A

Introduction to SPICE

The circuits encountered in microelectronics may contain a few devices or a few million devices.[1] How do we analyze and design these circuits? As the number of devices in a circuit increases, hand analysis becomes more difficult, eventually reaching a point where other methods are required. For example, one can *build* a prototype using discrete components and observe its behavior. However, discrete devices provide a poor approximation of modern integrated circuits. Furthermore, even for a few hundred devices, discrete prototypes become prohibitively complex.

Today's microelectronics employs simulation programs extensively. A versatile tool used to predict the behavior of circuits is Simulation Program with Integrated Circuit Emphasis (SPICE). While orginally developed as a public-domain tool (at University of California, Berkeley), SPICE has evolved into commercial tools such as PSPICE, HSPICE, etc., most of which retain the same format. This appendix provides a tutorial overview of SPICE, enabling the reader to perform basic simulations. More details can be found in [1].

A.1 Simulation Procedure

Suppose we have the circuit shown in Fig. A.1(a) and wish to use SPICE to study its frequency response. That is, we wish to verify that the response is relatively flat for $f < 1/(2\pi R_1 C_1) \approx 15.9$ MHz and begins to roll off thereafter [Fig. A.1(b)]. To this end, we apply a sinusoidal voltage to the input and vary its frequency from, say, 1 MHz to 50 MHz.

The procedure consists of two steps: (1) define the circuit in a language (format) that SPICE understands, and (2) use an appropriate command to tell SPICE to determine the frequency response. Let us begin with the first step. This step itself consists of three tasks.

(1) Label each node in the circuit. Figure A.1(c) depicts an example, where the labels "in" and "out" refer to the input and output nodes, respectively. The common (ground) node *must* be called "0" in SPICE. While arbitrary, the labels chosen for other nodes should carry some information about their respective nodes so as to facilitate reading the SPICE description of the circuit.

(2) Label each element in the circuit. Defining the type of the element (resistors, capacitors, etc.), each of these labels must begin with a *specific letter* so that SPICE recognizes the element. For example, resistor labels must begin with r, capacitor labels with c, inductor labels with l, diode labels with d, and voltage sources with v.[2] Our simple circuit now appears as shown in Fig. A.1(d).

[1] Recent microprocessors contain one billion MOS transistors.

[2] SPICE does not distinguish between lower-case and upper-case letters.

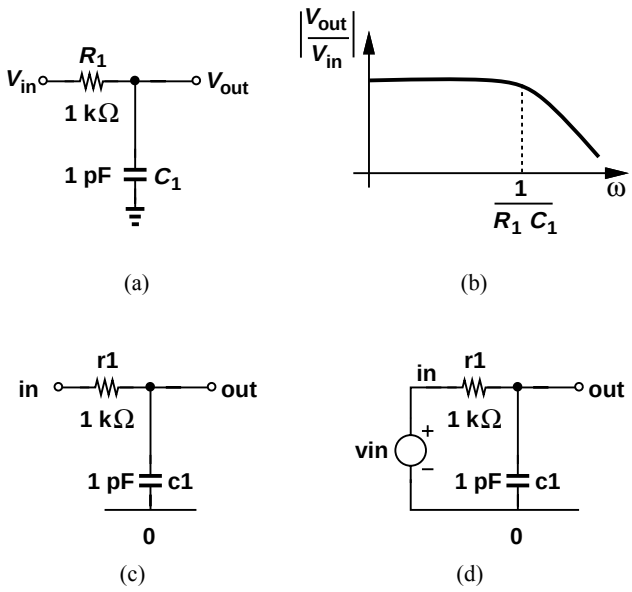


Figure A.1 (a) Simple RC circuit, (b) its frequency response, (c) with nodes labeled, (d) with elements labeled.

(3) Construct the "netlist," i.e., a precise description of each element along with the nodes to which it is tied. The netlist consists of text lines, each describing one element, with the following format for two-terminal devices:

```
elementlabel node1 node2 value
```

From the example in Fig. A.1(d), we begin the netlist with:

```
r1 in out 1k
c1 out 0 10p
```

Note that the units are specifie as a single letter (k for 10^3, p for 10^{-12}, etc.). For the input voltage source, we write

```
vin in 0 ac 1
```

where ac denotes our desire to determine the frequency (ac) response and hence designates V_{in} as a sinusoidal voltage source whose frequency will be varied. The value 1 at the end represents the peak amplitude of the sinusoid. Also note that the first node, "in," is assumed to be the positive terminal of the voltage source.

The netlist must also include the "type of analysis" that we wish SPICE to perform. In our example, SPICE must vary the frequency from one value to another, e.g., 1 MHz to 50 MHz. The corresponding command appears as

```
.ac dec 200 1meg 50meg
```

Note that each "command" line begins with a period. The first entry, "ac," requests SPICE to perform an "ac analysis," i.e., determine the frequency response. The second and third entries, "dec 200," tell SPICE to simulate the circuit at 200 frequency values in every decade of frequency (e.g., from 1 MHz to 10 MHz). The last two entries, "1meg 50meg," set the lower and upper

values of the frequency range, respectively. Note that "meg" denotes 10^6 and should not be confused with "m," which stands for 10^3.

