6

This chapter shows the fundamentals of using the Harmonic Balance simulator to look at the output spectrum, gain compression, and other measurements. Also, the E-Syn tool is used to build a filter for the mixer.

Lab 6: Harmonic Balance Mixer Simulations and E-Syn

OBJECTIVES

- Perform Harmonic Balance simulations
- Test Conversion Gain and Gain Compression
- Optimize values, display and manipulate data in various ways

About this lab: This lab is a continuation of the mixer design.

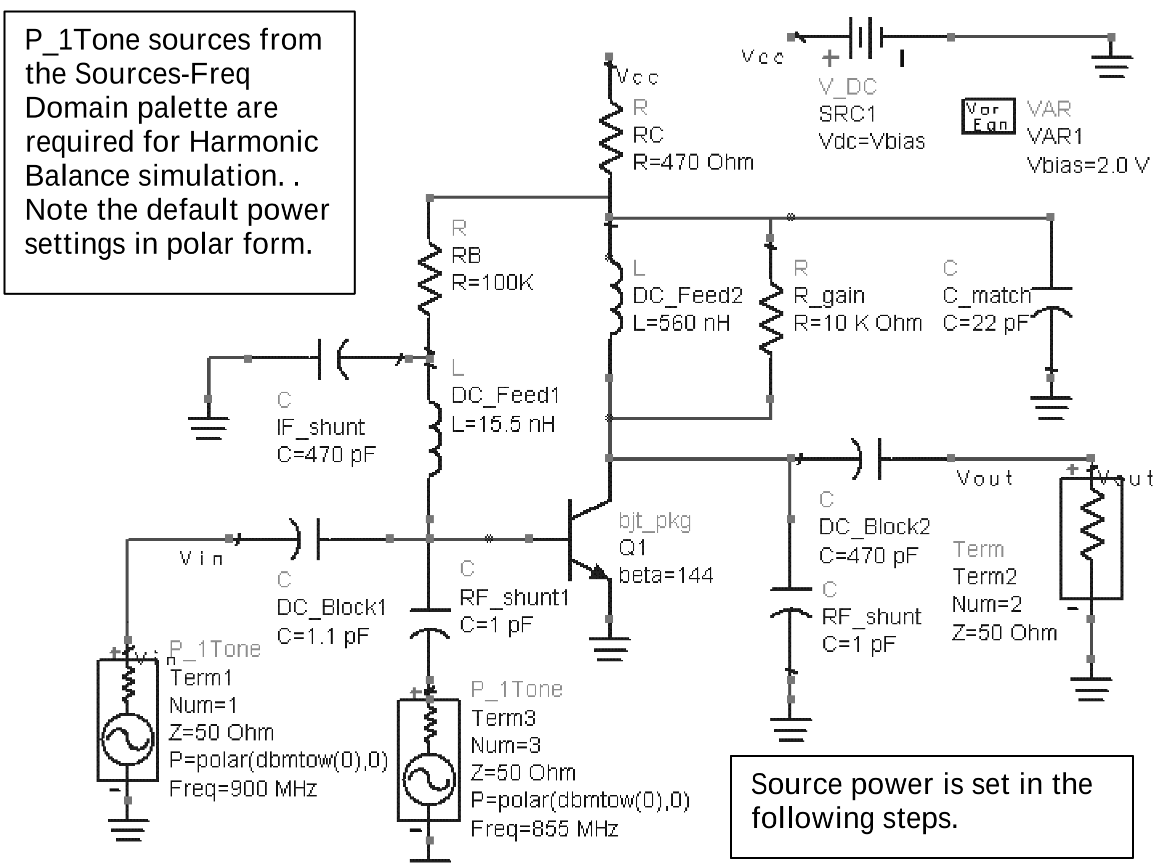
PROCEDURE

1. Create (copy or save as) a schematic design

a. Build the circuit shown here by copying the last lab and modifying it. To do this easily, save the last lab file with the new name: **hb_gain.** Be sure to delete the sources, simulations, optimizations, etc.

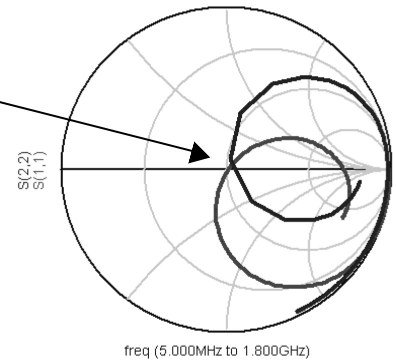
b. Insert two **P_1Tone** sources for the **RF** and **LO** and set the RF Freq = 900 MHz and the LO Freq = 855 MHz.

c. **Set DC_block 1= 1.1 pF** and **DC_Feed1 = 15.5 nH**.



Note on L and C values: The values shown in the previous schematic may be different from the ones you obtained in the last lab. However, these values should be used to keep the class consistent. The simulated S-parameters are still very good as shown here:

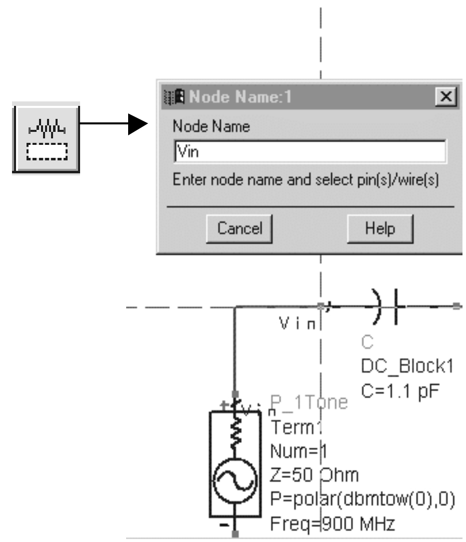
S-parameters for the hb_gain circuit: S-11 (900 MHz) and S-22 (45 MHz) intersect very close to 50 ohms and J zero for this circuit.



2. Insert node names: Vin and Vout

a. Defining a node name means that node data will be available in the data set. Also, the node name can be used as a variable or later on as the RF value. Insert the node name: **Vin**.

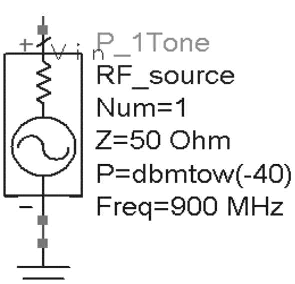
b. Insert a node name at the output: **Vout** (refer to the schematic if necessary).

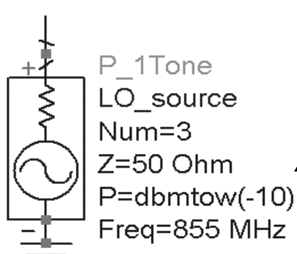


3. Set the Sources

a. Set the RF source as shown where the polar format has been simplified **P=dbmtow(-40).** Also, label the name from Port 1 to **RF_source** because the port number is already defined by Num=1 (shown here).

b. Set and label the LO source also as shown: **P= dbmtow(-10)** at 855 MHz.





SOURCE POWER FORMAT: P=polar(dbmtow(-10),0) is simplified: P=dbmtow(-10)

4. Insert a Harmonic Balance simulation controller

NOTE on setting HB setup: Freq[1] is the LO and Freq[2] is the RF. For all Harmonic Balance simulations, the source with the highest power level must be the first listed frequency: Freq[1]. Also, the Freq variables (one or more) must match the source frequencies for the mixing to be achieved. Notice that the index [ ] brackets refer to two separate frequencies and not the source port numbers (Num).

a. Set the controller freq as follows:

Freq[1]: 855 MHz and Order = 5   Freq[2]: 900MHz and Order = 4.

