12

This chapter shows the basics of simulating oscillators to determine several basic specifications.

Lab 12: Oscillator Simulations

About this lab exercise: This lab exercise is in two parts:

Part 1: You use a prebuilt example oscillator file and perform one simulation.

Part 2: You build an VHF VCO and preform several simulations.

OBJECTIVES

- Use OscTest Element to get frequency and S-parameter information.
- Build an oscillator and simulate numerous performance tests.

PART 1: PROCEDURE – Oscillator Example

1. Create a new project named: oscillator

2. Copy the example design into your directory

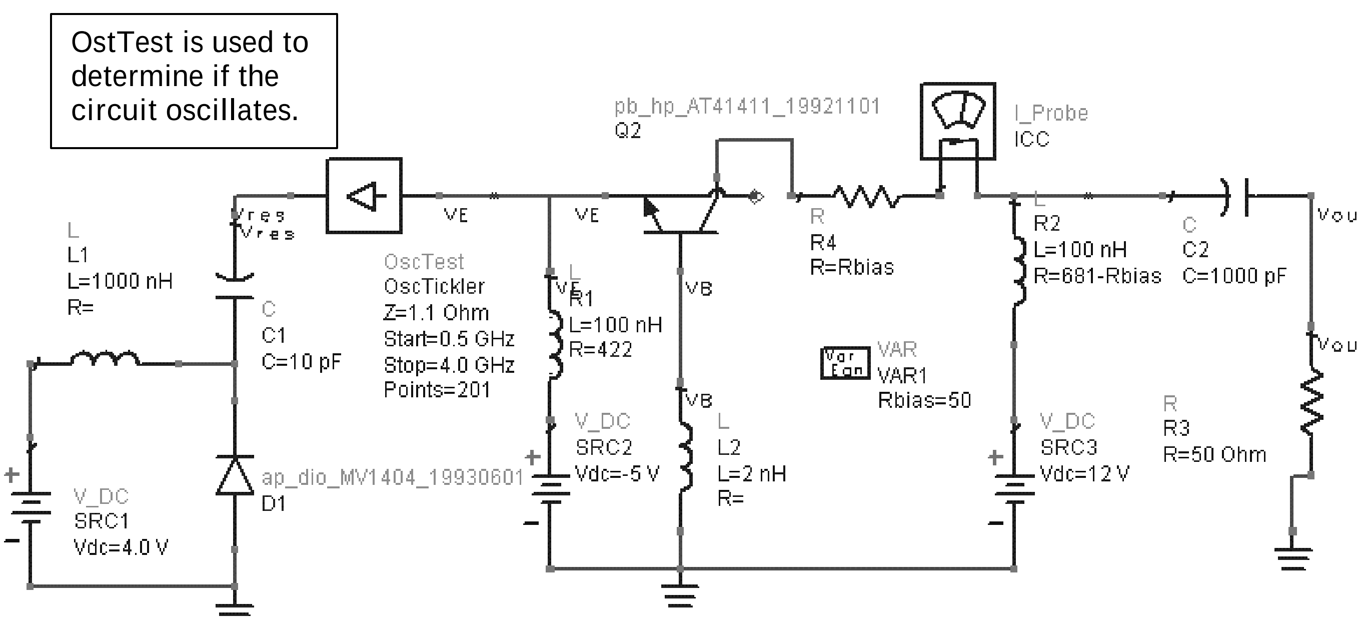
From the **Main window**, use the **Copy Design** command. When the dialog box appears, browse the example directory and copy from:

examples \ tutorial \ LearnOSC_prj \ networks \ Osctest_VCO.dsn

To Path: **oscillator_prj \ networks**

3. Open the new oscillator schematic design

a. Zoom in on the device and notice that it has two emitters. You can remove the pin labels in the Pin/Tee tab of the Options>Preferences dialog. Notice that the unconnected emitter is showing red (unconnected pin) – this is OK.

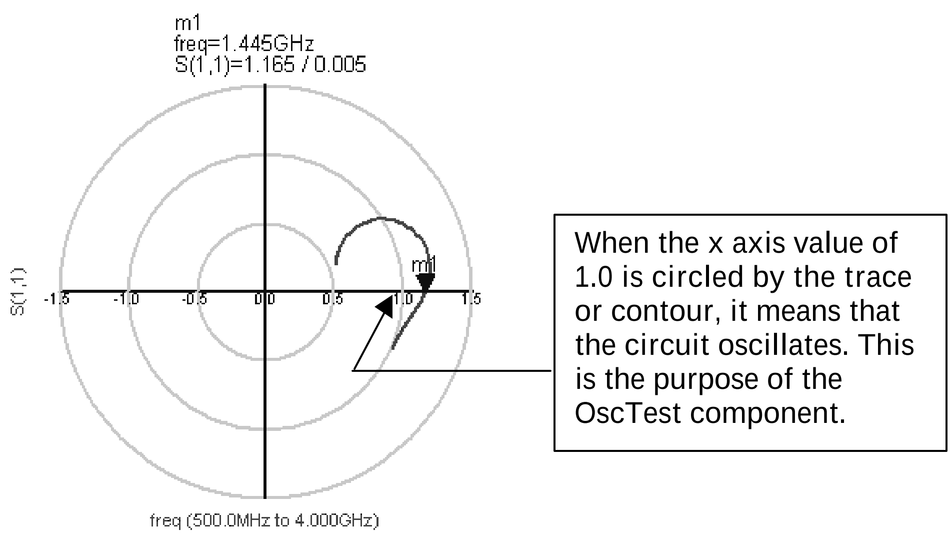


b. Notice the Os**cTest** component is named: OscTickler. Go to the S-parameter palette and locate this component. Basically, this component is an S-parameter simulation controller specifically designed to be inserted in series with the resonator. Therefore, you can simulate without another controller or ports.

