIM2. MOS2, MOS3, and MOS4 include second-order effects such as channel-length modulation, subthreshold conduction, scattering-limited velocity saturation, small-size effects, and charge-controlled capacitances [3].

II. CREATING CIRCUITS IN WINSPICE

A. Drawing the Circuit

The circuit to be simulated using WinSpice is shown in Fig. 1. First, WinSpice needs to know all about the circuit. Nodes in the circuits must be numbered, starting with 0 for the ground. Each node should have a unique number and no number should be skipped. Fig. 2 shows one way the nodes could be numbered. Instead of numbers, you can also assign text labels to the nodes (excepting ground, which is always node 0). Using text makes the circuit more readable.

The next step is to remove any loose end from the circuit. For instance, notice that Vdd, Vss, Vs (input voltage), and the output voltage (nodes 7, 1, 8 and 4 respectively) have no path for current to flow through them to ground. This ambiguity can be removed by inserting voltage sources for Vdd, Vss and the input. If the output is to be simulated as unloaded then no path to ground is needed. The output node becomes a point of reference for voltage only. Fig. 3 shows the circuit with the ambiguities removed for the bias sources.

B. Defining the Circuit Components

Now we are ready to convert each device and supply in the circuit into a descriptive command for WinSpice. The entire description of the circuit is shown in the appendix. The circuit file must be created with a text editor and must be saved with .cir extension. WinSpice does not have an schematic capture tool, as Workbench or OrCad, we have to work with text files in a command line environment, which makes it a little harder but, on the other hand, there is more control on the circuit.

The circuit to be analyzed is described to WinSpice by a set of element lines, which define the circuit topology and element values, and a set of control lines, which define the model parameters and the simulation controls. The first line in the input file must be the title, and the last line must be ".END".

An element line contains the element name, the circuit nodes to which the element is connected, and the values of the parameters that determine the electrical characteristics of the element. One line should be defined for each element in the circuit. The first letter of the element name specifies the element type. The format for the WinSpice element types is as follows. The strings XXXXX, YYYYY, and ZZZZZ denote arbitrary alphanumeric strings. For example, a resistor name must begin with the letter R and can contain one or more characters. Hence, R, R1, RSE, ROUT, and R3AC are valid resistor names [4].

Fields on a line are separated by one or more blanks, a comma, an equal ('=') sign, or a left or right parenthesis; extra spaces are ignored. A line may be continued by entering a '+' (plus) in column 1 of the following line; WinSpice continues reading beginning with column 2. A name field must begin with a letter (A through Z) and cannot contain any delimiters. A number field X may be an integer field (12, –44), a floating point field (3.14159), either an

integer or floating point number followed by an integer exponent (1e−14, 2.65e3), or either an integer or a floating point number followed by typical scaling factors.

Letters immediately following a number that are not scaling factors are ignored, and letters immediately following a scaling factor are ignored. Hence, 10, 10V, 10Volts, and 10Hz all represent the same number, and M, MA, MSec, and MMhos all represent the same scale factor. Note that 1000, 1000.0, 1000Hz, 1e3, 1.0e3, 1KHz, and 1K all represent the same number. Node names may be arbitrary character strings. The symbols supported for the scaling factors are shown in Fig. 4.

Each node in the circuit must have a DC path to ground. Every node must have at least two connections except for transmission line nodes (to permit unterminated transmission lines), which are not used in this tutorial, and MOSFET substrate nodes (which have two internal connections anyway).

III. DETAILED DESCRIPTION OF A SPICE NETLIST

In this section we describe the circuit file in the appendix. The first line of WinSpice source code is the title line and will be ignored by the compiler. We may leave it blank, but it is preferable to use it to identify the circuit. Its contents is printed verbatim as the heading for each section of output.

An asterisk in the first column indicates that that line is a comment line. Comment lines may be placed anywhere in the circuit description. Note that WinSpice also considers any line with leading white space to be a comment.

Line 22 is the last circuit line. This is the end line. Note that the period is an integral part of the command.

There are 4 independent sources in the circuit. General forms for elements like

independent sources, resistors, capacitors, transistors and others, can be referred in SPICEMAN tutorial [4].

Lines 2 to 5 have the voltage and current sources definitions. It is important to be consistent with the Fig. 2 node's labels. The same node should never be named with different numbers or labels. Line 2 creates a Vdd source from node 7 to node 0 with 5 volts DC voltage and 0 volts AC voltage. Line 3 creates a Vss source from node 1 to node 0 with a negative 5 volts DC voltage and 0 volts AC voltage. Finally, line 4 creates a Vin source from node 8 to node 0 with 0 volts DC voltage and 1 volt AC voltage. This source is going to be used in the frequency domain analysis. Line 5 defines a 100μA constant current source from node 9 to node 2.