We need two more lines to complete our netlist. The first line of the file is called the "title" and carries no information for SPICE. For example, the title line may read "My Amplifier." Note that SPICE always ignores the first line of the file, encountering errors if you forget include the title. The last line of the file must be a ".end" command. Our netlist now appears as:

```
Test Circuit for Frequency Response
r1 in out 1k
c1 out 0 10p
vin in 0 ac 1
.ac dec 200 1meg 50 meg
.end
```

Note that, except for the first and last lines, the order of other lines in the netlist is unimportant.

What do we *do* with the above netlist? We must "run" SPICE on this file, which we call, for example, test.sp. Depending on the operating system, running SPICE may entail clicking on an icon in a graphics interface or simply typing:

```
spice test.sp
```

After the simulation is successfully run, various node voltages can be plotted using the graphics interface that accompanies SPICE.

Figure A.2 summarizes the SPICE simulation procedure. The definition of (voltage or current) sources in the netlist must be consistent with the type of analysis. In the above example, the input

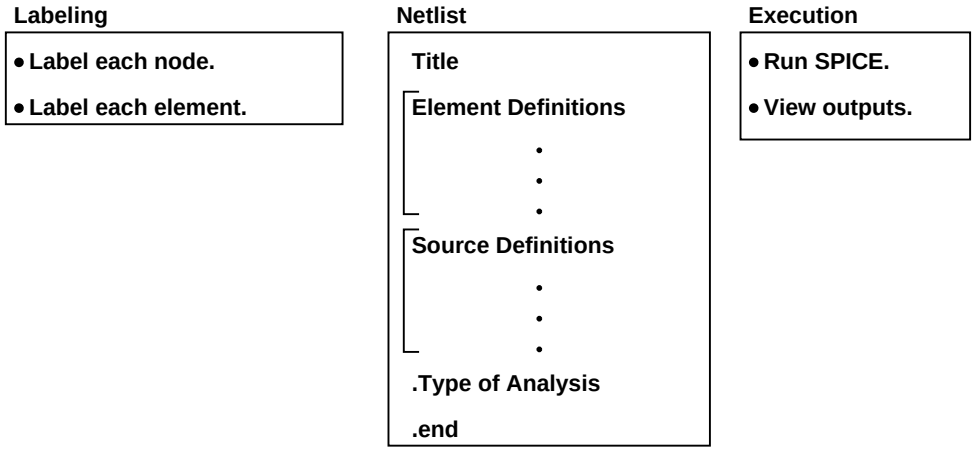


Figure A.2 Simulation procedure.

voltage source definition contains the entry "ac" so that SPICE applies the frequency sweep to V_{in} rather than other sources.

At this point, the reader may raise many questions: How are other elements defined in the netlist? How are the units specified? Is the order of the node labels in the netlist important? How are other types of analysis specified? We answer these questions in the following sections.

A.2 Types of Analysis

In addition to frequency response, other aspects of circuits may also be of interest. This section provides the (voltage or current) source descriptions and commands necessary to perform other

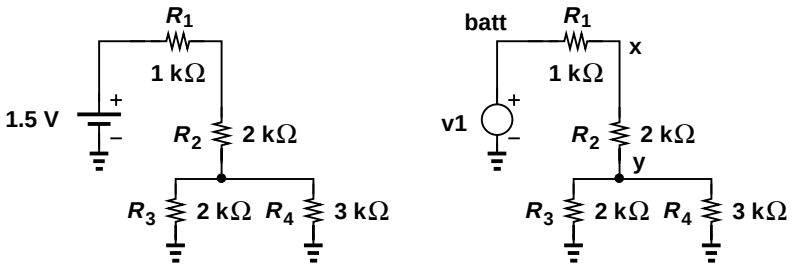
types of analysis.

A.2.1 Operating Point Analysis

In many electronic circuits, we must first determine the bias conditions of the devices. SPICE performs such an analysis with the .op command. The following example illustrates the procedure.

Example A.1

Determine the currents flowing through R_3 and R_4 in Fig. A.3(a).



(a) (b)

Figure A.3

Solution

We label the nodes as shown in Fig. A.3(b) and construct the netlist as follows:

```
Simple Resistive Network
v1 batt 0 1.5
r1 batt x 1k
r2 x y 2k
r3 y 0 2k
r4 y 0 3k
.op
.end
```

SPICE predicts a current of 0.214 mA through R_3 and 0.143 mA through R_4.