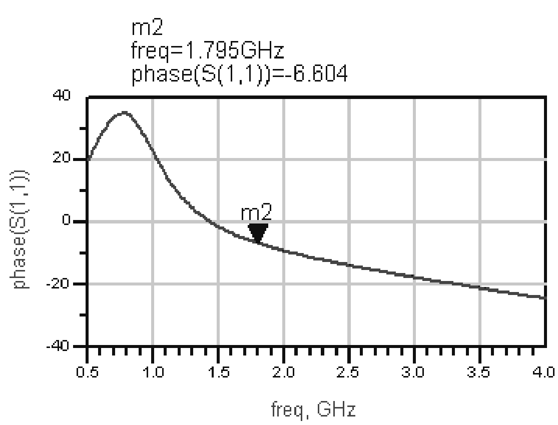
4. Simulate (dataset name: osc_test) and Display the results

a. Open a data display (save as: osc_basics) and plot S-11 on a polar plot.

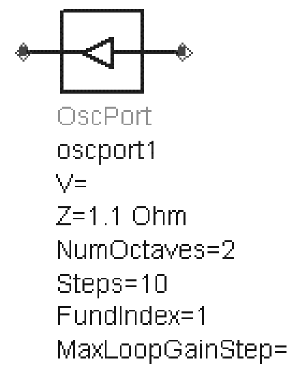
b. Put a marker on the trace where it crosses zero on the x-axis as shown. While this point indicates a frequency where phase is zero, it is not necessarily the desired frequency of oscillation. However, at the location where S11 = 1 + J0, a trace that circles this point indicates oscillation and that is the concept that OscTest validates.



c. Insert a rectangular plot of phase and you will see that near 1.8 GHz (designed value) the phase is not at zero either. But this is OK.



5. Replace the OscTest with an OscPort (use default settings) from Harmonic Balance

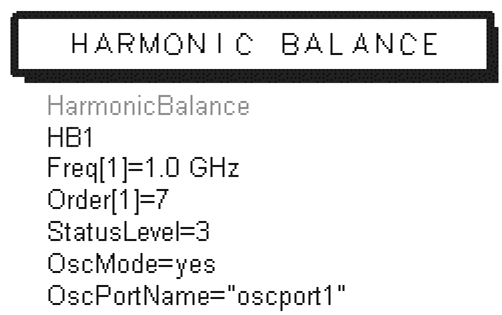


6. Insert a Harmonic Balance simulation controller

a. Insert a HB controller.

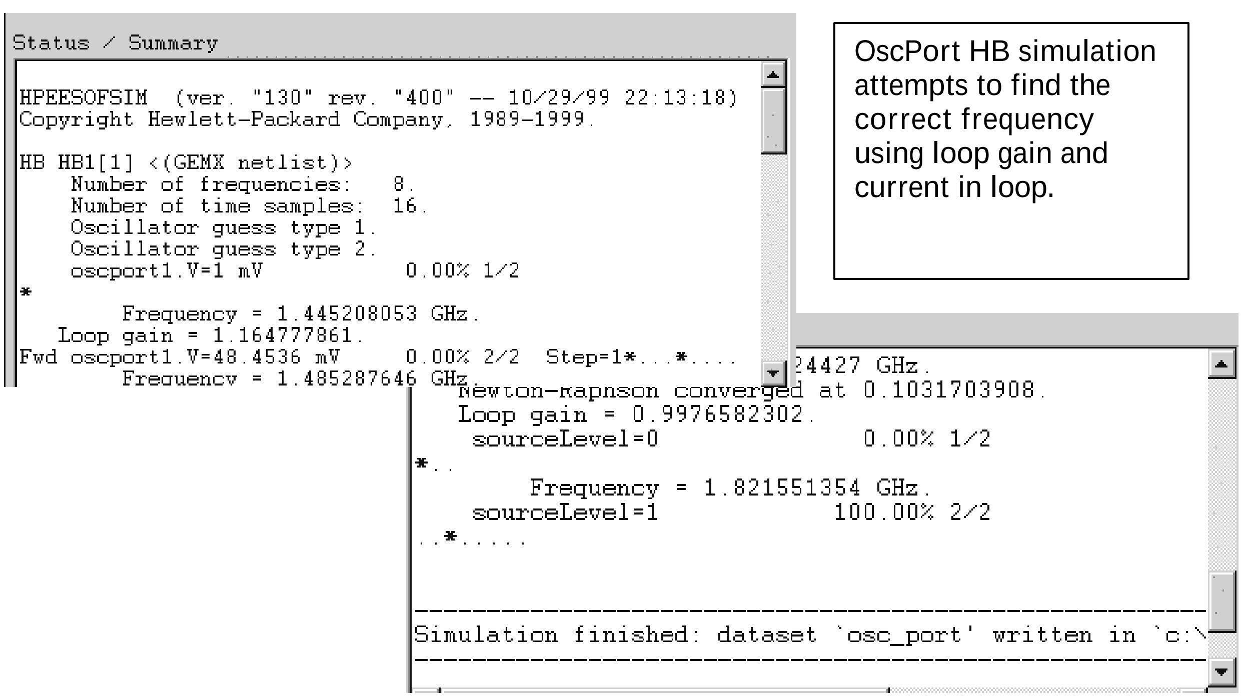
b. Set the Freq and Order as shown where 7 harmonics of the freq are chosen because 3, 7, 15, 31, etc are better for memory allocation during simulation which uses binary 2, 4, 8, 16, etc. Therefore, the DC component [0] plus 7 more fit better for 8 places of data storage.

