

ANALOG LAB STUFF
6.301 Solid State Circuits
Lab Reference

Fall Term 2010

September 29, 2010

Contents

1	Lab Reports and Policy	1
1.1	The Report	1
1.2	Check Off	2
1.3	Lab Assistance	2
1.4	Late Lab Policy	2
2	Building Hints	3
2.1	Lab Work	3
2.2	More Lab Hints	4
3	HSPICE Primer	6
3.1	Use of HSPICE	6
3.2	HSPICE Example	6
3.3	HSPICE Hints	8
3.4	Example HSPICE Output File	9
3.5	Using MATLAB® to View HSpice Output	11
3.6	Example MATLAB® Output Graph	12

Chapter 1

Lab Reports and Policy

1.1 The Report

You will prepare a report describing your design effort. Include a labeled schematic of your design and a summary table of parameters comparing calculated, simulated and measured values. Be sure to note any discrepancies between your calculations and observations. The lab report should explain three basic things: how you arrived at your design (including component values and the constraints involved), how you verified your design in the lab (including what your measurements were, and what they illustrate) and what accounts for the difference between your design calculations and the actual measurements. Calculations supporting your discrepancy analysis must be shown.

A fair portion of your grade depends upon what may seem at first to be a rather nebulous quantity: the *quality* of your lab report. Here are some guidelines which lead to a quality lab report:

1. Page 1 — neat schematic with all components labeled with their literals and values.
2. Page 2 — summary sheet, comparing all calculated, simulated, and measured specifications.
3. Page 3 on — analysis, including topology discussion (explain why you used this topology and what its tradeoffs were), bias discussion (what effect will transistor parameter variations have on operating points and performance?), frequency analysis (open and short circuit time constants), simulation results (attach SPICE printouts and graphs), and most importantly results (thoroughly discuss results, analyze discrepancies, justify your explanations, comment on measurement errors, and defend your choice of significant figures, etc.).
4. Your labs must be your own work and will be graded as such. In other words, do not work jointly. You are free to ask questions of the TAs and instructors at any time.
5. Unlike other courses with labs, 6.301 places great emphasis on analyzing the results you have observed. It is expected that students not only report results but also quantitatively analyze their results and compare them to the theory.
6. The lab report must be written so that a reasonably competent engineer can pick up your lab report and be able to understand all of the analysis and design without reference to other sources. This includes error and discrepancy analysis.
7. All calculated quantities (gain, bandwidth, etc.) appearing on the summary sheet must be justified in the write up. State your intermediate results in terms of literals (for example, $\text{gain} = R_1/R_2$).
8. Schematics must be neat and accurate. Label all components with their values and literals (for example $R_5 = 10\text{k}\Omega$). Graphs should be neatly drawn, labeled and dimensioned. Asymptotes, slopes, dc values, etc., should be indicated where appropriate.

9. The basic purpose of the lab report is to convince us that you understand the theory behind the experiments. Thus it is imperative that you explain any discrepancies between theory and practice. Use a few quick calculations to support your arguments.
10. Neat reports get better grades than sloppy reports. This isn't fair, but it's a fact of life. If your report is being read after 40 typed reports, and it is written in cursive with a blunt hard pencil, the grader may not take the time to carefully read every word of your error analysis. That would probably have unfortunate consequences on your final grade.

While brevity is not graded, a report written in a neat, clear, and concise manner will be most successful.

1.2 Check Off

In order to provide feedback to you concerning your performance and as a significant portion of the evaluation process, it is necessary for you to have an interview with the TA after you have handed in your lab report. To make effective use of our available resources (both staff and equipment) you must sign up for a specified time slot during check off week. The availability of the sign-up sheet will be announced in lecture. We suggest that you select your time early, since sign ups are on a first-come, first-serve basis.

The actual check-off procedure is as follows: You must be in the lab ready to demonstrate your circuit by the time of the interview. Be prepared to answer questions about your design and to demonstrate your understanding of the lab exercise and course material. Missed check-offs will NOT be rescheduled. You will forfeit the interview (30% of the lab grade). There are no exceptions to this policy.

1.3 Lab Assistance

Work independently. Discussions with your classmates about circuit topology or lab technique is allowed, but we expect that your design and implementation to be entirely original. The 6.301 staff will gladly answer your questions and assist you with your problems, but please think first, so that all members of the class can have access to the staff. Also remember that we are here to assist you, not to do the exercise for you.