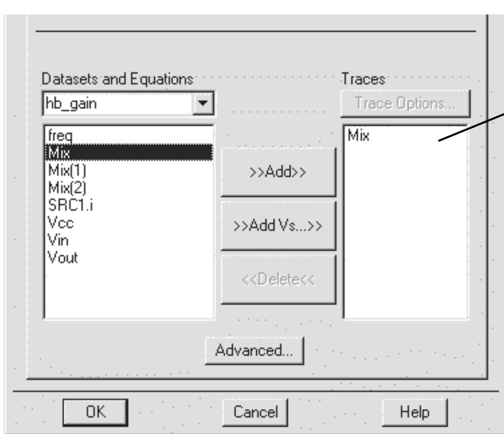
b. Set MaxOrder = 9 (max number of mixing products to be calculated)

HarmonicBalance
HB1
MaxOrder=9
Freq[1]=855 MHz
Freq[2]=900 MHz
Order[1]=5
Order[2]=4

5. Simulate and view the data

a. Use the dataset **hb_gain**. After the simulation, open a data display.

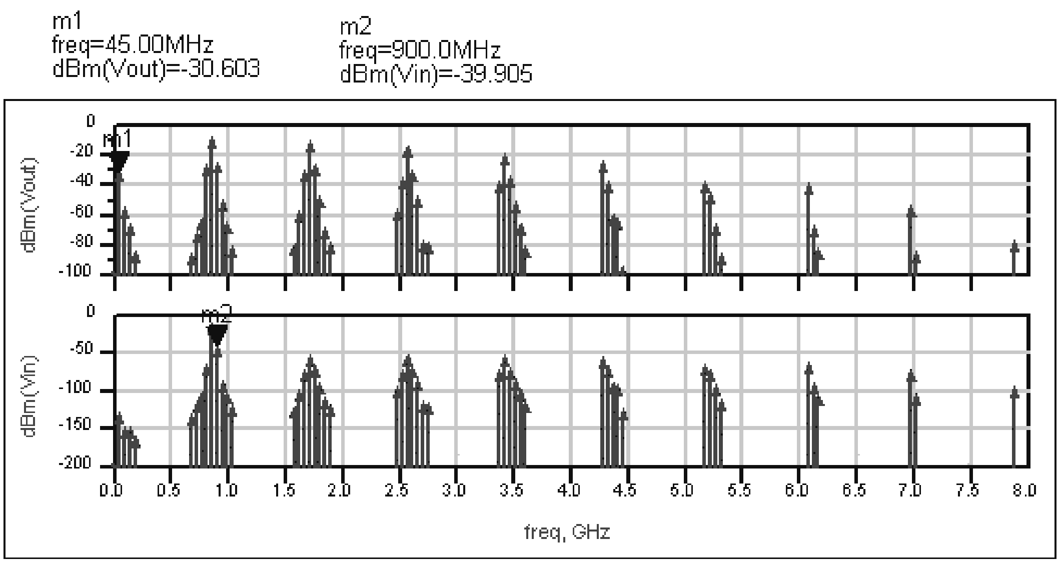
b. Insert a list of the data: **mix**. This shows you the mixing frequencies and mixing products from the simulation. Notice that the down-converted IF (45 MHz) is the result of mixing –1 times the RF and 1 times the LO. Also, Mix (1) is the LO and Mix (2) is the RF source.



freq	Mix	
	Mix(1)	Mix(2)
0.0000 Hz	0	0
45.00MHz	-1	1
90.00MHz	-2	2
135.0MHz	-3	3
180.0MHz	-4	4
675.0MHz	5	-4
720.0MHz	4	-3
765.0MHz	3	-2
810.0MHz	2	-1
855.0MHz	1	0
900.0MHz	0	1
945.0MHz	-1	2
990.0MHz	-2	3
1.035GHz	-3	4

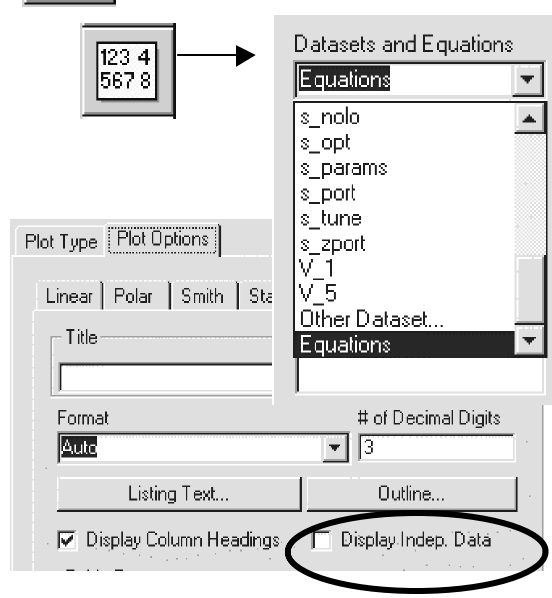
c. Insert a **stacked rectangular plot** and insert two data plots: **dBm** of **Vout** and **Vin**. These will be two separate plots in one frame.

d. In the Vout plot, put a marker on the 45 MHz IF. On the Vin plot, put a marker on the 900 MHz RF spectral tone. Your plot should look similar to the one shown here. All the mixing products (Max Order) appear along with the harmonics (Order) specified by the HB simulation controller. Also, the power is not exactly –10 dBm because the input impedance of the mixer is not exactly 50 ohms.

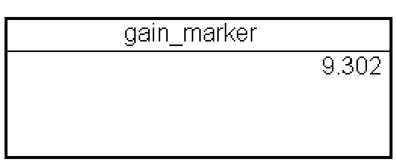


e. In the data display, insert an **equation** to calculate the conversion gain using the marker values as shown.

f. Insert a **list,** scroll down to **equations,** and add your equation: **gain_marker**. Also, use Plot Options un-check the **Display Indep Data** (freq). This is needed because the 2 markers are at different frequencies.



g. **Save the data & display**.



Design Note on gain: At this point, you have a good idea that the conversion gain is close. However, this method of using markers is not completely accurate because the dBm values returned by the marker are only valid if the impedance of the system is exactly 50 ohms. This is because the dBm function assumes 50 ohms. In the next steps, you will refine the simulation setup, learn more about using the dBm function and use a more accurate method of calculating gain.

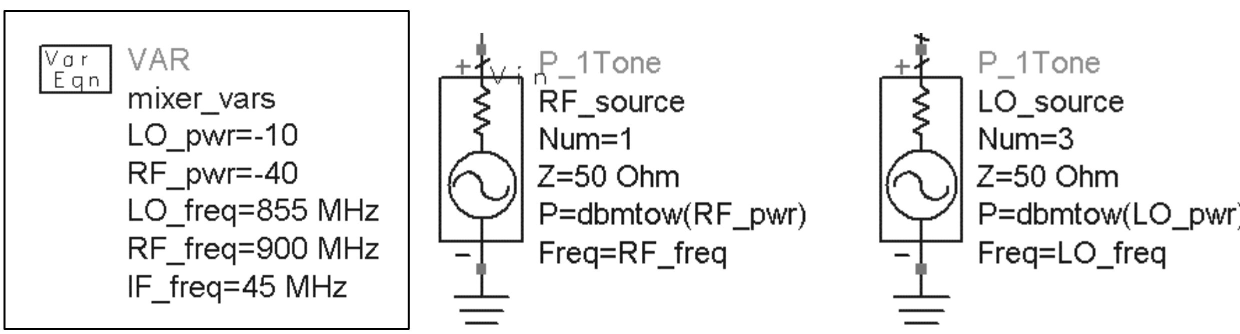
6. Use variables instead of fixed values for Freq and Power

The next few steps show how to use variable instead of “hard coded” numbers in a simulation setup. This is important for more complex circuit refinement and calculations in the remaining labs.