c. Go to the Display tab and click the StatusLevel, OscMode, and OscPortName boxes and then set them as shown. Increasing the status level to 3 will output more information to the status window, as you will see.



7. Simulate (dataset: osc_port) and display the results

a. Simulate and watch the status window.



OscPort HB simulation attempts to find the correct frequency using loop gain and current in loop.

b. In the data display, write two equations: 1) current in the loop which is index [0] at the I probe and 2) frequency of oscillation from the HB oscillator simulation result as shown.

Eqn loop_current = real (osc_port..ICC.i [0])

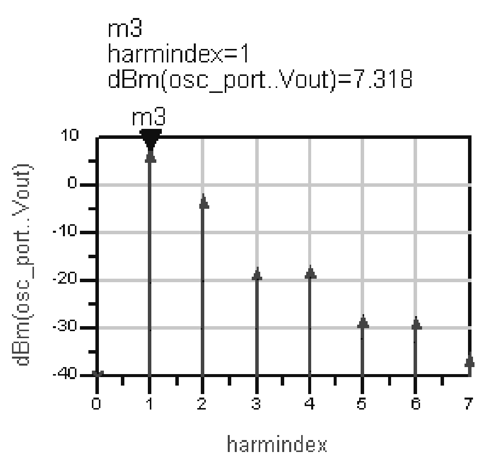
Eqn osc_freq=osc_port..freq[1]

Note on freq[1] value: Index value freq [1] in the case of an oscillator simulation, using the OscPort and HB, means the calculated frequency of oscillation and not the Freq [1] setting in the HB controller as you will see.

c. List the equation results in the data display.

loop_current	osc_freq
-0.011	1.806E9

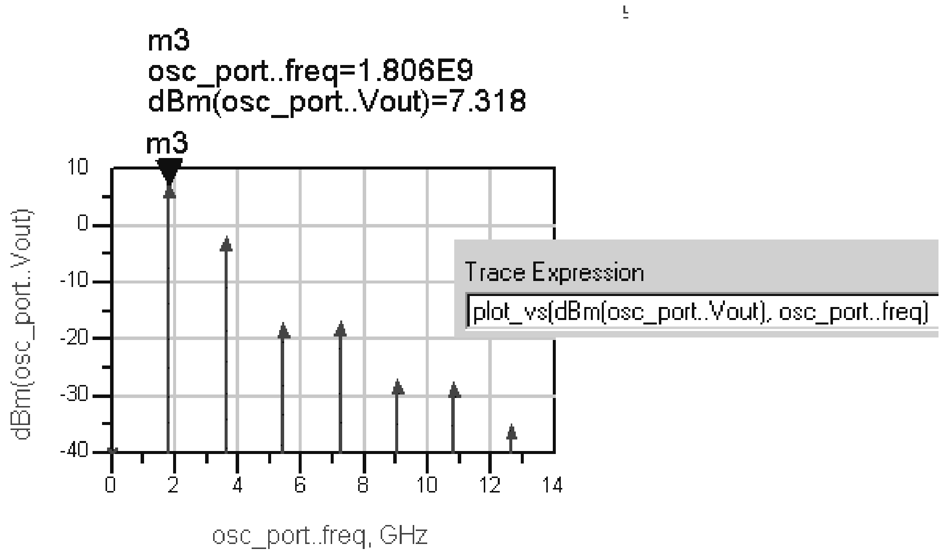
d. Plot **dbm of Vout** and you will see that the x-axis is given as an index value named harmindex, instead of frequency. This is because the calculated values from the OscPort HB simulation must be plotted against the calculated values of freq and not the freq variable settings in the HB controller. Insert a list of **freq** and you will see.



harmindex	osc_port..freq
0	0.0000 Hz
1	1.806GHz
2	3.611GHz
3	5.417GHz
4	7.222GHz
5	9.028GHz
6	10.83GHz
7	12.64GHz

Note on converting harmindex to freq: To plot the spectrum (freq) against the value in dBm, you will have to use the vs function (next step).