The circuit elements of the example are resistors and transistors. Transistors are simulated using a device model.

In the case of resistors, those are created directly, just indicating the nodes and some parameters. Lines 6 and 7 define two different resistors. The first one is between nodes 5 and 4, with 20 KΩ value. The second one is created between the nodes 5 and 0 with a 1 KΩ value.

A. Semiconductor Models

Most simple circuit elements, like resistors, typically require only a few parameter values. However, some devices (semiconductor devices in particular) that are included in WinSpice require many parameter values. Often, many devices in a circuit are defined by the same set of device model parameters. For these reasons, a set of device model parameters is defined on a separate .MODEL line and assigned a unique model name. The device element lines in WinSpice make reference to the model name. The most common semiconductor device models and parameters can be found in any classical textbook, as in [5].

For these more complex device types, each device element line contains the device name, the nodes to which the device is connected, and the device model name. In addition, other optional parameters may be specified for some devices: geometric factors and an initial condition.

Parameter values are defined by appending the parameter name followed by an equal sign and the parameter value. Model parameters that are not given a value are assigned the default values for each model type.

Transistors in lines 8 to 15 are simulated using a device model. For example, M_4 and M_5 are the names given to the transistors shown in the Fig. 1. The node numbers for the drain, gate, source and substrate of the transistor must be given in this specific order. Thus node 9 refers to the drain, node 9 is the gate and node 7 connects the source and the substrate to the supply voltage Vdd in the M_4 transistor. MODP is the name for the model used by all the p-channel transistors in the circuit. Transistor M_5 (and all the n-channel transistors) uses a model named MODN.

In lines 16 and 17, the definitions of the MODN and MODP models are specified. NMOS indicates a circuit element, in this case, an n-channel MOSFET. Level 2, defines a kind of model for MOSFETs, and VTO, KP and LAMBDA (λ) are some of the transistor's parameters. It should be noticed that model parameters that are not given a value are assigned the default values [3].

B. Analysis and Output Control

Lines 18 and 19 specify the analyses, while lines 20 and 21 indicate the plots requested. DC analysis is made in line 18. It causes the value of the voltage source Vin to be swept from 0 volts to 0.6 volts in increments of 0.001 volts. In line 19 an AC analysis is performed to the

example circuit. In this line, DEC stands for decade variation, 10 is the number of points per decade and the sweep starts at 1 KHz and stops at 15 MHz. Lines 20 and 21 make WinSpice to plot the DC and AC analysis.

IV. SIMULATION RESULTS

The results from the DC simulation are shown in Fig. 5 and Fig. 7, where V(4), which corresponds to the output voltage, is plotted versus Vin. It is noticed in Fig. 5 that the slope in the linear region of the curve (from Vin = 0 mV to Vin = 180 mV approximately), represents the feedback amplifier gain when the channel length modulation effect (or λ-effect) is considered. Fig. 7 shows the same analysis, but without the λ-effect. It is observed that the amplifier gain without the λ-effect rises to 21, being 15.9 approximately when this effect is considered.

The results from the AC simulation are shown in Fig. 6 and Fig. 8. We can calculate the bandwidth for this amplifier from those figures. The bandwidth considering the λ-effect (see Fig. 6) is approximately of 6.245 GHz and without the λ-effect (see Fig. 8) the bandwidth decreases to 5.167 GHz approximately. The gain shown in Fig. 6 is close to 16 because of the λ-effect. Without the λ-effect, the amplifier gain is about 21 (see Fig. 8). These plots confirm that the λ-effect reduces the gain but increases the bandwidth.

It is important to notice that the bandwidth of the amplifier depends not only on the ratio of the width (W) to the length (L) of the MOSFETs channels, but also depends on the values of W and L. It can be seen in Fig. 9 (with $\lambda \neq 0$) and Fig. 10 (with $\lambda = 0$) that the bandwidth of the amplifier decreases considerably (two orders of magnitude approximately), when the values of W and L in the MOSFETs increases by a factor of 10, in spite of keeping W/L constant. This makes sense, since the larger the dimensions of the MOSFETs the larger the parasitic capacitances associated.