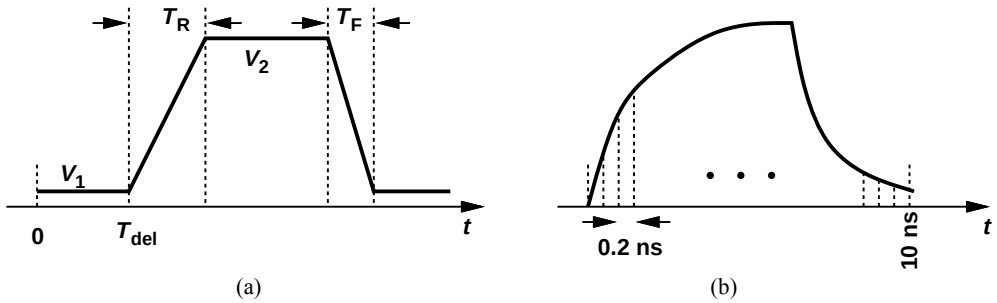
A.2.2 Transient Analysis

Suppose we wish to study the pulse response of the RC section shown in Fig. A.1(d). Called "transient analysis," this type of simulation requires changing the vin and .ac lines while maintaining the same netlist descriptions for R_1 and C_1. The voltage source must now be specified as

```
                V1 V2 Tdel Tr Tf Tw
vin in 0 pulse(0  1  0    1n 2n 5n)
```

where V_1, ..., T_w are defined as depicted in Fig. A.4(a).[3] We say V_{in} is a pulse that goes from 0

[3] The parentheses following the pulse description are for clarity and not essential.



(a) (b)

Figure A.4 (a) Def nition of pulse parameters, (b) illustration of time step.

V to 1 V with zero delay (T_{del}), a rising transition of 1 ns (T_R), a falling transistion of 2 ns (T_F), and a width of 5 ns (T_w). Note the f rst node, "in," is assumed to be the positive terminal of the voltage source.

How do we tell SPICE to perform a transient analysis? The command is as follows:

```
.tran 0.2n 10n
```

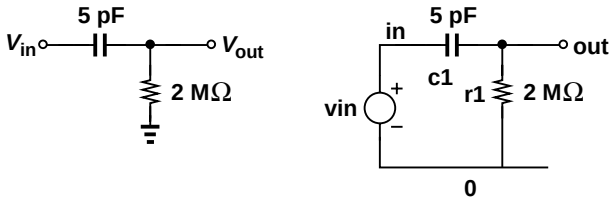
where 0.2n indicates the increments ("time steps") that SPICE must use in calculating the response, and 10n the total time of interest [Fig. A.4(b)].

The overall netlist now appears as:

```
Pulse Response Example
r1 in out 1k
c1 out 0 10p
vin in 0 pulse(0 1 0 1n 2n 5n)
.tran 0.2n 10n
.end
```

Example A.2

Construct a SPICE netlist for the pulse response of the circuit shown in Fig. A.5(a).



(a) (b)

Figure A.5

Solution

We begin with labeling the nodes and the elements [Fig. A.5(b)]. Given the time constant $R_1C_1 = 10$ μs, we postulate that the rising and falling transitions of the input pulse can be as long as approximately 1 μs and still appear "abrupt" to the circuit. For the pulsewidth, we choose 30 μs to allow the output to "settle." We therefore have

```
High-Pass Filter Pulse Response
c1 in out 5p
r1 out 0 2meg
```

```
vin in 0 pulse(0 1 1u 1u 30u)
.tran 0.2u 60u
.end
```

(The letter u in the pulse description denotes 10^{-6}.) Note that the timestep is chosen sufficiently smaller than the pulse transition times, and the overall transient time long enough to reveal the response after the input falls to zero.

Example A.3

Revise the SPICE netlist constructed in Example A.2 so as to observe the *step* response of the circuit.

Solution

We wish V_{in} to jump to 1 V and remain at this level. The pulse description, however, requires a pulsewidth value. Thus, we choose the pulsewidth sufficiently larger than our "observation window."

```
vin in 0 pulse(0 1 1u 1u 1)
.tran 0.2u 30u
```

A pulsewidth of 1 s proves quite versatile for step response analyses because most of our circuits exhibit a much faster response. Note that the overall transient time is now 30 μs, just long enough to show the response to the input rising edge.

Example A.4

Construct a SPICE netlist for the step response of the circuit depicted in Fig. A.6(a).

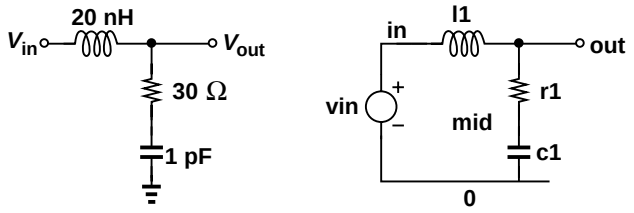


Figure A.6

Solution

We begin with labeling the nodes and the elements [Fig. A.6(b)]. How do we choose the transition time of the step? Ignoring the damping behavior of the circuit for now, we may consider $R_1/(2L_1) = 1.5$ ns as the time constant of the response and hence choose the transition time to be about 150 ps. The netlist is as follows:

```
My RLC Circuit
l1 in out 20n
r1 out mid 30
c1 mid 0 1p
vin in 0 pulse(0 1 0 150p 150p 1)
```

```
.tran 25p 500p
.end
```

Note that the falling transition time is unimportant here.