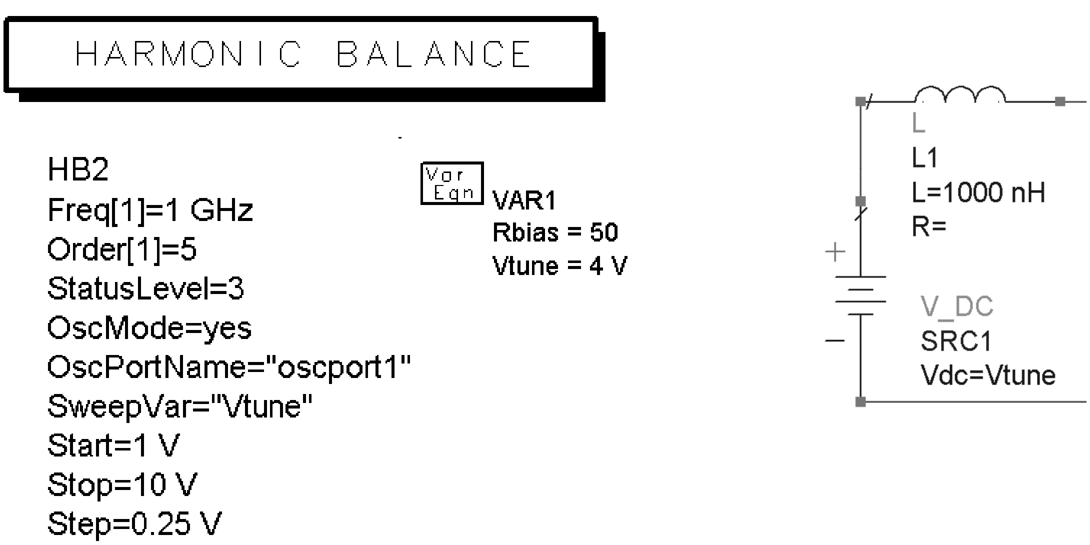
e. Insert a new plot of the dBm of Vout vs freq. The spectrum should look like this where the marker is on the frequency of oscillation.



8. Set up a sweep of the tuning voltage

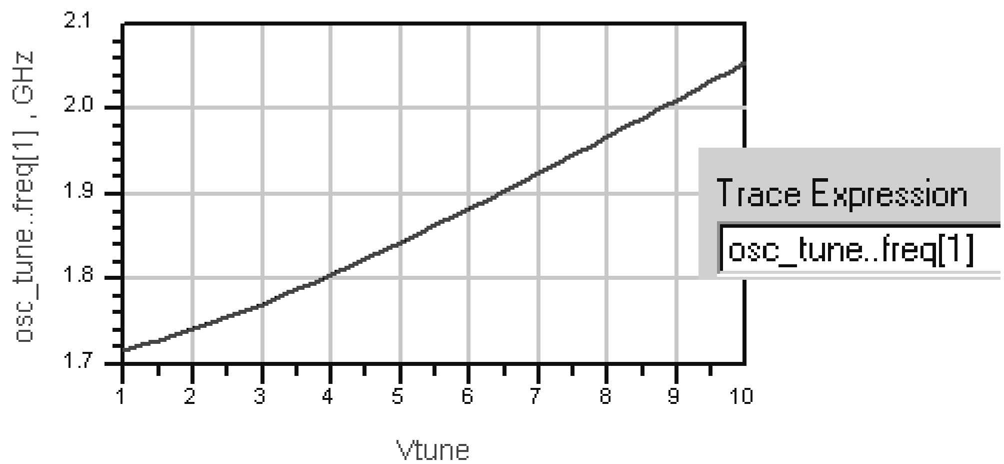
a. Modify the current HB controller or insert another (deactivate HB1), setting start, stop, and step as shown.

b. Edit the VarEqn to add Vtune and be sure to set: **Vdc = Vtune**.

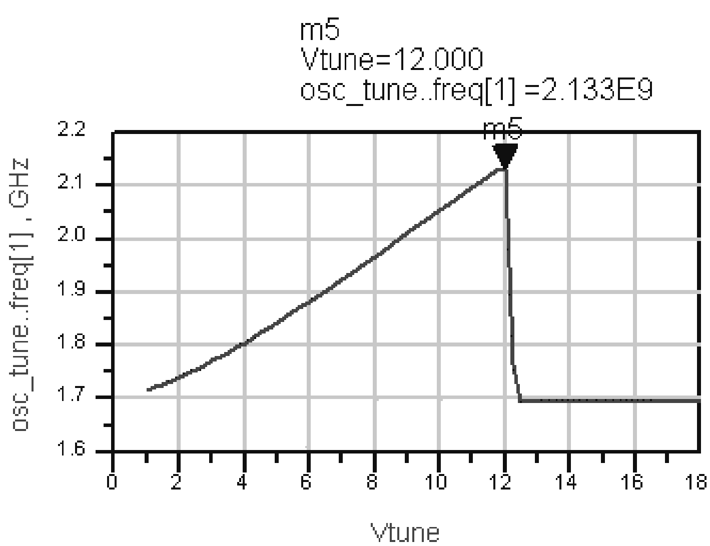


9. Simulate (dataset: osc_tune) and plot the results using an equation

a. After simulating with a new dataset name, insert a plot of freq and edit the trace to index freq to: **[1]**. This will result in a plot of the oscillation frequency at each value of Vtune.