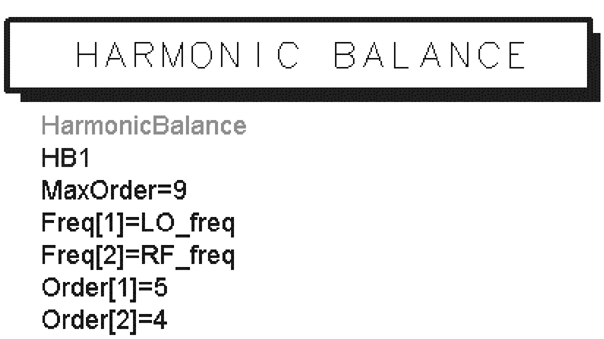
a. In the schematic, insert a variable equation from the Data Items palette or, simply type **VAR** in the component history field and press Enter.

b. Edit the VAR and assign the variables as shown for LO, RF, and IF frequency and power as shown. Assign the units (MHz) here and **do not set the units anywhere else** or they may multiply in the simulation.

c. Edit the sources and replace the values with the variables you just created for freq and power as shown:



d. Edit the Harmonic Balance controller as shown here. Notice that there is no MHz term required because it is already set in the VAR.



MaxOrder = number of mixing products calculated from Freq and Order:

Freq[1] is a variable or a number. Order [1]=5 means Freq[1] will be calculated with 5 harmonics.

7. Write two measurement equations to calculate the conversion gain.

Before simulating, you will write two measurement equations that will be used to accurately calculate conversion gain.

a. On the schematic, insert a measurement equation (you can also type **MeasEqn** in the component history field).

b. Write the first equation for **if_pwr = dBm(mix(Vout,{-1, 1}))**. This equation requires the mix function because you have a multi-toned simulation. Vout is the node voltage you want get the dBm value from and the index in curly braces refers to –1(LO) + 1(RF) which is 45 MHz:

```
Meas Eqn  MeasEqn
          conv_gain
          if_pwr=dBm(mix(Vout,{-1,1}))
          conv_gain=if_pwr-(-40)
```

c. Add a second equation (shown above) to compute the conversion gain using the if_pwr calculation: **conv_gain = if_pwr – (-40)**. Here, you are subtracting the applied RF power (-40 dBm) from the IF power.

d. **Simulate** again and when finished, go to the data display and list the equation values you just created. You should see values similar to those shown here. Notice that the value of gain is a little different than using the markers. This is because you are subtracting –40 dBm (available RF power) and not the marker value…the marker value is dBm of the standing voltage wave (result of constructive interference of forward and reflected signals). In either case, because the Z input is almost 50 ohms (little mismatch), this value of conversion gain is reasonably accurate.

freq	conv_gain	if_pwr
45.00MHz	9.397	-30.603

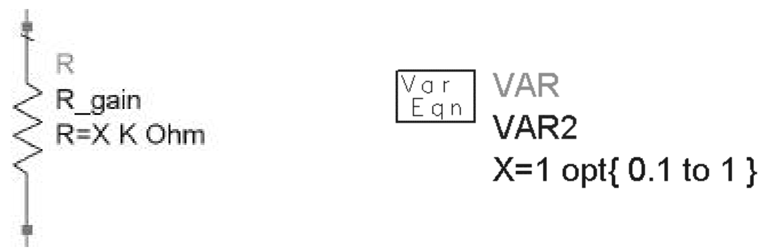
NOTE on power measurements and the dBm function: The dBm function operates on data that is assumed to be in a 50 ohm system. However, if your load is other than 50 ohms, insert a second argument such as: dBm (Vout, 75), for a 75 ohm system or even a complex value can be used. Refer to the Extra Exercises at the end of this lab for more details and for an exercise on using the pspec function to calculate gain.

8. OPTIMIZE conversion gain by adjusting the gain resistor

At this point, the conversion gain is not quite 10 dB. Therefore, the gain will be optimized by adjusting the 10K ohm gain resistor and using the existing equation *conv_gain* as the goal . However, instead of enabling the resistor, you will assign the resistor to a variable and then optimize the variable. This is the preferred method so the optimizer does not have to use scale factors such as p, M, K, etc.

a. On the schematic, change the value of the resistor (R_gain) to **X K Ohm**. Then insert a variable equation (**VAR**) and assign it as shown. You can use any variable name, X is just a suggestion.

b. Enable the variable X for optimization on-screen by typing in the value of 1, the **opt** function, and the range as shown here or use the Dialog box and the Optimization/Statistics Menu.



c. From the Optim/Stat/Yield palette, insert an optimization **goal** for the HB controller. Write the expression: **conv_gain** (your measurement equation in dB). Use **min**=**11** and **max=12** (you get better than 10 dB).

d. No entry is required for the range var "freq" because the IF is specified in the measurement equation.

e. Insert a nominal optimization controller using **Random** mode. Be sure to set the HB controller name (HB1) to the Siminstance Name. No other entries are necessary – you do not need an entry for the GoalName because if you do not specify a name, all goals will be used.

f. Simulate with a new dataset name: **hb_opt_gain**. After the simulation is finished, update the optimized value to the schematic. You should see a value of R load that is lower than 1K ohm nominal.

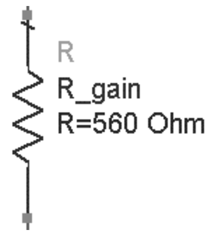
g. In the data display window, plot the results in a list as shown. Here, the optimizer set the gain resistor to about 560 ohms: (X = 0.56 * 1K VAR):

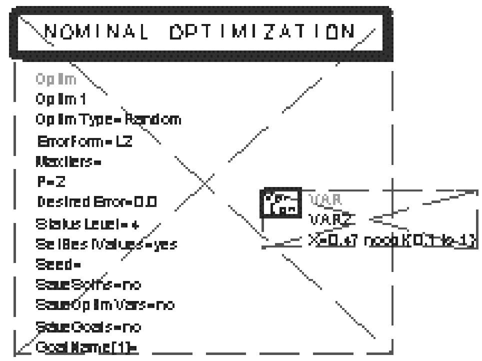
Optimized Gain (gain resistor @ 560 ohms)

freq	hb_opt_gain..conv_gain	hb_opt_gain..if_pwr
45.00MHz	11.128	-28.872

9. Fix the gain resistor value, deactivate the Optimization, Simulate

a. Change the gain resistor to 560 ohms even if you had a different value from the optimization.





b. Deactivate the optimization controller and the VAR used for the gain resistor or simply delete them from the schematic. They will not be used again.

c. **Simulate again** and examine the data. Afterward, **save the data display** and **close the data display window**. IMPORTANT – you will use this data, hb_opt_gain..dBm(Vout), for comparison in the Transient lab.

Mixer Design Note: At this point, the mixer has achieved the specifications of the dc supply budget and the conversion gain. The next part of this lab will be to test for the 1 dB compression point using two different methods available in ADS.