Example A.5

Suppose we wish to determine the frequency response of the RLC circuit illustrated in Fig. A.6(a). Revise the netlist accordingly.

Solution

We must often study both the transient and the ac response of circuits. For convenience, only *one* file should serve both purposes. Fortunately, SPICE allows us to "comment out" lines of the file by inserting a * at the beginning of each line. We therefore repeat the netlist from the above example, comment out the lines related to transient analysis, and add the lines necessary for ac analysis:

```
My RLC Circuit
l1 in out 20n
r1 out mid 30
c1 mid 0 1p
*vin in 0 pulse(0 1 0 150p 150p 1)
*.tran 25p 500p
*Added next two lines for ac analysis.
vin in 0 ac 1
.ac dec 100 1meg 1g
.end
```

(The letter g at the end of the .ac line denotes 10^9.) As seen above, comment lines can also serve as reminders.

A.2.3 DC Analysis

In some cases, we wish to plot the output voltage (or current) of a circuit as a function of the input voltage (or current). Called "dc analysis," this type of simulation requires that SPICE *sweep* the input across a range in sufficiently small steps. For example, we may write

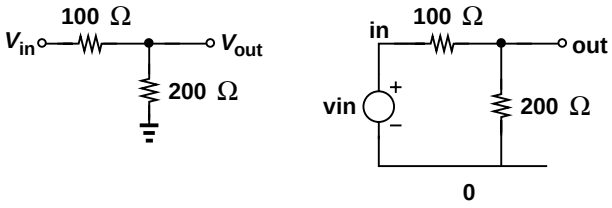
```
vin in 0 dc 1
         Lower  Upper Step
         End    End   Size
.dc vin  0.5    2     1m
```

The vin description specifies the type as dc with a nominal value of 1 V.[4] The dc sweep command begins with .dc and specifies vin as the source that must be swept. The following two entries denote the lower and upper ends of the range, respectively, and the last entry indicates the step size.

Example A.6

Construct a netlist to plot V_{out} as a function of V_{in} for the circuit shown in Fig. A.7(a). Assume

[4] This nominal value is arbitrary and unimportant in dc analysis.



(a) (b)

Figure A.7

an input range of −1 V to +1 V with 2-mV steps.

Solution

We label the nodes and the elements as illustrated in Fig. A.7(b). The netlist can be written as:

```
Voltage Divider
r1 in out 100
r2 out 0 200
vin in 0 dc 1
.dc vin -1 +1 2m
.end
```

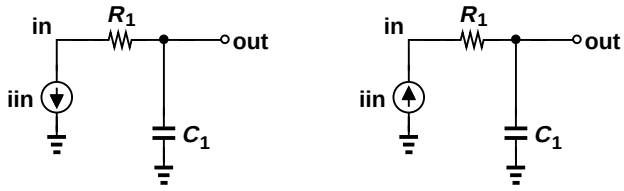
Note that the values of r1 and r2 are not followed by a unit so that SPICE assumes they are expressed in ohms.

A.3 Element Descriptions

In our study of SPICE netlists thus far, we have seen descriptions of resistors, capacitors, inductors, and voltage sources. In this section, we consider the descriptions of elements such as current sources, diodes, bipolar transistors, and MOSFETs.

A.3.1 Current Sources

The definition of current sources for various types of analysis follows those of voltage sources, with the understanding that the current flows *out of* the first node and *into* the second node specified in the description. For example, the current source in Fig. A.8(a) is expressed as



(a) (b)

Figure A.8 Circuits for illustrating the polarity of current sources in SPICE.

```
iin in 0 ac 1
```

for ac analysis.

If the circuit is conf gured as shown in Fig. A.8(b), then we must write

```
iin 0 in ac 1
```

Similarly, for pulse response, the current source in Fig. A.8(a) can be expressed as

```
iin in 0 pulse(0 1m 0 0.1n 0.1n 5n)
```

where the current jumps from 0 to 1 mA with zero delay and a rising transition time of 0.1 ns.

Example A.7

Study the response of the circuit depicted in Fig. A.9(a) to an input current step.