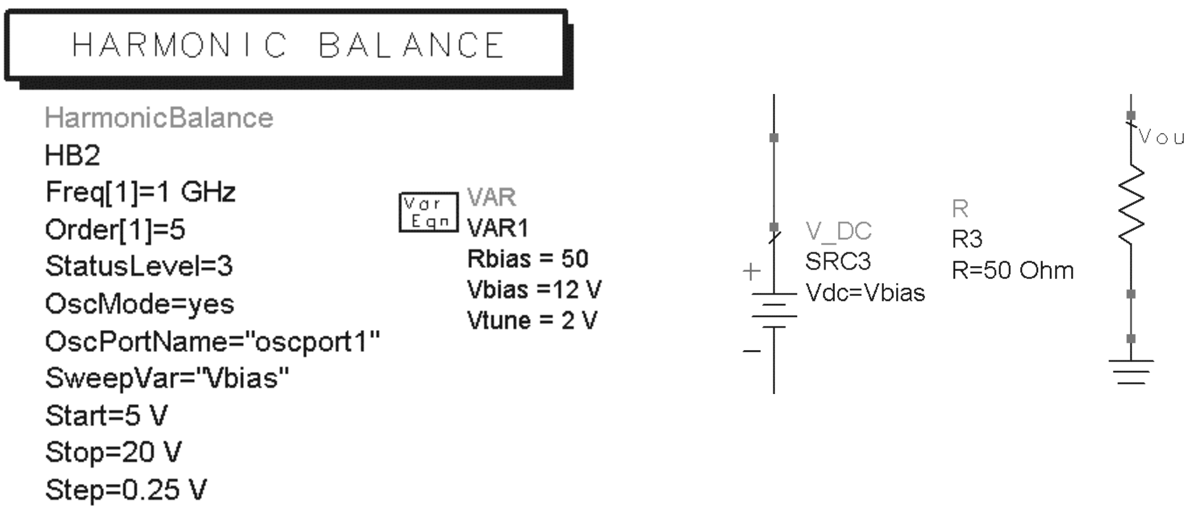
b. Increase the tuning range to 18 volts and watch the plot update. As you will see, near 12 volts, the oscillator is no longer working in a linear manner. In fact, it appears that the diode is now strictly a resistor instead of a variable capacitance. This is because the breakdown of this diode is exactly 12 volts in the model.



10. Frequency pushing by varying the bias

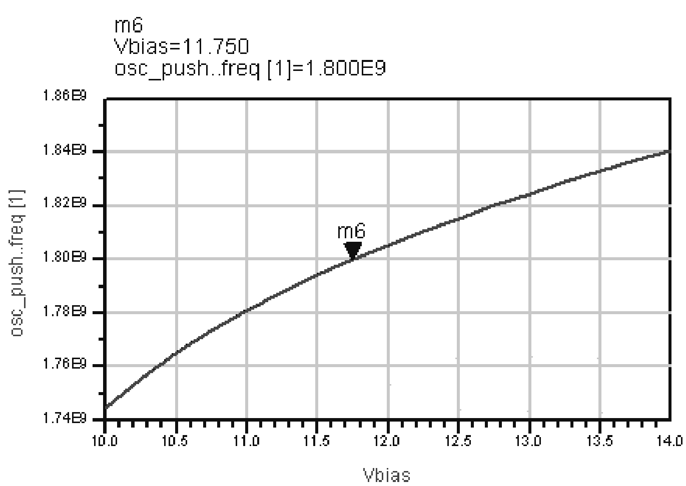
a. Add another variable Vbias initialized to 12 volts (original value) and assign it to the Vdc bias source as shown here.

b. Set the HB controller to sweep Vbias instead of Vtune as shown.



c. Simulate with the dataset name: osc_push.