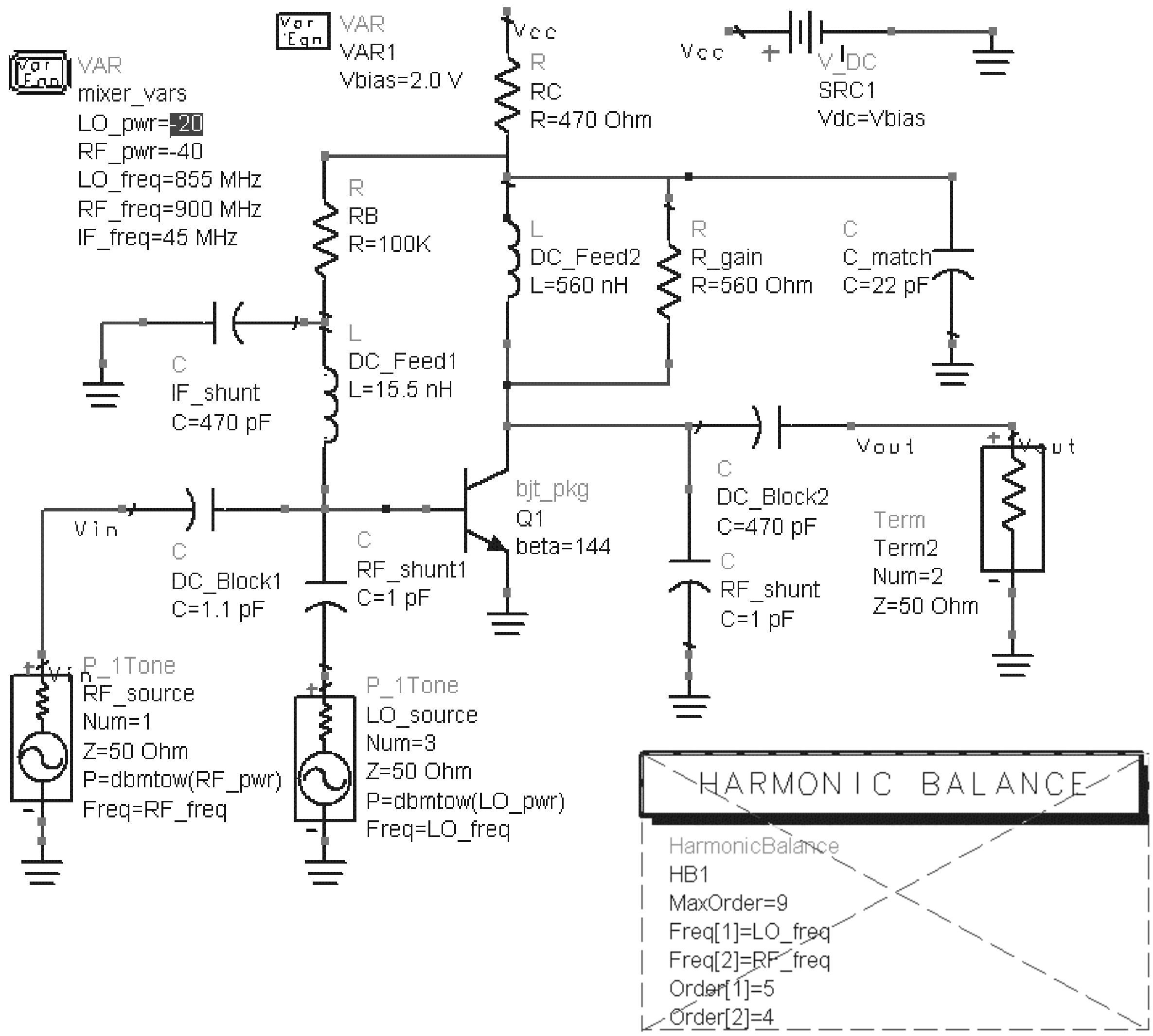
10. Set up an XDB gain compression test

The XDB . The easiest way to get a value for gain compression is to use the XDB simulation controller that is a form of Harmonic Balance.

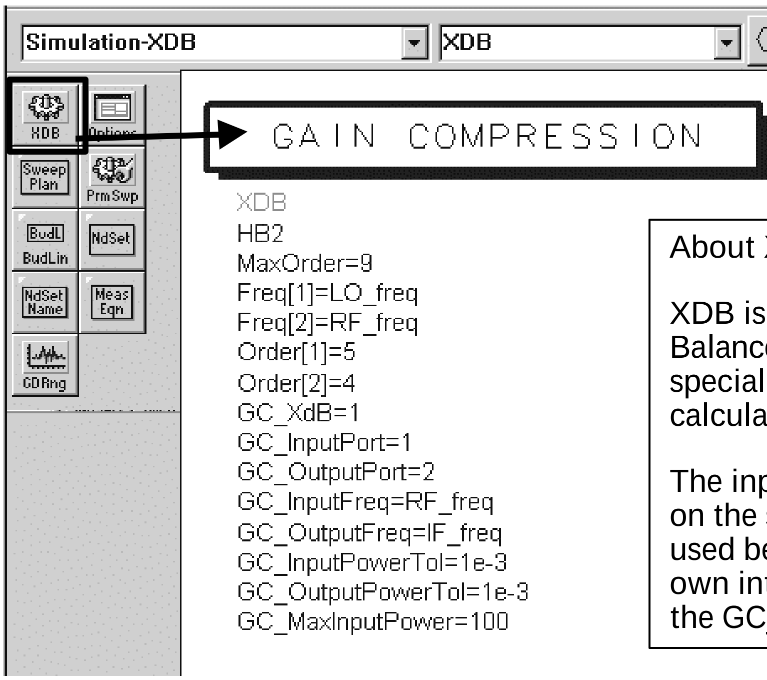
a. In the existing schematic design (hb_gain) use the **Save As** command and save the design as **hb_comp**.

b. Deactivate the HB controller and delete all other components so that the circuit looks like the one shown here where only the circuit elements, variables, and deactivated HB controller remain.

c. Set LO_pwr to –20 because the BJT will operate better for this lab.



d. Insert the **Gain Compression** controller, **XDB**, on the schematic and set it up as shown. Edit the controller to display the MaxOrder or any other settings desired.



About XDB:

XDB is basically a Harmonic Balance controller that is specially designed to calculate gain compression.

The input power (RF_pwr) on the source or var is not used because XDB uses its own internal power sweep at the GC_InputPort=1.

NOTE: GC_XdB is set to **1 dB** compression by default but can be changed, GC_Input and GC_Output must match the port numbers (num=1 and num=2). In this lab, num=3 is the LO source.

e. Insert a measurement equation for the IF output power - you can use the if_pwr equation from the hb_gain schematic using the Edit > Copy / Paste commands or write a new one.

MeasEqn
meas1
IF_pwr= dBm (mix(Vout,{-1,1}))

NOTE on syntax for the mix function: The mix function is used with HB or XDB simulations where mixing occurs (max order > 2). To access data using the mix function, you supply two arguments (in parenthesis) separated by a comma. The first argument here (Vout) is a voltage or node. The second argument {in curly braces} describes the mixing product. Here, -1 means –1 times the LO freq mixed with +1 times the RF freq which equals 45 MHz. For the mix function in ADS, the use of curly braces generates a matrix which contains the index into the simulation data.

11. Simulate XDB (gain compression) and display the results

a. Simulate the XDB measurement with a new dataset name: **hb_xdb.**

b. In the data display, insert a list with **inpwr** and **outpwr** values - these are automatically generated in the dataset by the XDB simulation.

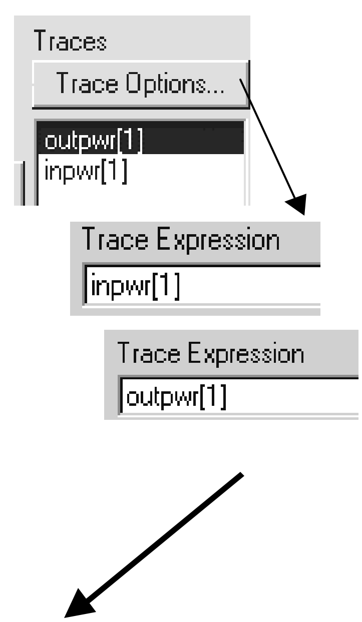
Looking at the list and you see that the independent variable **freq** is listed along with inpwr and outpwr. However, you specified the exact input and output frequencies in the controller (RF_freq and IF_freq). It is true that harmonics were used to accurately achieve the solution but their power levels are not part of the dataset. Therefore, you need to display only one value of this inpwr and outpwr on your list in the data display. In the next step you will do this - it is called indexing into the data.

freq	inpwr	outpwr	
0.0000 Hz	-31.25 dBm	-21.27 dBm	0
45.00MHz	-31.25 dBm	-21.27 dBm	1
90.00MHz	-31.25 dBm	-21.27 dBm	2
135.0MHz	-31.25 dBm	-21.27 dBm	3
180.0MHz	-31.25 dBm	-21.27 dBm	4
675.0MHz	-31.25 dBm	-21.27 dBm	etc.
720.0MHz	-31.25 dBm	-21.27 dBm	
765.0MHz	-31.25 dBm	-21.27 dBm	
810.0MHz	-31.25 dBm	-21.27 dBm	
855.0MHz	-31.25 dBm	-21.27 dBm	
900.0MHz	-31.25 dBm	-21.27 dBm	
945.0MHz	-31.25 dBm	-21.27 dBm	

Index numbers of each piece of data in the list.