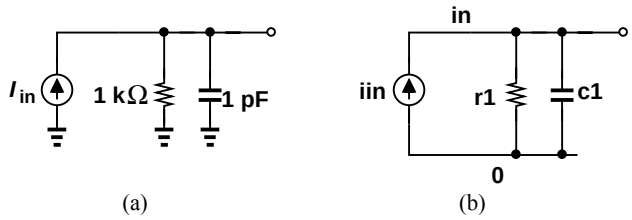


Figure A.9

Solution

Labeling the circuit as shown in Fig. A.9(b) and noting a time constant of 1 ns, we write

```
Step Response Example
r1 in 0 1k
c1 in 0 1p
iin 0 in pulse(0 1m 0 0.1n 0.1n 1)
.tran 20p 3n
.end
```

A.3.2 Diodes

Unlike passive elements studied thus far, diodes cannot be specif ed by a "value." Rather, equation $I_D = I_S[\exp(V_D/V_T) - 1]$ suggests that the value of I_S must be provided. Thus, in the example illustrated in Fig. A.10, we have

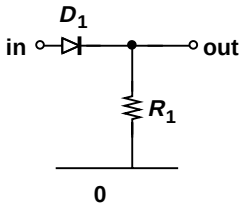


Figure A.10 Simple diode circuit.

```
    Anode Cathode  Is
d1   in    out    is=1f
```

where the element name begins with d to denote a diode, the first node indicates the anode, and the second represents the cathode. The last entry specifies the value of I_S as 1×10^{-15} A.[5]

In some cases, a reverse-biased diode may serve as a voltage-dependent capacitor, requiring that the value of the junction capacitance be specified. Recall that $C_j = C_{j0}/\sqrt{1 + |V_R|/V_0}$, where $V_R < 0$ is the reverse-biase voltage. We must therefore provide the values of C_{j0} and V_0 to SPICE.

The above diode line may then evolve to

```
d1 in out is=1f, cjo=1p, vj=0.7
```

(Note that the third letter in cjo is an o rather than a zero.) SPICE recognizes vj as V_0 for diodes.

Example A.8

Determine the step response of the circuit shown in Fig. A.11 if V_{in} jumps from 0 to 1 V and D_1 satisfies the parameters given above.

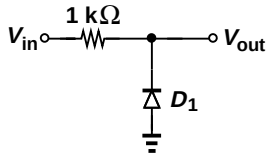


Figure A.11

Solution

The voltage dependence of the junction capacitance of D_1 makes the analysis of this circuit difficult. For V_{out} near zero, D_1 experiences a small reverse bias, exhibiting a capacitance close to C_{j0}. As V_{out} rises, however, the capacitance falls, and so does the *time constant* of the circuit. Thus, SPICE proves quite useful here.

Labeling the circuit in our mind, we write the netlist as:

```
Step Response Example
r1 in out 1k
d1 out 0 is=1f, cjo=1p, vj=0.7
vin in 0 pulse(0 1 0 0.1n 0.1n 1)
.tran 25p 3n
.end
```

As we encounter more sophisticated devices, the number of parameters that must be specified for their SPICE description increases, thereby making the task of netlist construction cumbersome and error-prone. For example, today's MOSFETs require *hundreds* of parameters in their SPICE descriptions. To avoid repeating the parameters for each element, SPICE allows the definition of "models." For example, the above diode line can be written as

```
d1 in out mymodel
.model mymodel d (is=1f, cjo=1p, vj=0.7)
```

[5]Note that f stands for femto and *not* for farad. That is, a capacitor expressed as 1f in SPICE description assumes a value of 1 fF.

Upon reaching the fourth entry in the diode line, SPICE recognizes that this is not a *value*, but a model name and hence seeks a .model command that defines the details of "mymodel." The letter "d" in the .model line specifies a diode model. As seen below, this letter is replaced with "npn" for an npn bipolar transistor and "nmos" for an NMOS device.

Example A.9

Plot the input/output characteristic of the circuit shown in Fig. A.12(a). Assume D_1 and D_2 follow the above diode model.

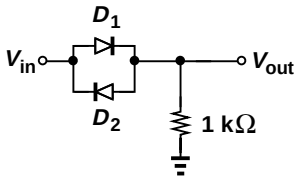


Figure A.12

Solution

Labeling the circuit in our mind, we write the netlist as:

```
Diode Circuit
d1 in out mymodel
d2 out in mymodel
r1 out 0 1k
vin in 0 dc 1
.dc vin -3 +3 2m
.end
```

A.3.3 Bipolar Transistors

The definition of bipolar transistors requires special attention to the order of the terminals. Consider the example shown in Fig. A.13, where Q_1 is expressed as:

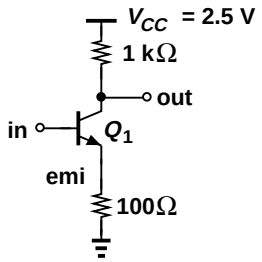


Figure A.13 Common-emitter stage.

```
      Collector Base  Emitter Substrate Model
q1     out       in     emi       0     bimod
```

where the device name begins with the letter q to indicate a bipolar transistor, and the first four nodes represent the collector, base, emitter, and substrate terminals, respectively. (In most cases,

the substrate of npn transistors is tied to ground.) As with diodes, the parameters of the transistor are expressed in a model called, for example, bimod:

```
.model bimod npn (beta=100, is=10f)
```

Example A.10

Construct the SPICE netlist for the circuit of Fig. A.13. Assume the input must be swept from 0.8 V to 0.9 V.