1.4 Late Lab Policy

The due date for the lab report is stated on the lab assignment. All work (design, lab work, write-up) must be finished and turned in on time. In the rare event that you do not complete all required work on time, half credit will be given to labs turned in by Friday 5pm. Please refer to the late policy in the first-day handout.

You cannot pass this course unless you complete all of the laboratory projects on time.

Chapter 2

Building Hints

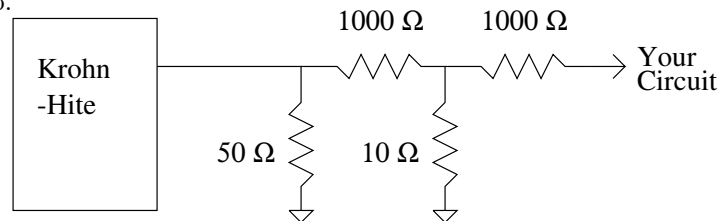
2.1 Lab Work

In addition to the lab kits (the green suitcases that you will check out) there are oscilloscopes, probes, function generators and digital multimeters out on the benches just waiting to be used. Along with this equipment go a few operating procedures:

1. The 6.301 lab area is organized into a number of lab stations with all the equipment already in place. If you find one piece of equipment inoperative at a station, do not “borrow” a replacement from another station. Fill out a “defective equipment tag”, attach it to the defective instrument, and use another station. The equipment is connected to an alarm system, so “borrowing” equipment can be very loud and embarrassing.
2. When properly used, the oscilloscopes can provide accurate measurements of voltages, frequency and risetimes, etc. Be sure to account for the 10 M Ω , 10 pF loading the the 10X probes provide (this is why we require your circuit to drive (at least) a 10 pF load; this load is in addition to the interrail capacitance). The full bandwidth of the scopes is 150 MHz. Be sure the 20 MHz bandwidth limit button is OUT for all of our labs.
3. The signal generators are sometimes cranky. Do not assume that the frequency dial gives an accurate reading of what the generator is putting out. Measure the frequency on the scope. Do not assume that the sine wave output amplitude is constant with frequency.
4. The digital multimeters can be either line or battery powered. Normally they will be operated off the line via the battery charger. There should be no need for you to ever disconnect the power cord. The DMM is only accurate between 100 Hz and 10 kHz on AC volts. Be sure to disconnect the DMM from your circuit before making any high frequency measurements. Also, please turn the multimeters off when you are finished using them.
5. You probably don’t want to use the DMM to measure voltages at high impedance nodes. The input impedance of the meter (20 k Ω) will load the node and give low readings.
6. We have a fancy Hewlett Packard Vector Impedance meter that can accurately measure c_{μ} , with the proper collector-base bias voltage. There should be an instruction sheet explaining its use posted in lab. If you have any questions, ask the TA.
7. If you are unfamiliar with the operating procedure for any piece of lab equipment, ask a TA for help. Don’t wait until after you have accidentally damaged the equipment.

2.2 More Lab Hints

1. Experience indicates that 90% of all circuit malfunctions may be traced to simple causes. Wiring errors, incorrect component values, loose connections, and poor construction techniques cause the most problems. Checking bias points and tracing signals through the circuit are effective debugging techniques. Your circuit obeys the laws of cause and effect. Do not be afraid to apply a little science to your efforts to debug your circuit.
2. You are expected to be aware of anything about the labs that is presented in class. That means that you are responsible for everything that is said in class, handed out in class (including this Lab Reference), and written on the chalkboard in lab.
3. Assume typical hybrid-pi parameters. If you have to go to lab and make a few measurements (on β and c_μ for example), do it! The better your model is, the better your design will be. You can make arguments that the spec sheet data is pessimistic, if you have data to back you up.
4. You are expected measure all the components used in your circuit. Nothing can be attributed to “resistor tolerances”, unless you have measured the resistors and calculated new results, run another SPICE simulation, or otherwise proven that *it is, in fact, true*.
5. Your circuit must be completely free of oscillations at checkoff. Oscillations are any periodic signals that you can see at the output of your circuit that you aren’t putting in at the input.
6. Neatness REALLY matters! Try to keep lead lengths short to minimize parasitic inductances. Keep wires out of the signal path. Don’t be afraid to clip component leads to make a neater, tighter layout. (See the section on Nerd Kit layout for more hints on layout and minimizing parasitic capacitances.)
7. At high frequencies, familiar circuit elements behave mysteriously. Lengths of wire, ground busses, and small valued resistors look like inductors (for this reason, keep leads short and use only one ground bus for your whole circuit). Large valued resistors look like capacitors due to stray shunt capacitance. Electrolytic capacitors look inductive (for this reason, bypass all electrolytic capacitors with 0.1 μF ceramic capacitors).
8. For high frequency measurements, the coax cable with alligator clips does not work very well to bring the signal into the circuit. Use the BNC connector on the chassis of your lab kit with short leads.
9. Since we are building a very high gain amplifiers, you will need a very small input voltage to test your circuit. Since the signal generators in lab cannot effectively produce small signals, you must build an attenuator for your circuit. Also, be sure to properly terminate the output of the signal generator. Make your measurements from the attenuator input to your circuit output (you will probably not be able to measure the 10mV signal at the attenuator output). Be sure to carefully measure the exact attenuation ratio.