V. COMPARISON WITH ANALYTICAL RESULTS

The first step for the analysis is to calculate the DC bias point for each transistor. The internal parameters of the MOSFET transistors are: $\mu_n C_{ox} = 60\mu A/V^2$, $\mu_p C_{ox} = 30\mu A/V^2$, $V_{tn} = 0.8V$, $V_{tp} = -0.8V$, $\lambda = 0.03V^{-1}$. Then

$$K_n = \frac{1}{2}\left(\frac{W}{L}\right)(\mu_n C_{ox}) = (0.5)(20)(60\mu A/V^2) = 0.6mA/V^2 \tag{1}$$

$$K_n = K_5 = \frac{1}{2}K_1 = \frac{1}{2}K_2 = \frac{1}{2}K_6 = \frac{1}{2}K_7 = \frac{1}{2}K_8 \tag{2}$$

$$K_p = K_3 = K_4 = \frac{1}{2}\left(\frac{W}{L}\right)(\mu_p C_{ox}) = (0.5)(20)(30\mu A/V^2) = 0.3mA/V^2 \tag{3}$$

Since M_1 and M_2 are identical, and knowing that $I_{G8} = 0A$, then

$$I_{DS5} = I_1 = I_{DS4} = I_{DS3} = I_{DS2} = I_{DS1} = 100\mu A \tag{4}$$

We know that

$$V_{GS1} = V_{GS2} = \sqrt{\frac{I_{DS1}}{2K_n}} + V_{tn} = 1.088V \tag{5}$$

$$V_{GS3} = V_{GS4} = \sqrt{\frac{I_{DS5}}{K_p}} + V_{tp} = 1.377V \tag{6}$$

and

$$V_{GS6} = V_{GS5} = V_{GS7} = \sqrt{\frac{I_{DS5}}{K_n}} + V_{tn} = 1.208V \tag{7}$$

Since $V_{GS1} = V_{GS2}$, then the bias current through R_1 is zero and

$$I_{DS6} = I_{DS7} = I_{DS8} = 2K_n(V_{GS6} - V_{tn}) = 200\mu A \tag{8}$$

then

$$V_{GS8} = \sqrt{\frac{I_{DS8}}{2K_n}} + V_{tn} = 1.208\text{V} \tag{9}$$

It should be noticed that the Early effect was neglected in the previous DC analysis.

Now we perform the AC analysis. The total open loop voltage gain is the product of the differential stage gain times the common drain stage gain. The gain of the differential stage is approximated using

$$A_{dif} = \frac{V_{od}}{V_s} = g_{m2}(r_{oM2} \,||\, r_{oM3} \,||\, r_{inM8}) \tag{10}$$

Since the input impedance of a MOSFET is extremely large, (10) can be simplified as

$$A_{dif} = g_{m2}(r_{oM2} \,||\, r_{oM3}) \tag{11}$$

Since $I_{DS2} = I_{DS3}$ then

$$r_{oM2} = r_{oM3} = \frac{1}{\lambda I_{DS2}} = 333.5\text{K}\Omega \tag{12}$$

Evaluating the transconductance for M_2,

$$g_{m2} = 2k_2(V_{GS2} - V_{tn}) = 6.912 \times 10^{-4} \tag{13}$$

subsequently

$$A_{dif} = g_{m2}(r_{oM2} \,||\, r_{oM3}) = (6.912 \times 10^{-4})(1.67 \times 10^{5}) = 115.2 \tag{14}$$

The common drain stage gain is calculated from

$$A_{drain} = \frac{g_{m8}(r_L \,||\, r_{oM8} \,||\, r_{oM7})}{1 + g_{m8}(r_L \,||\, r_{oM8} \,||\, r_{oM7})} = 0.942 \tag{15}$$

where

$$r_L = R_1 + R_2 = 21\text{K}\Omega \tag{16}$$

$$r_L \,||\, r_{oM8} \,||\, r_{oM7} = r_L \,||\, \frac{1}{\lambda I_{DS8}} \,||\, \frac{1}{\lambda I_{DS7}} = 16.78\text{K}\Omega \tag{17}$$

The total open loop voltage gain can then be calculated using (14) and (15) as

$$A = A_{dif} A_{drain} = (115.2)(0.942) = 108.5 \tag{18}$$

Knowing the open loop voltage gain, the amplifier feedback voltage gain is calculated using

$$A_f = \frac{A}{1 + A\beta} = \frac{108.5}{1 + (108.5)\left(\frac{1}{21}\right)} = 17.59 \tag{19}$$

Comparing the gain result in Fig. 6 with the analytical result previously obtained, we can realize that both are similar, 15.9 for the WinSpice result and 17.59 for the previous analysis.

When the Early effect is not considered, the gain is about the 21 (see Fig. 8). This result corresponds to the feedback network gain (A_{fn}), which is calculated using

$$A_{fn} = \frac{1}{\beta} = \frac{R_2 + R_1}{R_2} = 1 + \frac{R_1}{R_2} = 1 + 20 = 21 \tag{20}$$

The voltage gain in (20) corresponds to the gain in Fig. 8, where the λ-effect is not considered.