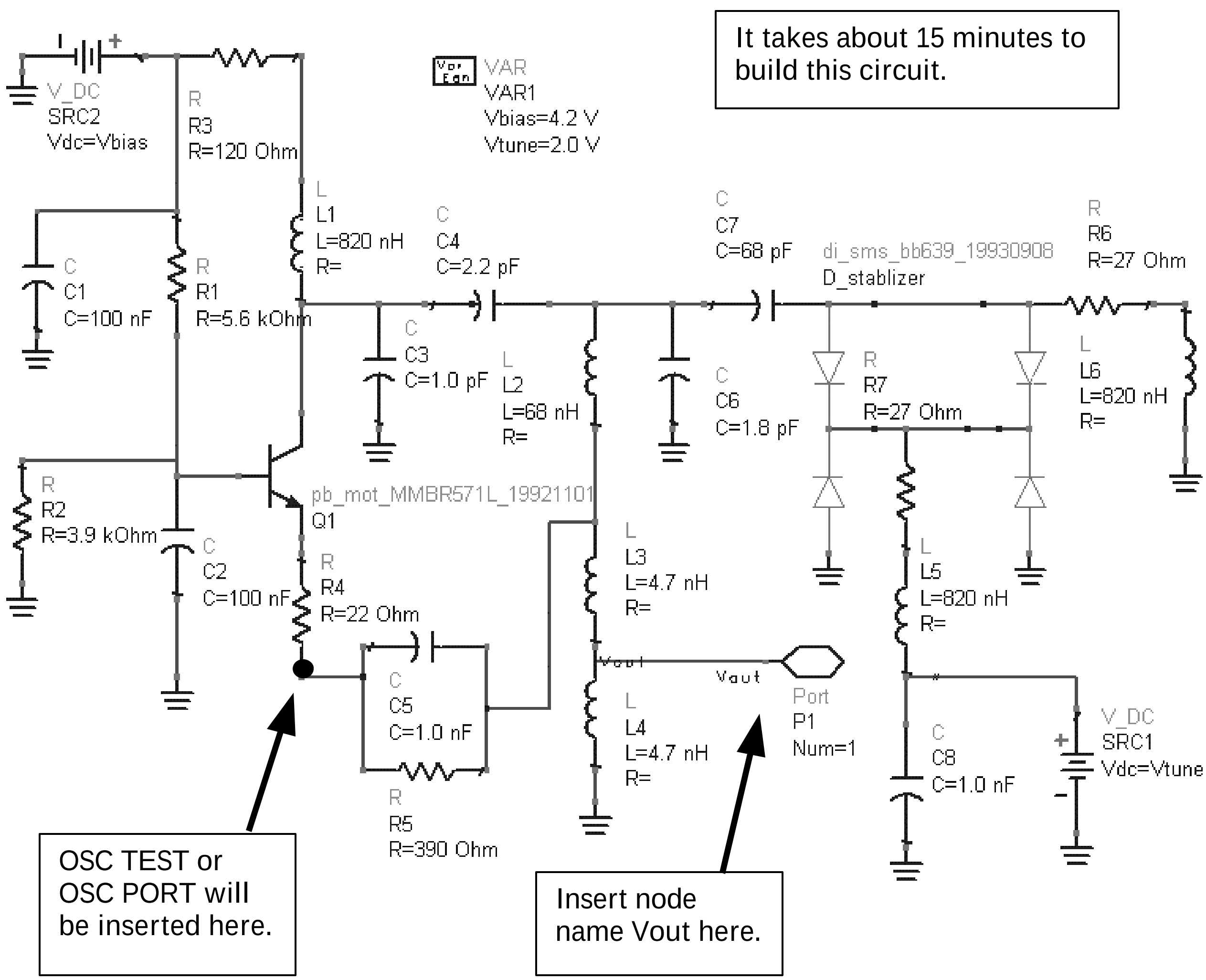
d. Plot the oscillation frequency = freq [1]. You will see that it varies with the variation in bias voltage as shown.



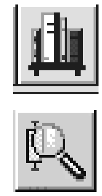
PART 2: PROCEDURE for VHF VCO

<u>DIRECTIONS</u>: This part of the lab has few instructions. So, be careful and use all the skills you have learned so far in the course to construct and test this transistor-based VHF production oscillator. The transistor is from the ADS analog parts library and the diodes (stability) are also from the library. Also, note the save VarEqn used to set the values of bias and tuning.

THE NEXT PAGE HAS SOME SUGGESTED STEPS FOR BUILDING THIS CIRCUIT.



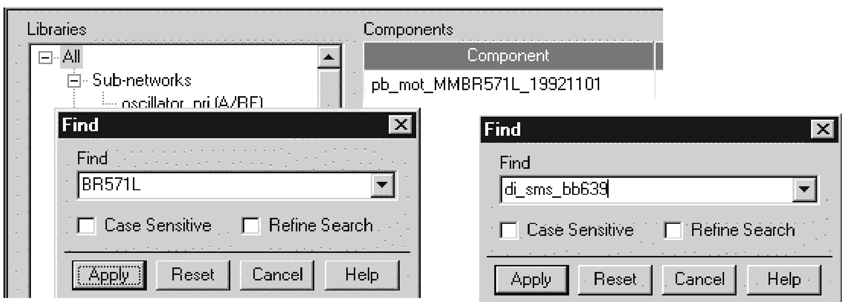
Suggested steps to start building the circuit left to right:



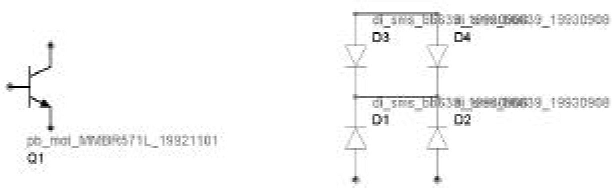
a. Find the library diode and transistor: Click on the library icon and then click the find icon and type in the first few numbers and letters as shown.

di_sms_bb639_19930908 | 4 diodes

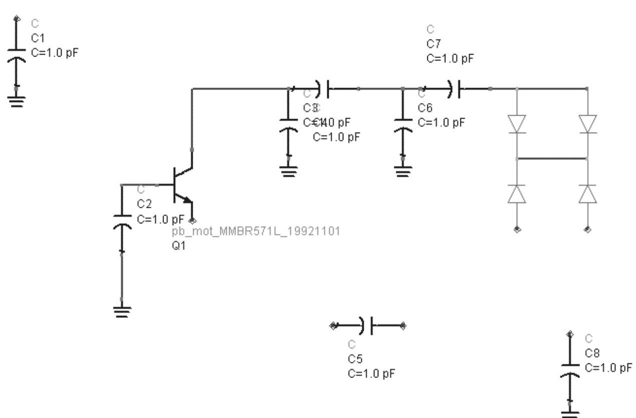
pb_mot_MMBR571L_19921101 | 1



b. Insert 4 diodes and 1 transistor. Connect and wire the diodes as shown and position the transistor to the left of the diodes



c. Insert eight (8) capacitors from left to right, wire them together, and put grounds as shown.



Use F5 to move the component text if desired.