10. You shouldn’t have any high impedance nodes that have to be lifted from the protoboard.
11. The output capacitance loading ($C_L = 10$ pF) is provided by the scope probe. However, some of the probes in lab are 10 pF probes, some are 13 pF, and some are 67 pF! Be sure that you are using the proper probe.
12. Open circuit time constants yield a conservative result for the bandwidth. In fact, it may not be possible to to make $\sum \tau_{j_o}$ small enough to meet the bandwidth specification for Lab 2. A circuit that

fails the OCT bandwidth spec by a small margin ($\sim 20\%$) will probably meet spec if all the individual τ_{jo} 's are about equal. That is why we use HSPICE to check our calculations and calculate a more exact result for the bandwidth. The point is to use the OCT method to optimize your design and get the best paper design you can before you go to HSPICE or go into lab.

13. Please don't approach the staff for help until you have fully analyzed your circuit (this includes gain, operating point, and open circuit time constants).

Chapter 3

HSPICE Primer

3.1 Use of HSPICE

Do not use HSPICE as a design tool. It is only to be used to verify your hand calculations and (if you wish) to explain the error analysis of your lab results. If you are spending hours and hours running HSPICE simulations, STOP! You are doing something wrong.

3.2 HSPICE Example

This example shows the simulation of a cascode amplifier (shown in Figure 3.1) with HSPICE on Athena¹.*.

To begin, login to Athena*, make a spice subdirectory off of your home directory, and create the input textfile `cas.sp` in it.

```
mkdir spice
cd spice
emacs cas.sp &
```

The file `cas.sp` is the input file to HSPICE. It is a list of elements, the nodes to which they are connected, and a list of commands. This file asks for the operating point, an AC analysis, and a plot of the voltage magnitude at the output (given $V_S = 1$ at the input for the AC analysis). It contains the data shown in Figure 3.2.

To run HSPICE on the file, you must add the HSPICE commands to your search path. That is accomplished with the following command.

```
add hspice
```

Now, you can run HSPICE commands in that window. Unless you have a good memory and can read really fast, you will need to redirect the output of HSPICE to a file. That way you can print it out and look at it carefully. This is done with the command:

```
hspice cas.sp > cas.out
```

HSPICE will simulate the circuit and write the (four page!) result to the file `cas.out`. You may then edit the file in Emacs (if you wish, perhaps taking out some of the blank lines, and saving only the important data) or print it out using :

```
enscript -2rG -h -Pprintername cas.out
```

¹We assume that you have an Athena account, that you know some simple Unix commands, and that you know a little Emacs. If you do not, get an account NOW, take some Athena minicourses, and talk to the TA.

*Athena is MIT's UNIX-based computing environment. OCW does not provide access to it.