The next step shows how to index into the data and display only one value of inpwr or outpwr.

c. **Index the data:** Using XDB, the power sweep occurred internally in the simulator to determine inpwr and outpwr for the compression point at the RF and IF frequencies you specified. Therefore, use brackets, called sweep indexers in ADS, to display only one value each of inpwr and outpwr. To do this, **edit the plot**, **select the trace**, and then use **Trace Options** to append brackets and an index number to the trace expression (data). When the dialog appears, add the index brackets with a number such as [1] or [0] or [2] etc. to the trace expression as shown here and your XDB results will read:



inpwr[1] and **outpwr [1]**

XDB simulation results: 1 dB compression

inpwr[1]	outpwr[1]
-31.251	-21.268

12. Setup a Harmonic Balance compression test

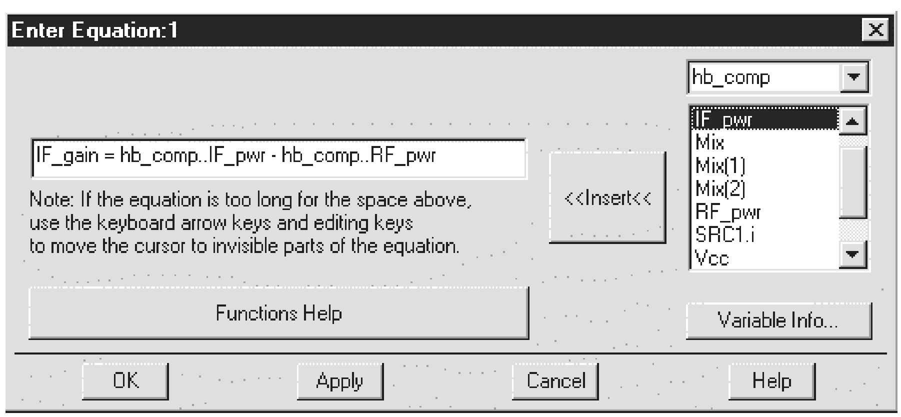
In this next simulation, you will set up the power sweep and determine the 1 dB compression point by plotting the output power against the input power. To do this, you will sweep the variable RF_pwr –50 to –20 dBm.

a. In the current schematic, **deactivate the XDB** controller and set up a HB controller as shown. Remember to edit the controller and click the Display tab to show SweepVar and the Start, Stop, Step. You already have the variables defined in the VarEqn.

HarmonicBalance
HB1
MaxOrder=9
Freq[1]=LO_freq
Freq[2]=RF_freq
Order[1]=5
Order[2]=4
SweepVar="RF_pwr"
Start=-50
Stop=-20
Step=1

b. Simulate with a new dataset name **hb_comp**.

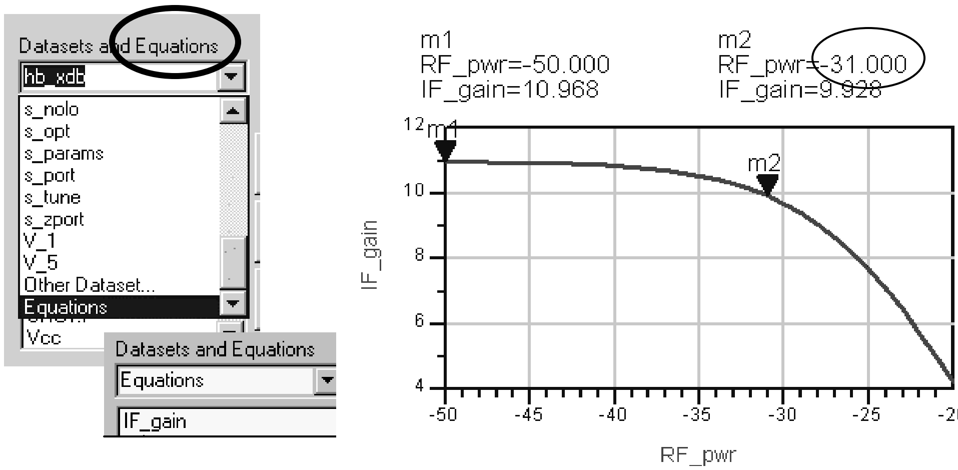
c. In the data display, insert a new IF_gain equation = IF_pwr – RF_pwr. When the dialog appears, select the hb_comp dataset and enter the equation using the insert button:



You should get the following valid (black) equation ready to plot:

Eqn IF_gain = hb_comp..IF_pwr - hb_comp..RF_pwr

d. Insert a plot and when the dialog appears, scroll down to the equation and put two markers 1 dB apart as shown here. You should see that the RF_pwr is about –31 dBm when the circuit goes into 1 dB of compression – similar to the result from the XDB simulation.



e. Create one more plot with IF_pwr vs RF_pwr as shown here using the **hb_comp** data.

f. Create an equation that will be a line. The line extrapolates the linear value of IF power as if there was no compression.
Insert an equation:

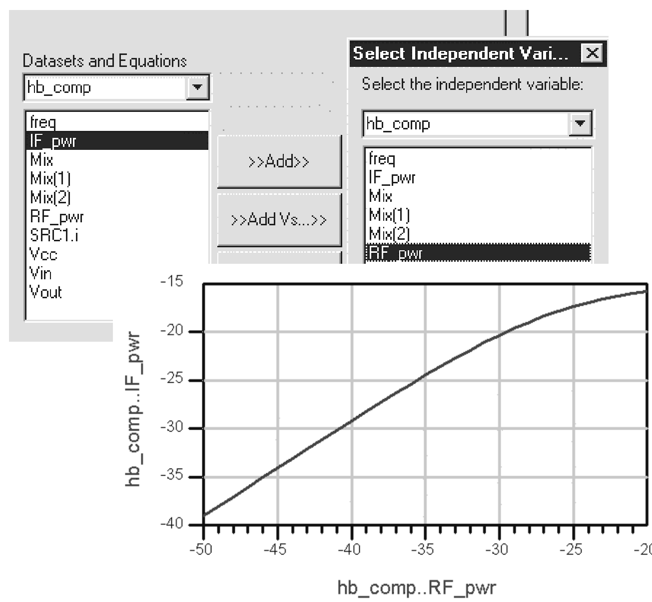
Eqn line = hb_comp..RF_pwr + IF_gain [0]

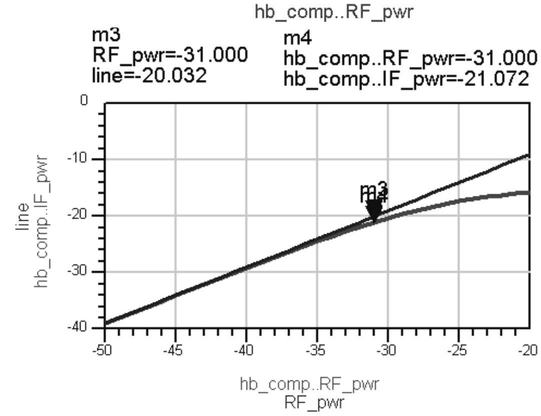
where [0] is the lowest power level.

g. Add the line equation to your existing plot. Then put markers on the two traces when the difference between the line and the IF_pwr is 1 dB.