Solution

The netlist is as follows:

```
Simple CE Stage
q1 out in emi 0 bimod
remi emi 0 100
rout out vcc 1k
vcc vcc 0 2.5
vin in 0 dc 1
.dc 0.8 0.9 1m
.model bimod npn (beta=100, is=10f)
.end
```

Two observations prove useful here. (1) The two resistors are labeled according to the nodes to which they are attached. This approach allows us to fin each resistor more readily than if it is simply labeled by a number, e.g., r1. (2) In the above netlist, the term "vcc" refers to two *distinct* entities: a voltage source (the fir t entry on the vcc line), and a node (the second entry on the vcc line).

The model of a bipolar transistor can contain high-frequency effects. For example, the base-emitter and base-collector junction capacitances are denoted by cje and cjc, respectively. The effect of charge storage in the base region is represented by a transit time, tf (equivalent to τ_F). Also, for integrated bipolar transistors, the collector-substrate junction capacitance, cjs, must be specified Thus, a more complete model may read:

```
.model newmod npn (beta=100, is=10f, cje=5f, cjc=6f, cjs=10f, tf=5p)
```

Modern bipolar transistor models contain hundreds of parameters.

Example A.11

Construct the netlist for the circuit shown in Fig. A.14(a), and obtain the frequency response from 100 MHz to 10 GHz. Use the above transistor model.

Solution

Labeling the circuit as depicted in Fig. A.14(b), we write

```
Two-Stage Amp
cin in inb 0.5p
rinb inb vcc 50k
q1 x inb emi 0 newmod
rx x vcc 1k
remi emi 0 2k
```

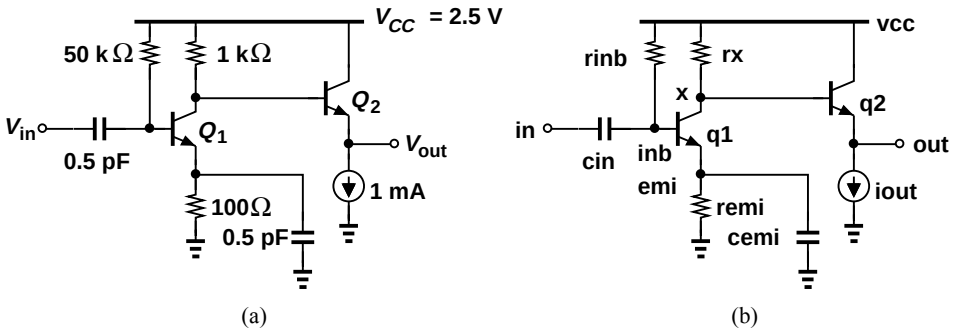


Figure A.14

```
cemi emi 0 0.5p
q2 vcc x out 0 newmod
iout out 0 1m
vcc vcc 0 2.5
vin in 0 ac 1
.ac dec 100 100meg 10g
.model newmod npn (beta=100, is=10f, cje=5f, cjc=6f, cjs=10f, tf=5p)
.end
```

A.3.4 MOSFETs

The defi ition of MOSFETs is somewhat similar to that of bipolar transistors but contains more details regarding the *dimensions* of the device. Unlike bipolar transistors, MOSFETs are both biased and "sized" so as to achieve certain smal-signal properties. For example, both the transconductance and the output resistance of MOSFETs depend on the channel length.

In order to understand how the device dimensions are specif ed, we f rst consider the top view illustrated in Fig. A.15(a). In addition to the channel width and length, we must also provide the source/drain dimensions so that SPICE can calculate the associated capacitances. To this end,

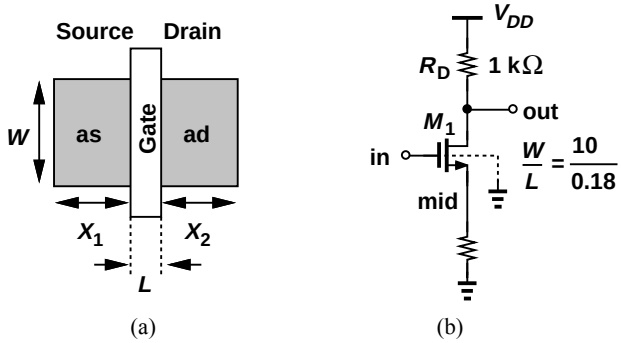


Figure A.15 (a) Top view of a MOSFET, (b) a common-source stage.

we specify the "area" and "perimeter" of the source and drain junctions. Denoted by "as" and "ps" for the source, respectively, (and "ad" and "pd" for the drain), the area and perimeter are

computed as follows: $as = X_1 \cdot W$, $ps = 2X_1 + 2W$, $ad = X_2 \cdot W$, $pd = 2X_2 + 2W$. In most cases, $X_1 = X_2$ and hence $as = ad$ and $ps = pd$.