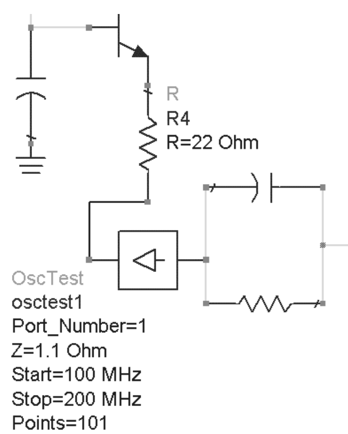
d. Refer back to the first step showing the entire schematic and continue building by inserting the inductors and then the resistors in the same left to right manner. Also, add all the grounds and sources. Then wire the circuit together as shown in the schematic. Don't forget the Vout node name.

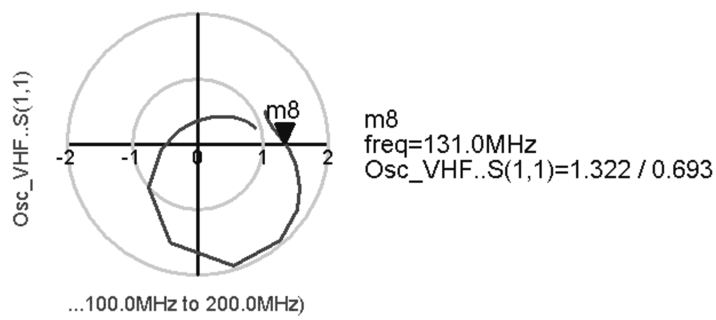
   Finish the circuit and check it - assign all the values and variables.

11. Insert the OSC TEST and Simulate to check for oscillation

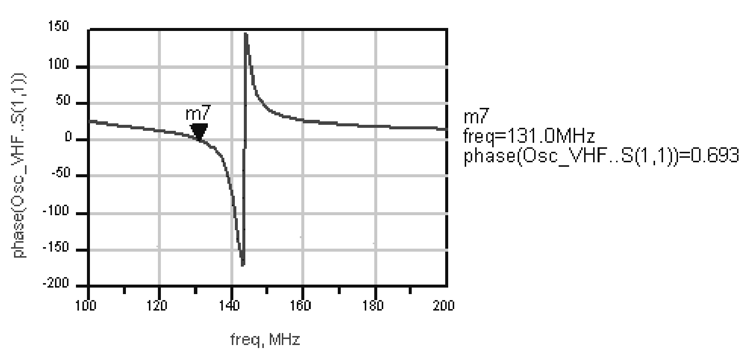
a. Insert an OSCTEST in the emitter leg of the transistor, after the resistor, as shown here.

b. Set the simulation from 100 MHz to 200 MHz at 101 points. Simulate and plot the results. As you can see, the trace encircles the reflection coefficient real value of 1. Therefore, the circuit oscillates. Also, the marker is at 131 MHz which is the designed value for Vtune = 2 V.





c. You can also plot the mag and phase results if desired.



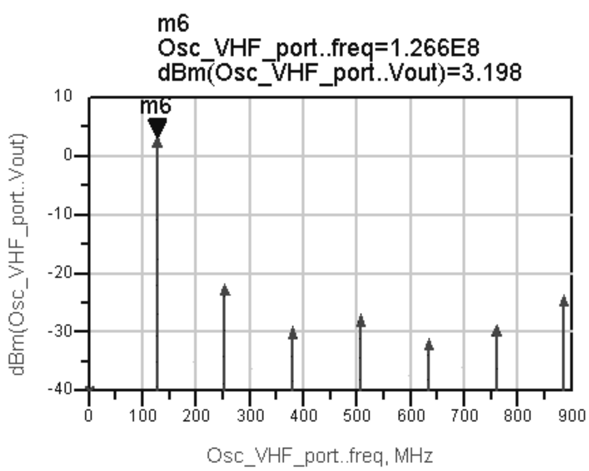
12. OSCPORT HB simulations

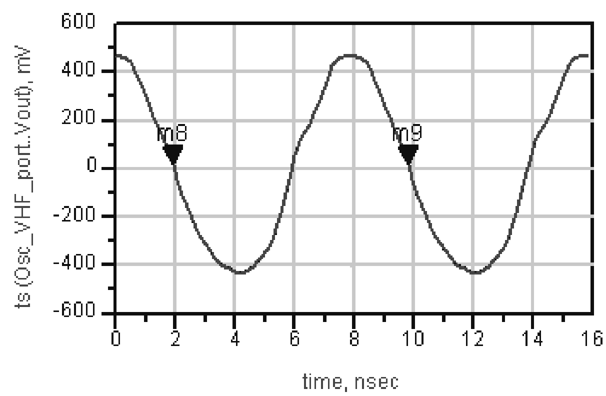
a. Replace the OscTest with an OscPort and insert a HB controller. Remember that the Freq[1] value should be close to the oscillation frequency or the HB simulator will have difficulty converging on a solution.

b. Set a dataset name (Osc_VHF_port) and simulate.

c. Plot dBm of Vout vs freq and your results should be similar to the results shown here, close to 127 MHz which is slightly different than the designed value of 131 MHz.

d. Plot the waveform using the ts function on Vout with markers and an equation to verify the frequency as shown.