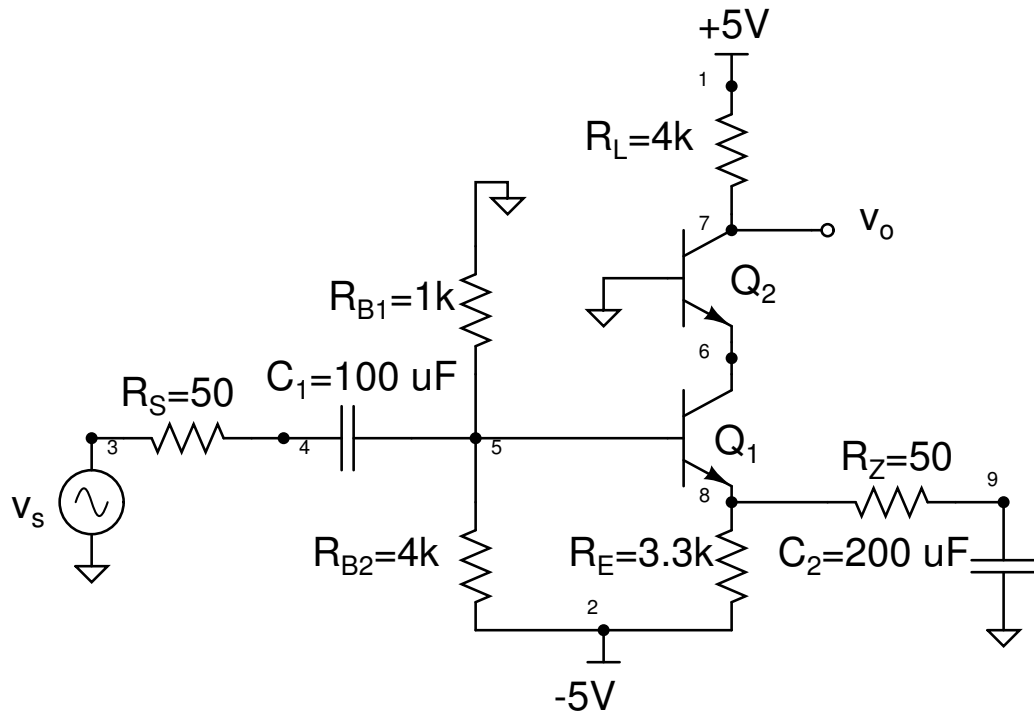


Figure 3.1: Cascode amplifier schematic (small numbers are HPISCE nodes)

```

cascode example simulation
.model npn1 npn is=15fA bf=200 vaf=50 rb=50 cje=2pF tf=250ps cjc=2pF
q1 6 5 8 npn1
q2 7 0 6 npn1
vcc 1 0 5
vee 2 0 -5
vs 3 0 ac 1
rs 3 4 50
rb1 0 5 1k
rb2 5 2 4k
re 8 2 3.3k
rz 8 9 50
rl 1 7 4k
c1 4 5 100uF
c2 9 0 200uF
.options nomod post
.op
.ac dec 10 1 200X
.plot v(7)
.end

```

Figure 3.2: Cascode amplifier HSPICE input file

You may continue editing files in Emacs and running simulations with HSPICE until you finish.

Another way you can look at your data is with GWAVE. Use "gwave ckt1.ac0" (or other file name) to view, convert and print HSPICE (or other SPICE) simulation results. To produce plots, go to *View*→*Variable Names*→*yourfile* and drag the variable(s) of interest into the desired graph pane. For more detailed information on GWAVE, (gwave.sourceforge.net).

There are other options including the MATLAB hpice toolkit, (www.cppsims.com/download_hpice_tools), and LTSPICE (www.linear.com/designtools/software)

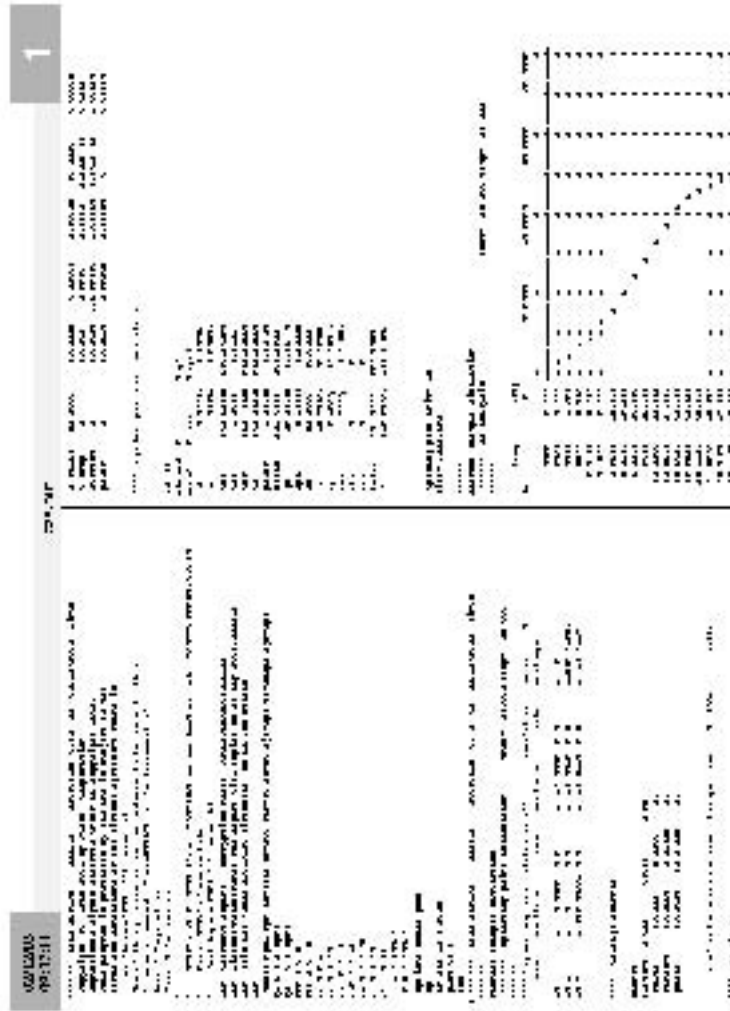
That is all. If you have any questions, ask the TA.