Notice that RF_pwr is consistent with the previous results. Using XDB, you specify the dB value.

h. Save the schematic design and data display (hb_comp).





13. Use E-syn to build a low-pass Filter

The mixer IF has LO and RF feedthrough as you have seen in the spectrum. Therefore, one easy way to build a filter is to use the ADS E-syn tool.

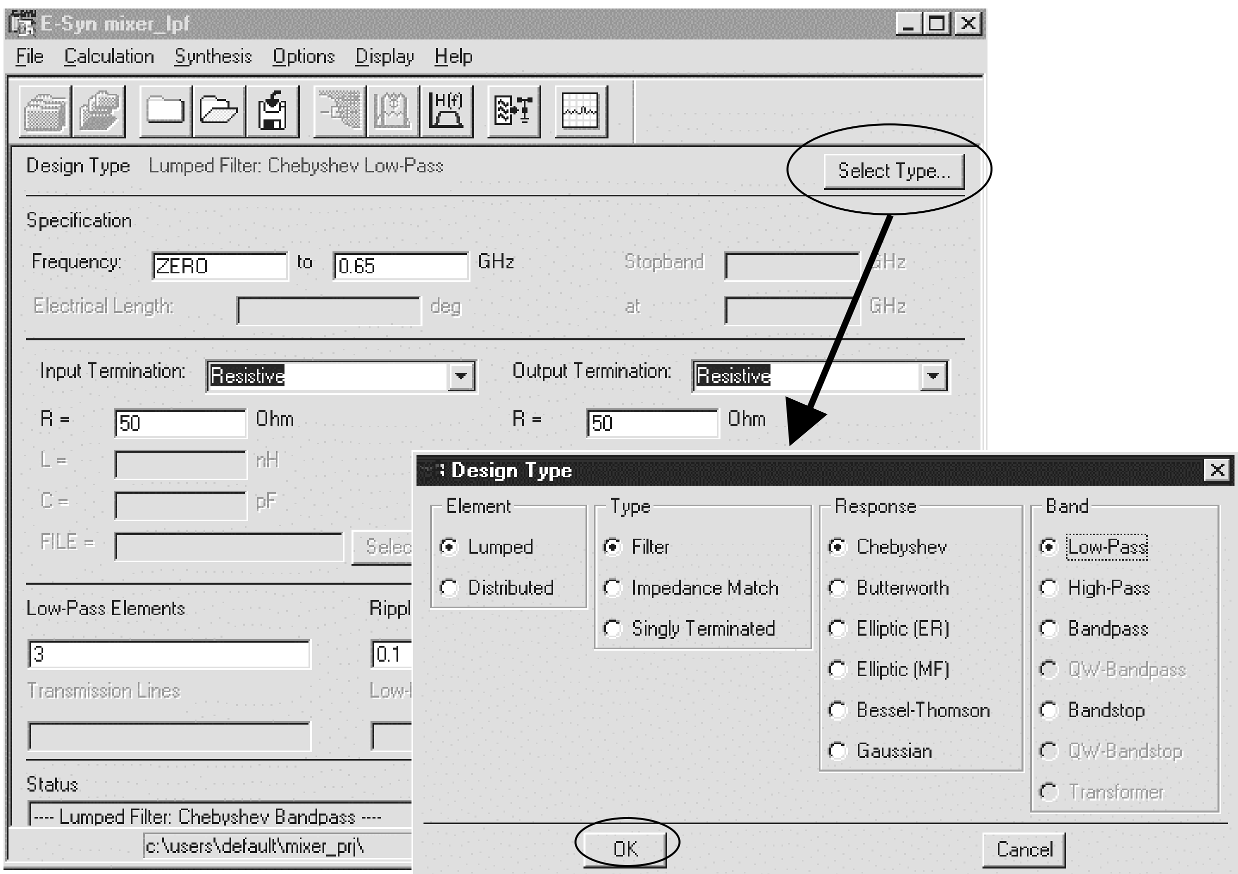
a. Use the Save As command to save a new schematic as: **hb_esyn**.

b. In the schematic, click: **Tools > E-syn > Start E-syn**

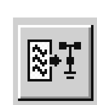
a. When the E_syn window first appears, click File > Save As and give the E-syn file a name: **mixer_lpf**. This will make it easy to keep track of the filter and the data.

b. Click the **Select Type** button. Then select a **Lumped, Filter, Chebychev, Low-Pass** and click OK.

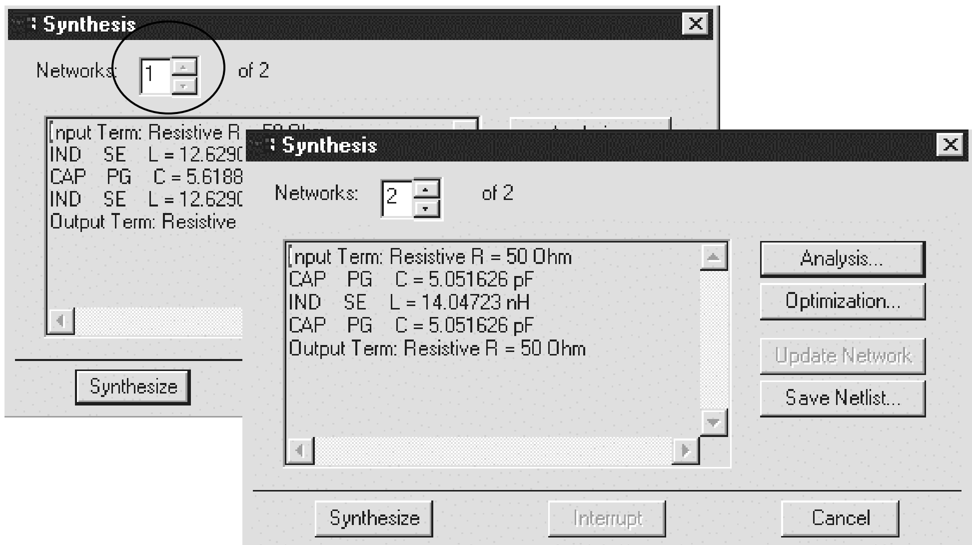
c. In the E-syn main window, set the Frequency spec: ZERO to 0.65 GHz.



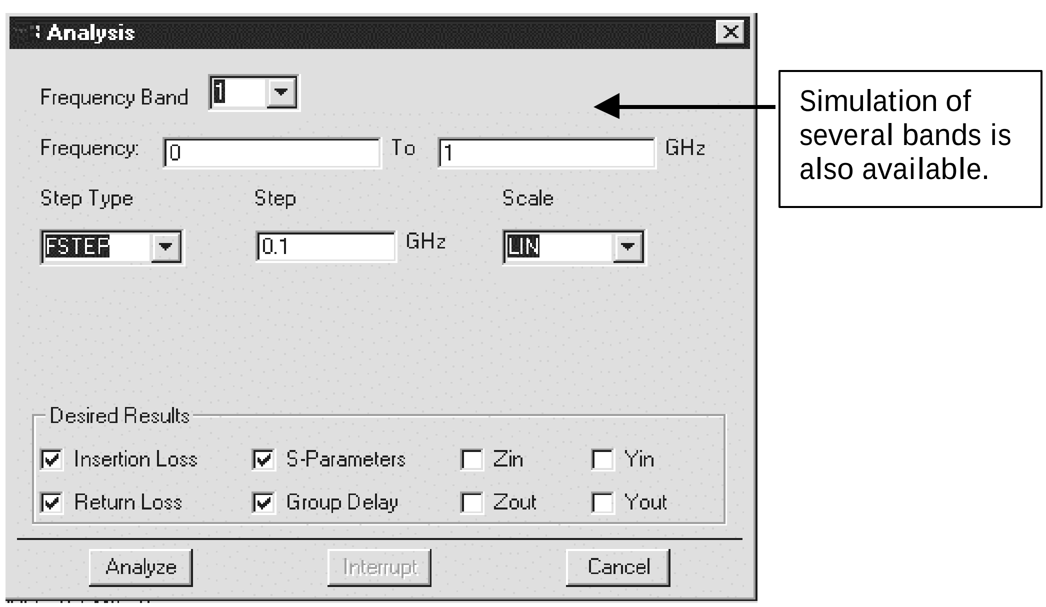
d. Then start the E-syn synthesizer by clicking the menu command Synthesis… or click the synthesis icon as shown here.