The value of $X_{1,2}$ is determined by "design rules" for each specif c technology. As a rule of thumb, we assume $X_{1,2} \approx 3L_{min}$, where L_{min} denotes the minimum allowable channel length (e.g., 0.18 μm). In this section, we assume $X_{1,2} = 0.6$ μm.

Now consider the example shown in Fig. A.15(b), where the dashed line attached to M_1 indicates its substrate. Before considering the dimensions, we have:

```
   Drain Gate Source Substrate Model
m1  out  in    mid      0      nmos
```

As with the bipolar transistor, the terminal names appear in a certain order: drain, gate, source, and substrate. Now we add the dimensions:

```
   Drain Gate Source Substrate Model
m1  out  in    mid      0      nmos  w=10u l=0.18u as=6p
+ps=21.2u ad=6p pd=21.2u
```

(The + sign allows continuing a line on the next.) The order of the dimensions is unimportant, but it is helpful to maintain a consistent pattern throughout the netlist so as to make it more "readable." Note that

```
as=6p
```

denotes an area of 6×10^{-12} m^2.

The model of the MOSFET must provide various parameters of the transistor, e.g., mobility (uo), gate oxide thickness (tox), threshold voltage (vth), channel-length modulation coeff cient (lambda), etc. For example,

```
.model mymod nmos (uo=360, tox=0.4n, vth=0.5, lambda=0.4)
```

Note that the default unit of mobility is cm^2/s, whereas the units of other parameters are based on the metric system. For example,

```
tox=0.4n
```

translates to 0.4×10^{-9} m $= 40$ $\mathring{\text{A}}$.

Example A.12

Figure A.16(a) shows a two-stage amplifie . Construct a SPICE netlist to plot the input/output characteristic of the circuit. The substrate connections are not shown with the understanding that the default is ground for NMOS devices and V_{DD} for PMOS transistors.

Solution

Labeling the circuit as depicted in Fig. A.16(b), we write

```
MOS Amplifier
m1 x in 0 0 nmos w=5u l=0.18u as=3p ps=11.2u ad=3p pd=11.2u
rx x vdd 1k
m2 out x vdd vdd pmos w=10u l=0.8u as=6p ps=21.2u ad=6p pd=21.2u
rout out 0 500
vdd vdd 0 1.8
vin in 0 dc 1
```

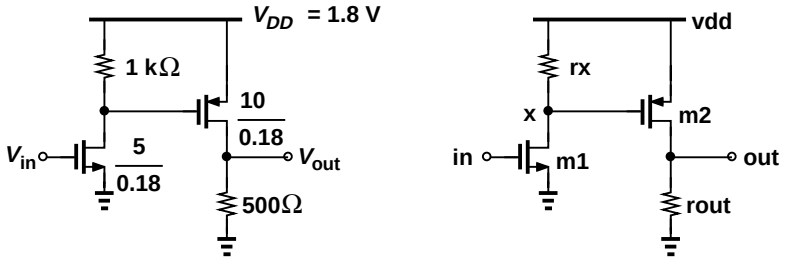


Figure A.16

```
.dc vin 0 1 1m
.model mymod nmos (uo=360, tox=0.4n, vth=0.5, lambda=0.4)
.end
```

For high-frequency analysis, we must specify the junction capacitance of the source and drain areas. As illustrated in Fig. A.17, this capacitance is partitioned into two components: the "area"

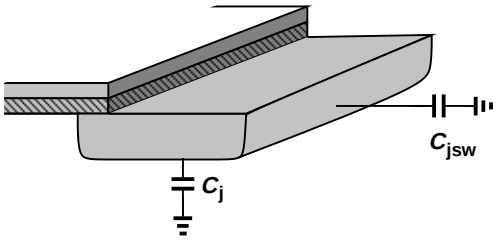


Figure A.17 Area and sidewall capacitances.

capacitance, C_j, and the "sidewall" capacitance, C_{jsw}. This separation is necessary because the values of C_j and C_{jsw} (e.g., per unit area) are typically unequal.

In SPICE, the above capacitance components are def ned differently. The area capacitance is specifie per unit area, e.g., $C_j = 3 \times 10^{-4}$ F/m^2 ($= 0.3$ fF/μm^2), whereas the sidewall capacitance is define per *unit width*, e.g., $C_{jsw} = 4 \times 10^{-10}$ F/m ($= 0.4$ fF/μm). With these specificati ns, SPICE simply calculates the overall junction capacitance as $C_j \cdot ad + C_{jsw} \cdot pd$. For example, with the above values of C_j and C_{jsw}, the drain junction capacitance of M_1 in Example A.12 is equal to:

$$C_{DB1} = (3 \times 10^{-12}\ \text{m}^2) \times (3 \times 10^{-4}\ \text{F/m}^2) + (11.2 \times 10^{-6}\ \text{m}) \times (4 \times 10^{-10}\ \text{F/m}) \quad \text{(A.1)}$$

$$= 5.38\ \text{fF}. \quad \text{(A.2)}$$

Note that, if the area and perimeter values are absent in the netlist, SPICE may use a default value of zero, thus underestimating the capacitances in the circuit.