3.3 HSPICE Hints

1. HSPICE is not a design tool (neither are spreadsheets). It is only to be used to check your hand calculations. Do not run HSPICE simulations until you have fully analyzed your circuit (this includes all open circuit time constants). This insight that you will gain from doing calculations by hand is very valuable.
2. Do not use vendor supplied SPICE models. They are often abnormally optimistic. Use only the model you calculated in Lab 1.
3. HSPICE is only as good as the model you give it. It doesn't have any information about the hybrid-pi model that you don't have. If the list of hybrid-pi parameters (under the "bipolar junction transistors subckt" heading) from HSPICE doesn't match your calculations exactly (within a few percent), then something is wrong with your model. Make sure that your capacitor values match exactly.
4. Exception: HSPICE thinks that $c_{jef} \approx c_{je0}$, but Gray and Meyer (correctly) says $c_{jef} = 2c_{je0}$.
5. Similarly, HSPICE uses the same equations you do to calculate operating point voltages and currents, gain, power consumption, etc., so no discrepancy can be attributed to "different calculation methods". (The exception to this is the bandwidth, of course. HSPICE does not use open circuit time constants).
6. You must make the HSPICE values for c_{μ} match those you have measured or calculated for your devices. There are a couple of ways to accomplish this. The preferred method is to make some measurements, and fit the model from Gray & Meyer to the data by doing a non-linear regression (this, however, is far beyond the scope of this course). Alternatively, you can fudge the model by trying to get a simple "best-fit" to the operating points that you use. Or, you can forget the c_{μ} model, set MJC equal to 1 and VJC equal to 1000, and use a different value of CJC for each of your transistors. Be sure to state your method in your lab report.

3.4 Example HSPICE Output File

Remember replace these two pages with the actual enscript printouts.
 enscript -2rG cas.out



3.5 Using MATLAB® to View HSpice Output

Professor Michael Perrott has written a convenient MATLAB interface for HSpice on Athena*. It allows you to import the output of your spice simulations to Matlab. You can then use MATLAB to view the HSpice output, or to perform data processing on it. The alternative interface, AWaves, is considerably less user friendly. The package is located in the 6.301 locker. To download into your home directory and untar it, type:

```
attach 6.301
cp /mit/6.301/matlab/hspice_toolbox.tar ~/
cd ~/
tar -xvf hspice_toolbox.tar
```

This will create a directory called `Hspice_toolbox` in your home directory containing the files you need to use it. There are good directions in the files `document.ps` and `document.pdf` which describe the tool functionality. To get started, fire up

```
add MATLAB:
MATLAB
```

You need to add the HSpice package to your path, and be in the directory containing your HSpice files. Commands from here on are typed into MATLAB.

```
addpath /mit/< your username here >/Hspice_toolbox
cd spice (or wherever you keep your spice files)
```

Next, load the file containing the spice data into MATLAB. If you were running an AC simulation, it will be named something like `lab2.ac0`. A transient simulation will have a filename like `lab2.tr0`. To load and examine your data, type:

```
mydata = loadsig('lab2.ac0')
lssig(mydata)
```

This will display the various fields of data available to be plotted, with the same nodenames as they were named in the spice input file. The fields can be plotted with the `plotsig` command. For example, to plot the node named `Vout`:

```
plotsig(mydata, 'Vout')
```