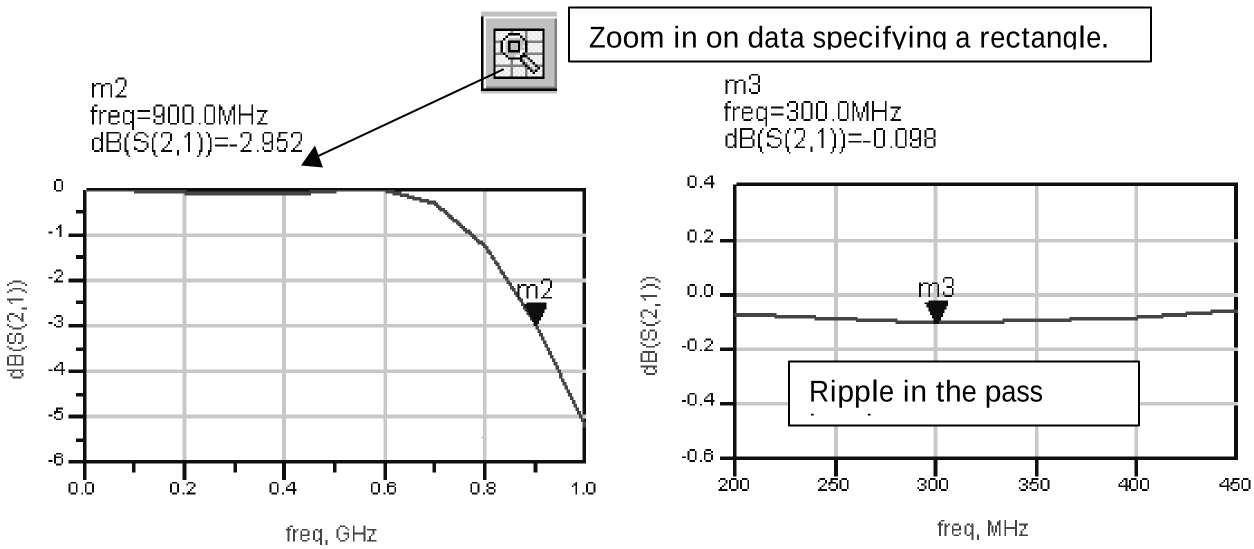
e. When the dialog appears, click the Synthesize button and the filter(s) will be synthesized. In the dialog box, you can see that 2 filters have been created and 1 of 2 is shown as LCL filter. Go ahead and click to see the 2^{nd} filter which is a CLC filter. You will use the **CLC** topology.



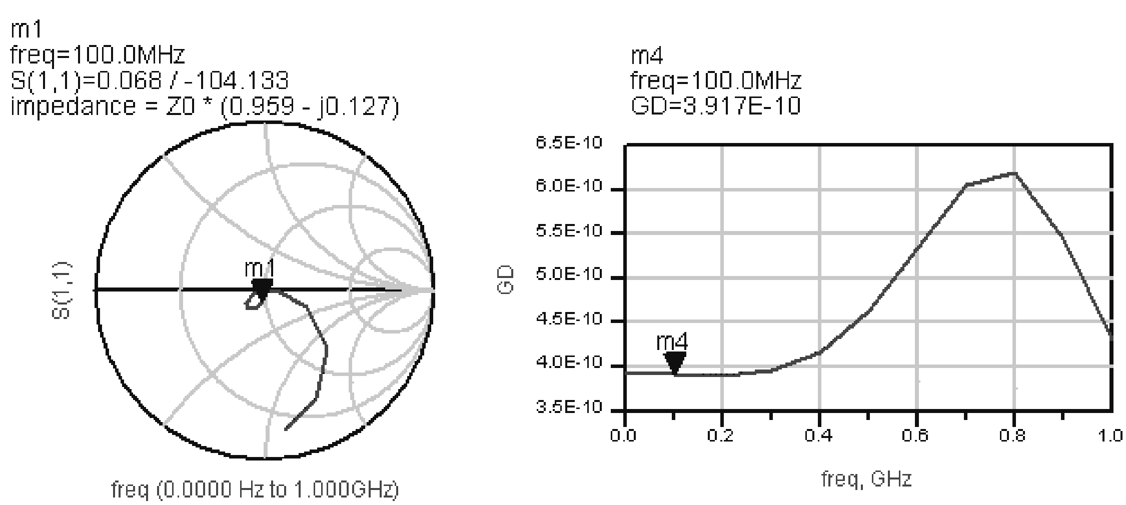
f. In the Synthesis dialog (shown above), click the **Analysis** button and a dialog will appear. Set up the analysis (simulation) for this low-pass IF filter as shown. Use only **1 band**, from **0 to 1 GHz** and a frequency step, FSTEP = **0.1** GHz which is 100 MHz. Also, click S-parameters and Group delay boxes to get this data. Then click the **Analyze** button and the simulation will run. Look in the E-syn main window status area to see if the analysis is complete.



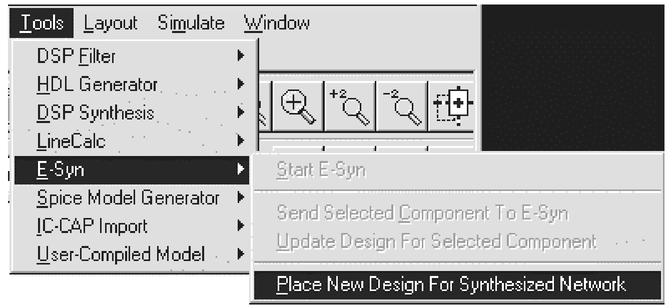
g. If the analysis is complete, click the data display icon in the E-syn main window. The data display will open and the default dataset should be set to the E-syn file name you saved earlier: **mixer_lpf**. Plot S-21 in one plot. Then copy that plot using **Ctrl C Ctrl V** to get a second plot. Then use the **zoom** (rectangle) icon to show a zoomed in portion of the pass band ripple, It should be less than 0.1 dB as shown.



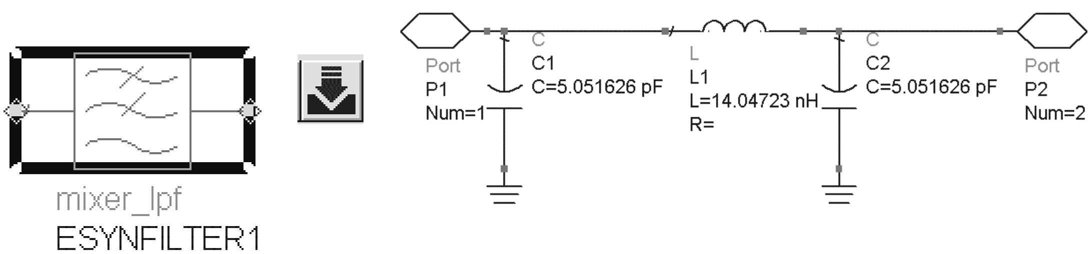
h. Plot S-11 in a Smith Chart and plot GD (group delay) in a rectangular plot to verify that the pass-band S11 is near 50 ohms and that the group delay is flat in the pass band. For greater resolution, you could do another analysis using 100 MHz or 10 MHz steps, etc. But this is good enough for the purposes of this lab exercise.