VI. CONCLUSIONS

We described a simple example on how to use WinSpice for circuit simulation. This quick tutorial is intended to help electrical engineering students to use WinSpice for simulating typical analog electronic configurations. When using simulation tools we can vary many parameters, and request the results by simply modifying just a few code lines.

For the simulated CMOS feedback amplifier, we verified that the channel length modulation effect (λ-effect) clearly decreases the gain of the amplifier; with WinSpice we show that by changing just a few parameters in the original circuit description.

WinSpice can be a great tool for analog design. It can also be very useful for teaching

and learning electronics. Using WinSpice we are able to analyze quickly and trustily, electrical and electronic circuits. Even more, we have more control on the circuit with its command line environment. This last feature will be exploited in future work.

REFERENCES

[1] http://www.winspice.com/

[2] VLSI CAD SPICE3 Support, http://bear.ces.cwru.edu/eecs_cad/cad_spice.html

[3] SPICE Manual, http://www.reg.trlabs.ca/vlsi/simulation/winspice/winspice.html

[4] T. Quarles, A. R. Newton, D. O. Pederson and A. Sangiovanni, *WinSPice3´s User´s Manual*. Berkeley, CA: Department of Electrical Engineering and Computer Sciences, University of California, March 2001.

[5] A.S. Sedra and K.C. Smith, *Microelectronics circuits*. Mexico, DF: Oxford, 1999, pp. 480-512.

FIGURES

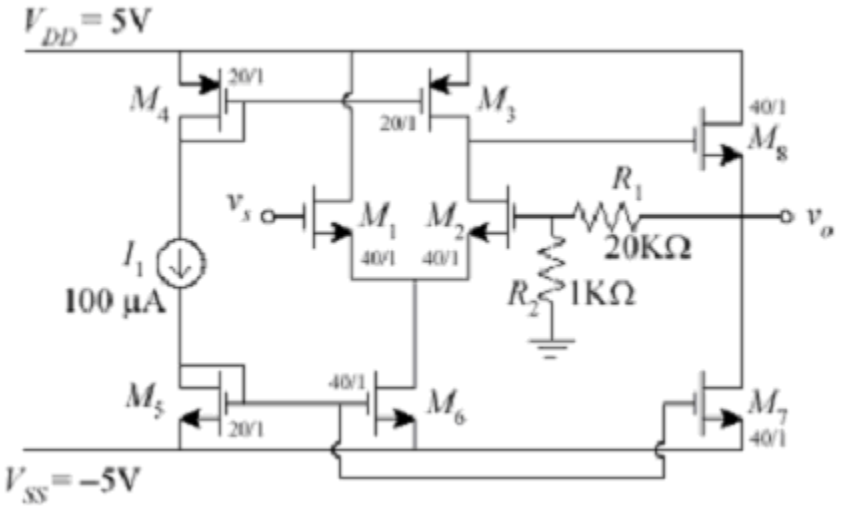

Fig. 1. Schematic of the CMOS feedback amplifier described in the Appendix.

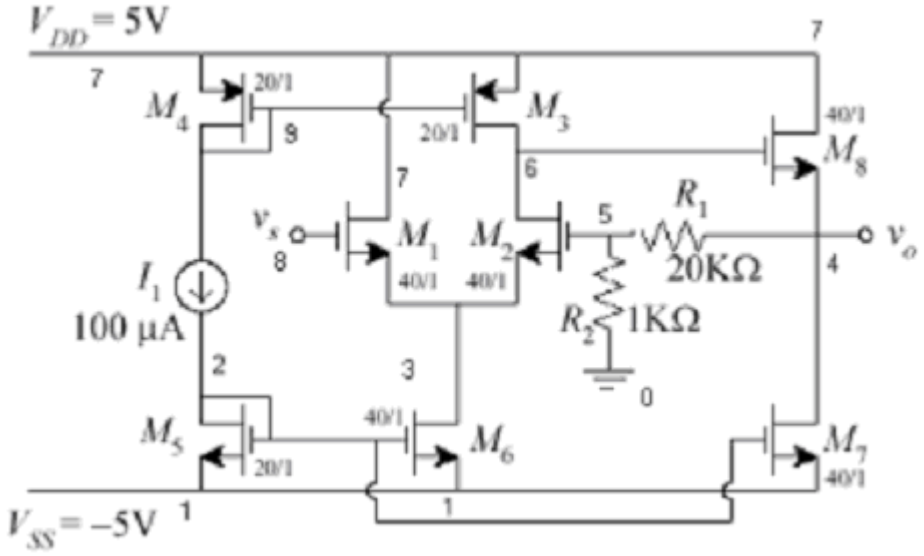

Fig. 2. Numbering the nodes of the CMOS feedback amplifier.