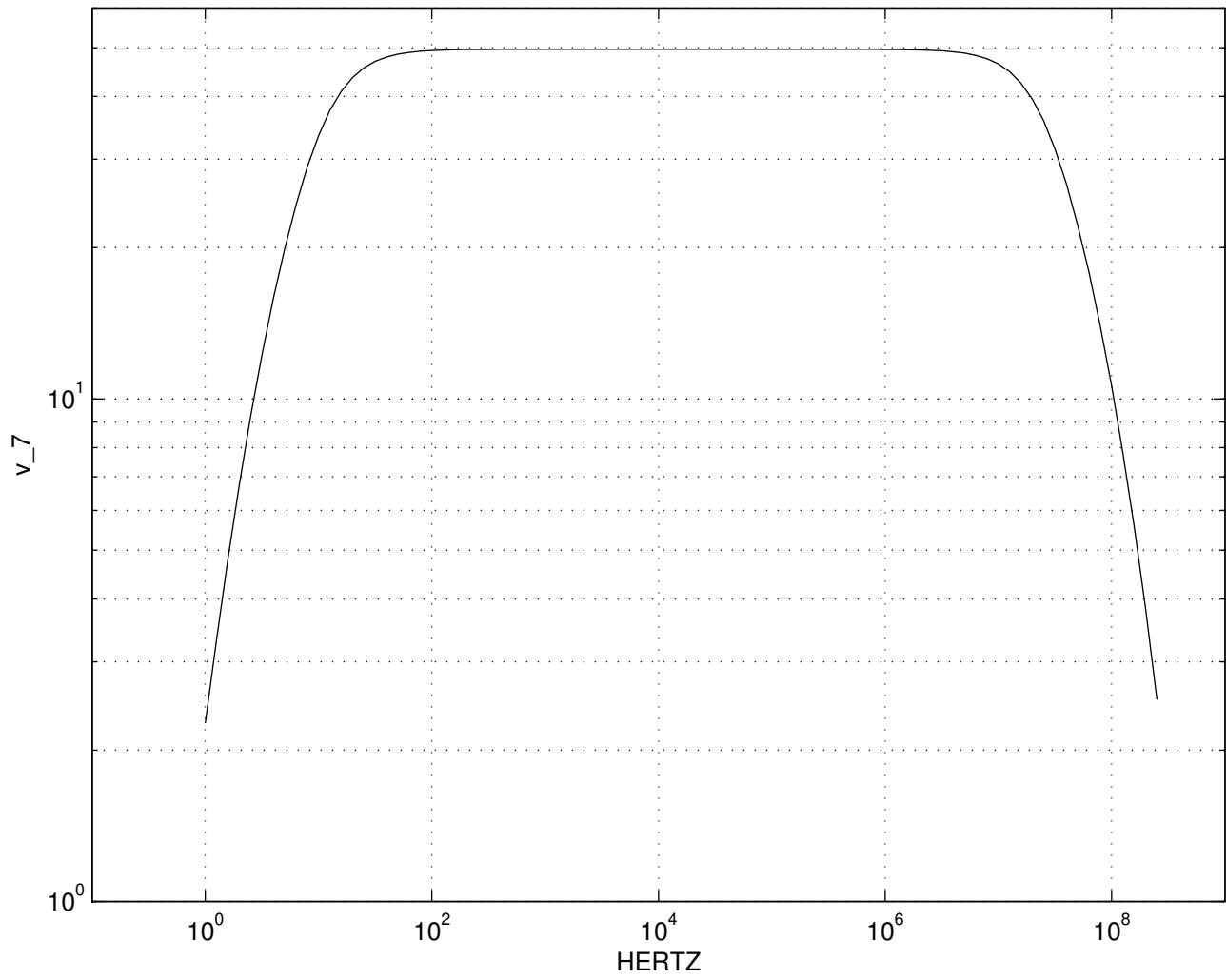
If you are looking at the frequency response of your circuit, you want a log frequency axis, and you need to tell MATLAB whether you are looking at the magnitude or the phase of your response. If you want a typical bode plot:

```
plotsig(mydata, 'abs(Vout); angle(Vout)', 'logx')
```

Other standard MATLAB commands can be used on the data to be plotted as well. If you want to load the output data into a MATLAB array to process it directly, the `evalsig` command is used:

```
vout = evalsig(x, 'Vout')
```

3.6 Example MATLAB® Output Graph



MIT OpenCourseWare
<http://ocw.mit.edu>

6.301 Solid-State Circuits
Fall 2010

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.