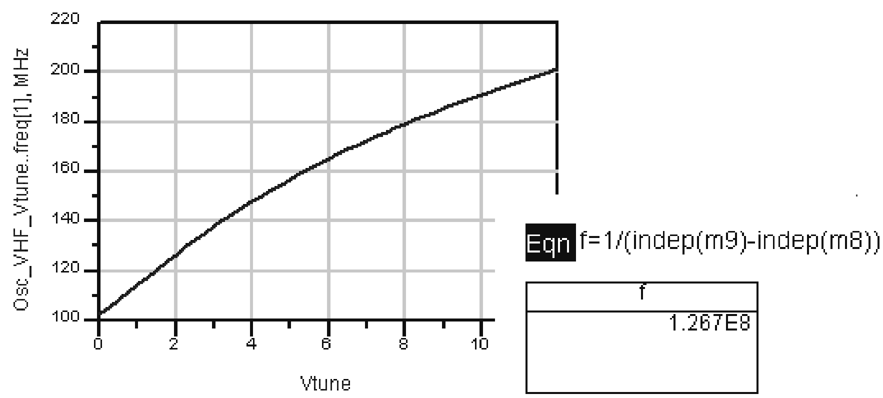




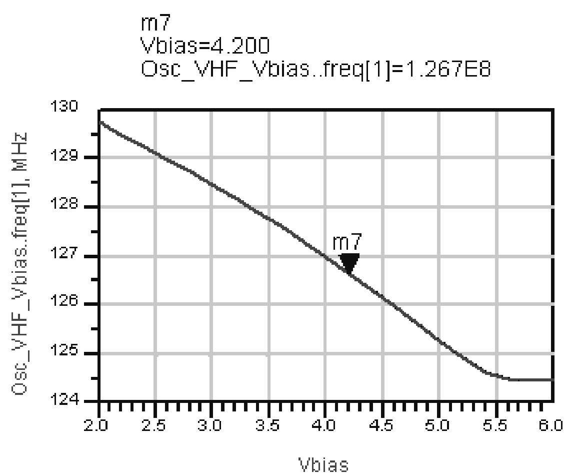
Harmonics show some distortion but it is more visible when the ts function is used.

13. Sweep voltages

a. Sweep the tuning voltage 0 to 12 volts in 0.25 volt steps to check the range and plot the results. Remember to set up a dataset name and plot freq [1].



b. Sweep the bias voltage and plot the results.



14. Oscillator Phase Noise

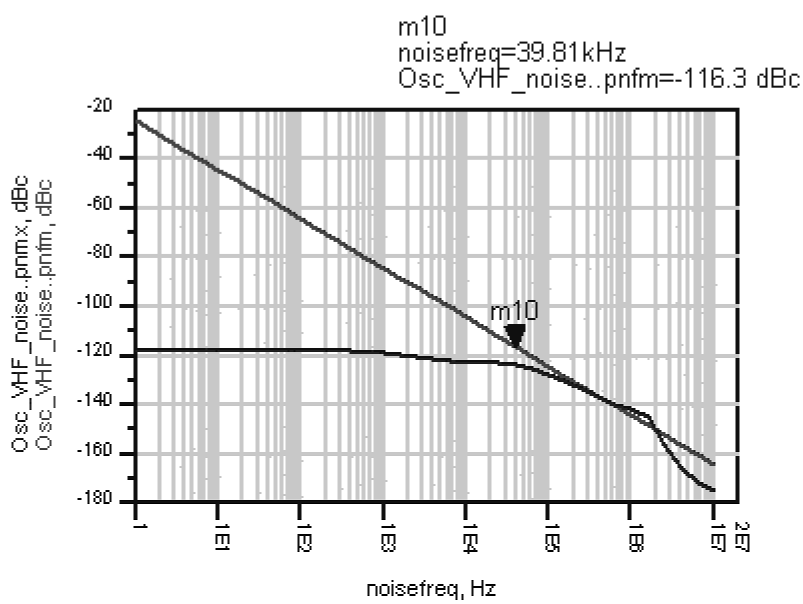
a. To do this test, put a 50 ohm resistor on the RF output. Set the HB controller as shown:

HARMONIC BALANCE

HarmonicBalance
HB_phasenoise
Freq[1]=100 MHz
Order[1]=15
StatusLevel=3
NLNoiseStart=1.0 Hz
NLNoiseStop=10.0 MHz
NLNoiseDec=5
PhaseNoise=yes
FM_Noise=yes
NoiseNode[1]="Vout"
SortNoise=Sort by value
OscMode=yes
OscPortName="oscport1"

Edit the HB controller and set the noise from 1 to 10 MHz using 5 points per decade instead of steps.

b. Simulate and watch the status window for all the results and information that OscPort provides. Then plot the phase noise results: pnfm (1/f noise) and pnmx in a log plot.



Marker shows divergence between noise data.

EXTRA EXERCISES:

1. In Part 2, try putting a 50 ohm resistor on the Vout node and note any differences with this load. Then sweep the load and look at the results.
2. In Part 2, try setting the HB controller over sampling to 3 or 4 and also set the number of harmonics to 15 and see if the oscillator harmonics or waveform improve with the simulation.
3. In Part 2, try redesigning the oscillator to have better harmonic roll off.
4. In Part 1, try measuring phase noise.

THIS PAGE IS INTENTIONALLY BLANK.