The source/drain junction capacitances exhibit a voltage dependence that may not follow the square-root equation associated with "abrupt" pn junctions. SPICE allows an equation of the form

$$C = \frac{C_0}{\left(1 + \dfrac{V_R}{\phi_B}\right)^m}, \quad \text{(A.3)}$$

where C_0 denotes the value for zero voltage across the junction, and m typically falls in the range of 0.3 to 0.4. Thus for C_j and C_{jsw}, we specify

```
(cjo, mj)
```

and

```
(cjswo, mjsw)
```

A more complete MOS model may therefore appear as:

```
.model mymod nmos (level=1, uo=360, tox=0.4n, vth=0.5, lambda=0.4,
+cjo=3e-4, mj=0.35, cjswo=40n, mjswo=0.3)
```

where the "level" denotes a certain complexity for the model. In practice, higher levels with many more parameters are used. Similarly, a PMOS model may be constructed as follows:

```
.model mymod2 pmos (level=1, uo=150, tox=0.4n, vth=-0.55,
+lambda=0.5, cjo=3.5e-4, mj=0.35, cjswo=35n, mjswo=0.3)
```

A.4 Other Elements and Commands

A.4.1 Dependent Sources

In addition to the independent voltage and current sources studied above, we may need to incorporate dependent sources in simulations. For example, as mentioned in Chapter 8, op amps can be viewed as voltage-dependent voltage sources. Similarly, a MOSFET acts as a voltage-dependent current source.

Consider the arrangement shown in Fig. A.18, where the voltage source tied between nodes C and D is equal to three times the voltage difference between nodes A and B. For simplicity,

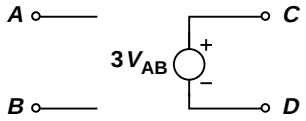


Figure A.18 Voltage-dependent voltage source.

we call (A, B) the "input nodes," (C, D) the "output nodes," and the factor of 3, the "gain." Such a voltage-dependent voltage source is expressed as

```
    Output        Input   DC   Gain
    Nodes         Nodes  Value
e1  c d   poly(1)  a b    0     3
```

Note the element name begins with the letter "e" to signify a voltage-dependent voltage source. The next two entries are the output nodes, with the first representing the positive terminal. The entry poly(1) indicates a first-order polynomial relationship between V_{CD} and V_{AB}. Next, the controlling (input) nodes are specified and the zero is entered to denote a zero additional dc voltage. Finally, the gain is specified In a more general case, this expression can realize $V_{CD} = \alpha + \beta V_{AB}$, where α is the dc value (zero in the above example) and β is the gain (3 in the above example).

Example A.13

The circuit of Fig. A.19(a) employs an op amp with a gain of 500. Construct a SPICE netlist for the circuit.

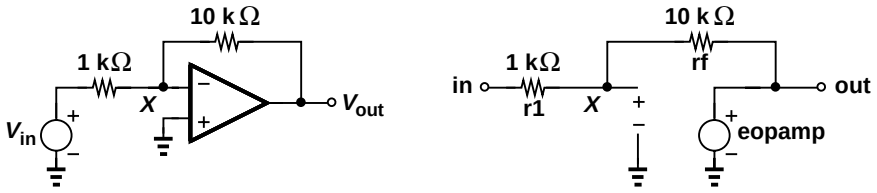


Figure A.19

Solution

We firs draw and label the circuit as shown in Fig. A.19(b). Thus,

```
r1 in x 1k
rf x out 5k
eopamp out 0 poly(1) x 0 0 -500
```

For the voltage-dependent current source depicted in Fig. A.20, the description is as follows:

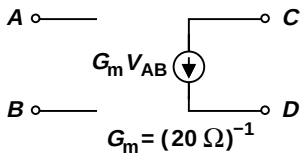


Figure A.20 Voltage-dependent current source.

```
g1 c d poly(1) a b 0 0.05
```

where the letter g denotes a voltage-dependent current source and the gain is specif ed as $1/(20\ \Omega) = 0.05\ \Omega^{-1}$.

Current-controlled voltage and current sources are also described in a similar manner, but they are rarely used.

A.4.2 Initial Conditions

In the transient analysis of circuits, we may wish to specify an initial voltage at a node with respect to ground. This is accomplished using the .ic command:

```
.ic v(x)=0.5
```

This example sets the initial voltage at node X to 0.5 V.

References

1. G. Roberts and A. S. Sedra, *SPICE*, Oxford University Press, 1997.