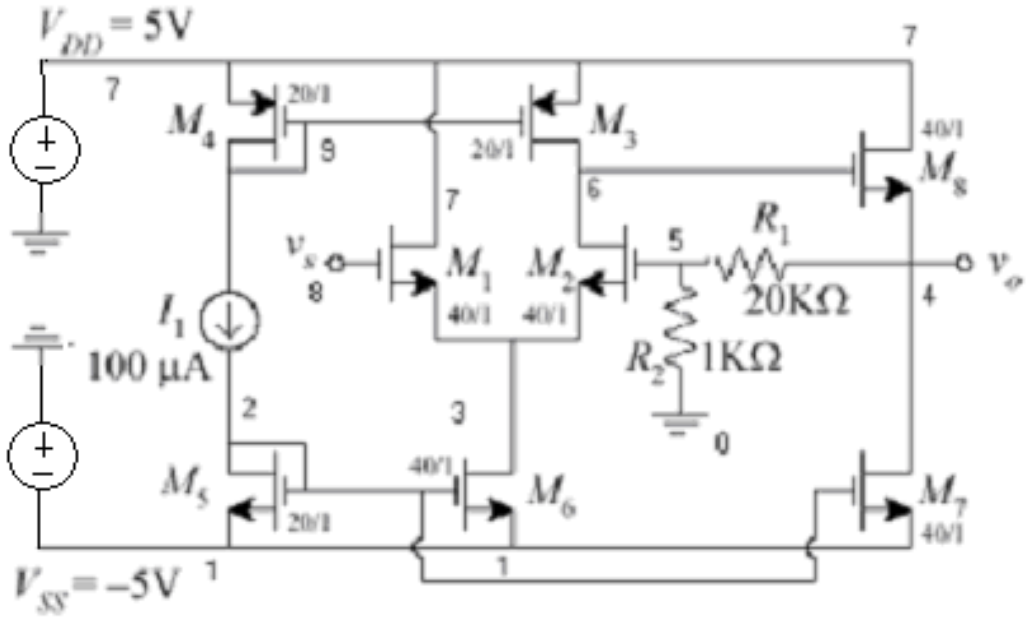

Fig. 3. Indicating the voltage power supplies connections (implicit in Fig. 2).

T = 10^{12}	**G = 10^{9}**	**Meg = 10^{6}**	**K = 10^{3}**	**mil = 25.4^{-6}**
m = 10^{-3}	**u = 10^{-6}**	**N = 10^{-9}**	**p = 10^{-12}**	**f = 10^{-15}**

Fig. 4. Symbols supported for scaling factors in WinSpice (notice that WinSpice is case-insensitive).

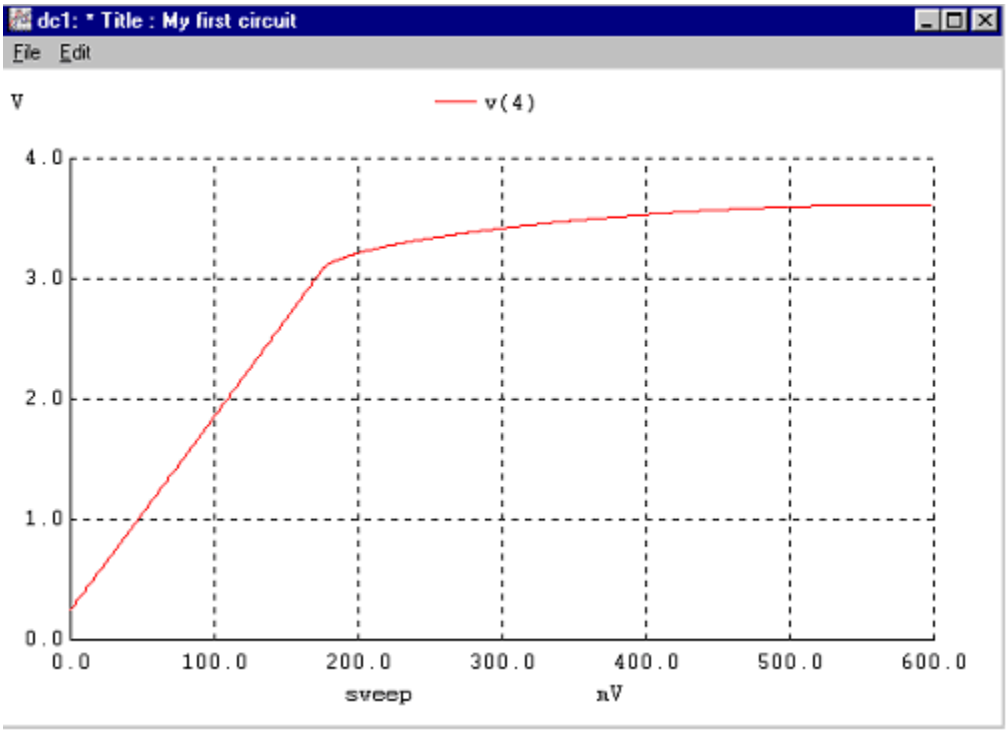

Fig. 5. DC simulation of the CMOS feedback amplifier (with $\lambda \neq 0$).

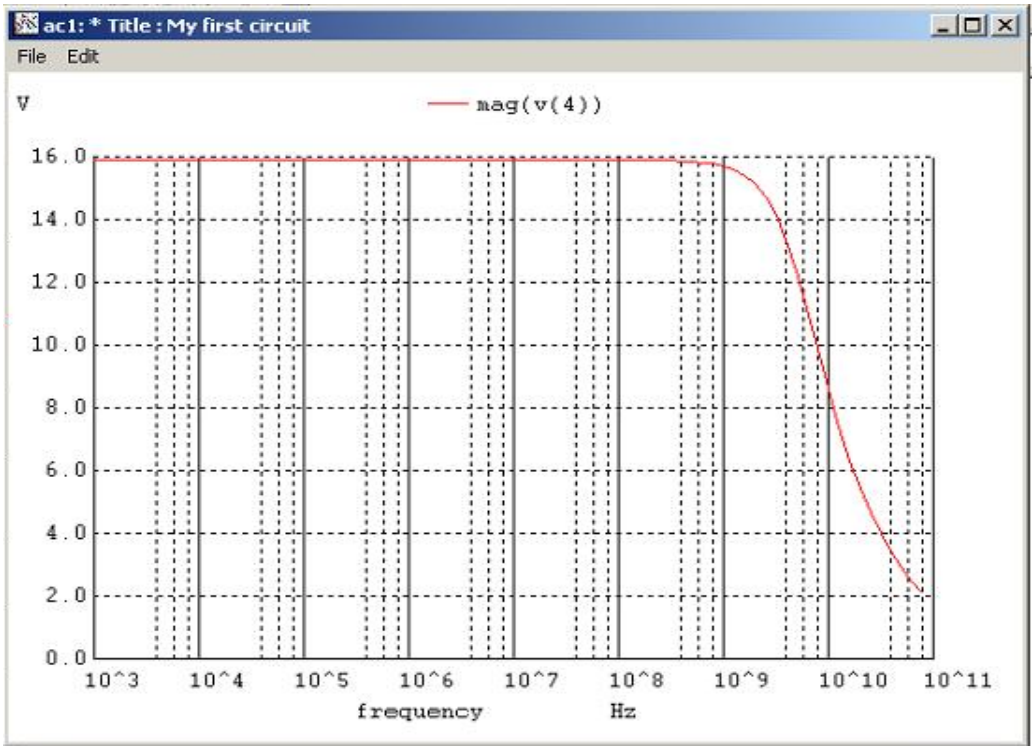

Fig. 6. AC simulation of the CMOS feedback amplifier (with $\lambda \neq 0$).

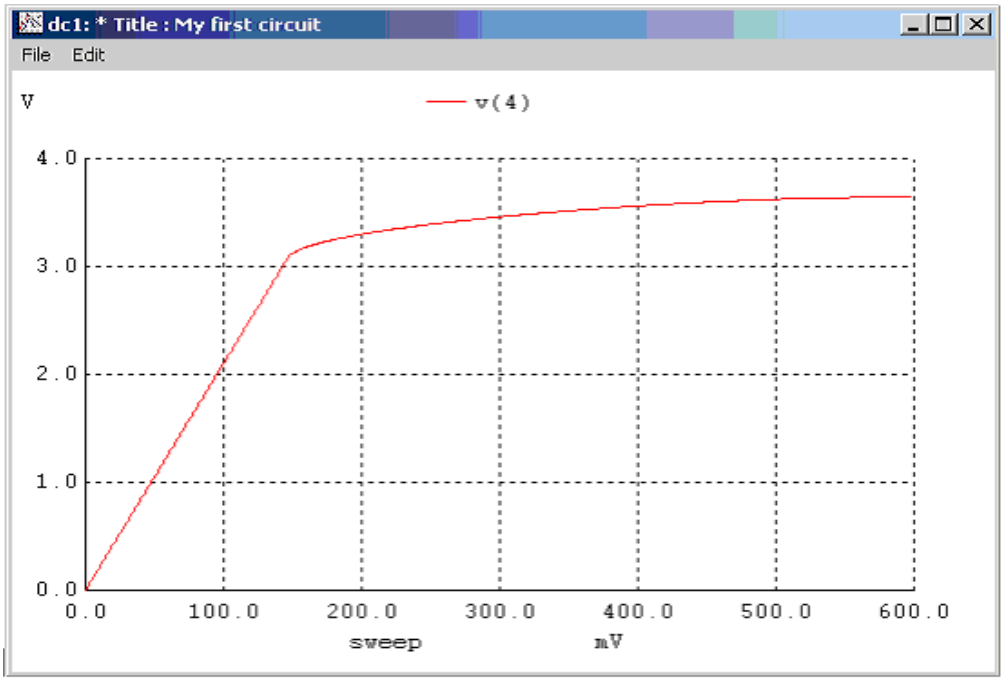

Fig. 7. DC simulation of the CMOS feedback amplifier (with $\lambda = 0$).

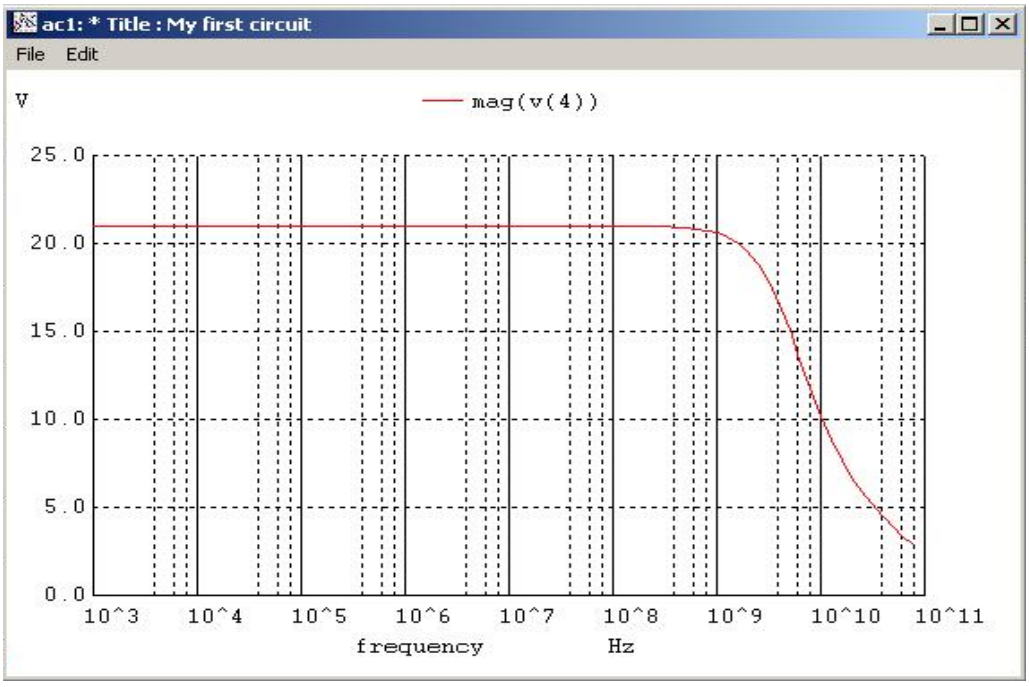

Fig. 8. AC simulation of the CMOS feedback amplifier (with $\lambda = 0$).

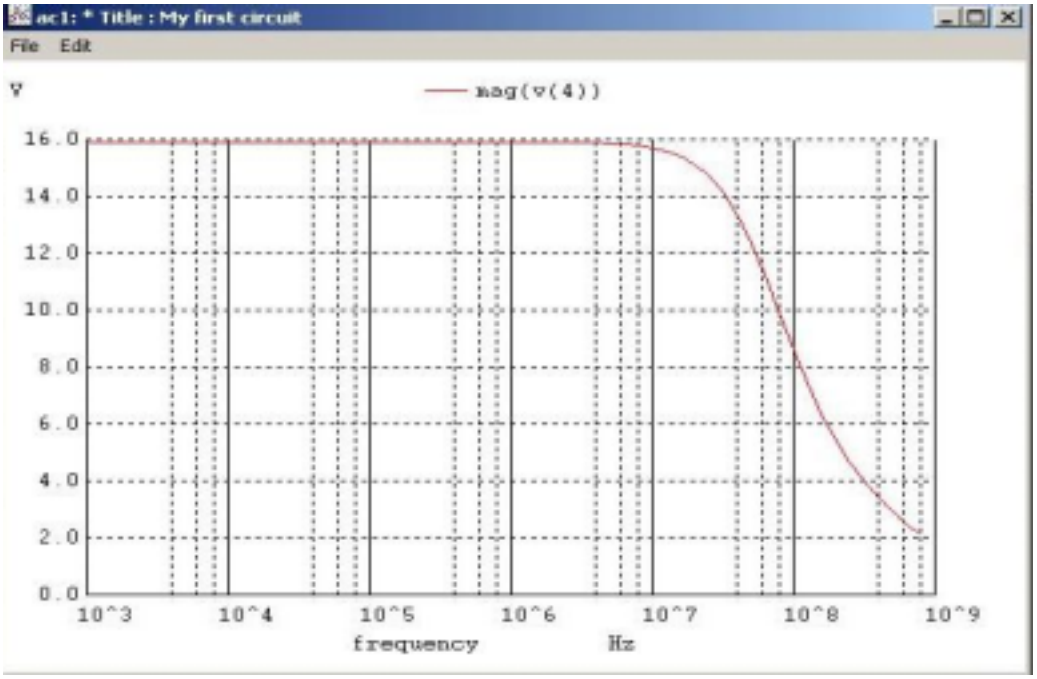

Fig 9. AC simulation of the CMOS feedback amplifier (with $\lambda \neq 0$) increasing the values of *W* and *L* by a factor of 10.

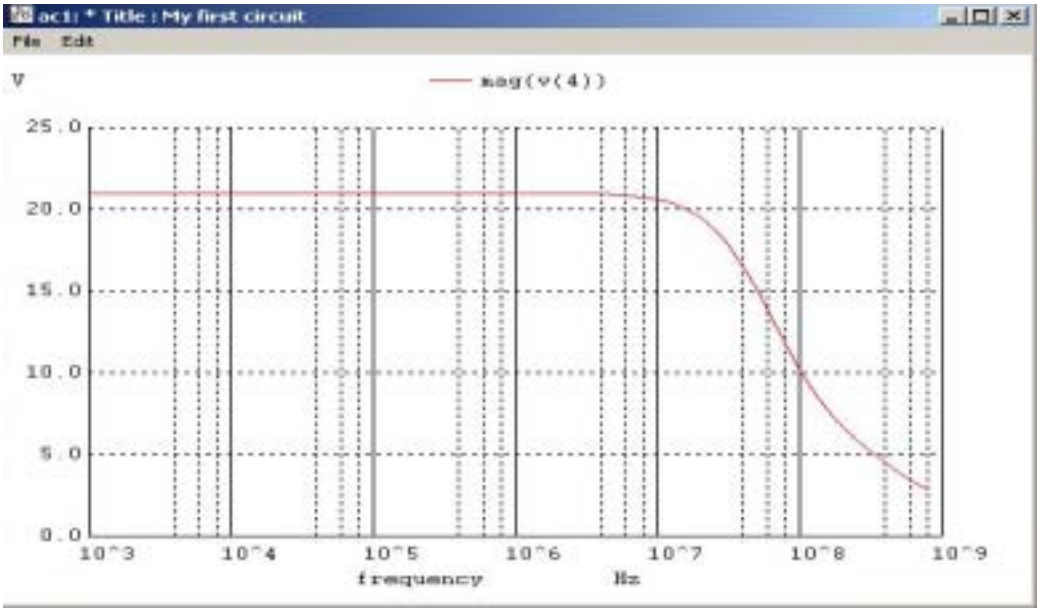

Fig 10. AC simulation of the CMOS feedback amplifier (with $\lambda = 0$) increasing the values of *W* and *L* by a factor of 10.

APPENDIX

WinSpice Netlist of a CMOS Feedback Amplifier

```
 1 : A CMOS Feedback Amplifier
 2 : vdd 7 0 dc 5v ac 0
 3 : vss 1 0 dc -5v ac 0
 4 : vin 8 0 dc 0 ac 1
 5 : iss 9 2 dc 100u
 6 : r1 5 4 20k
 7 : r2 5 0 1k
 8 : m1 7 8 3 1 modn  l=1u w=40u
 9 : m2 6 5 3 1 modn  l=1u w=40u
10 : m3 6 9 7 7 modp  l=1u w=20u
11 : m4 9 9 7 7 modp  l=1u w=20u
12 : m5 2 2 1 1 modn  l=1u w=20u
13 : m6 3 2 1 1 modn  l=1u w=40u
14 : m7 4 2 1 1 modn  l=1u w=40u
15 : m8 7 6 4 1 modn  l=1u w=40u
16 : .model modn nmos level=2 vto=+.8v kp=0.00006 lambda=0
17 : .model modp pmos level=2 vto=-.8v kp=0.00003 lambda=0
18 : .dc vin 0v .6v 1mv
19 : .ac dec 10 1000 90000meg
20 : .plot dc v(4)
21 : .plot ac v(4)
22 : .end
```