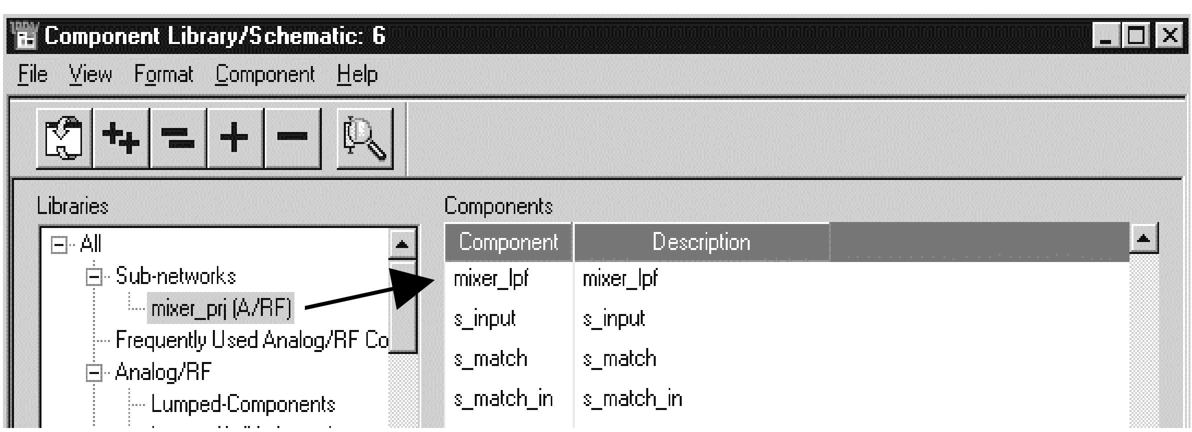
i. Now that the filter is reasonable, it is time to make it a useable component (sub-circuit). To do this, go back to the schematic window (hb_esyn) and click**: Tools > E-syn > Place New Design For Synthesized Network** – this menu command only appears after synthesis. A dialog box will appear for you to name the filter. Type in name and click OK.



j. The E-syn component will be automatically attached to your cursor – place on the schematic in an open area, select it, and push into it to see the lumped element sub-circuit. Afterward, push out and back to the schematic.



k. In the schematic window, click the library icon and verify that the filter is a sub-circuit (Sub-network) that is also available here available.



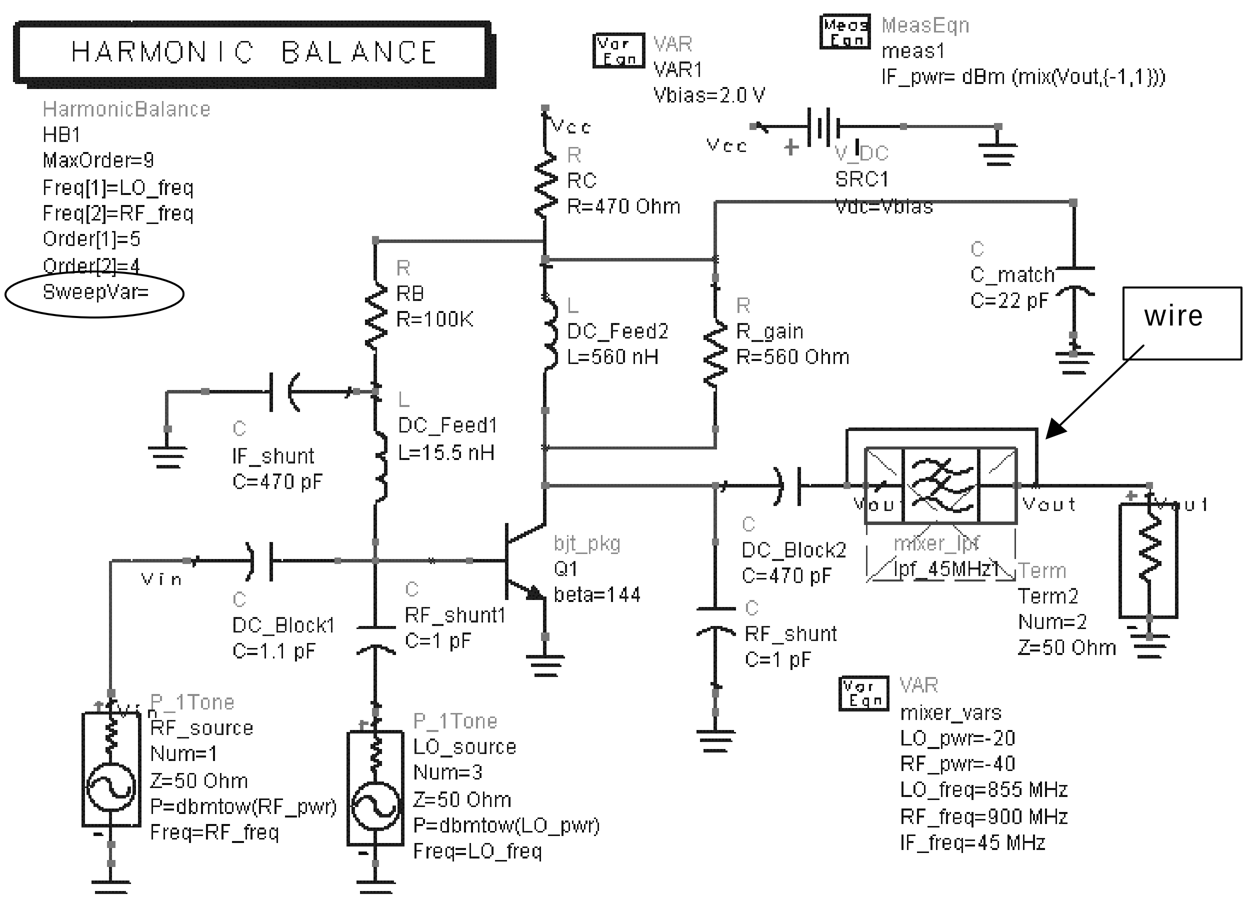
14. Perform a HB simulation with the filter connected to Vout

a. Connect the filter to the output of the mixer as shown. The mixer_lpf shown here has been assigned a symbol and name using the File > Design Parameters menu similar to the bjt_pkg you did in lab 2.

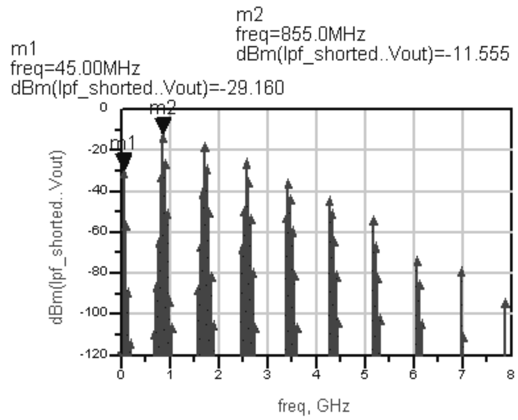
b. Deactivate the filter and connect a wire around it. After the first simulation, you will remove the wire and activate the filter.

c. Remove the Sweep Variable (SweepVar) from the HB controller.

Design Note: Effects of the filter on the mixer - The filter now affects the output impedance of the mixer by presenting a different termination to higher order frequencies. In turn, this may have a slight second order effect on the gain of the mixer, approximately 1dB.



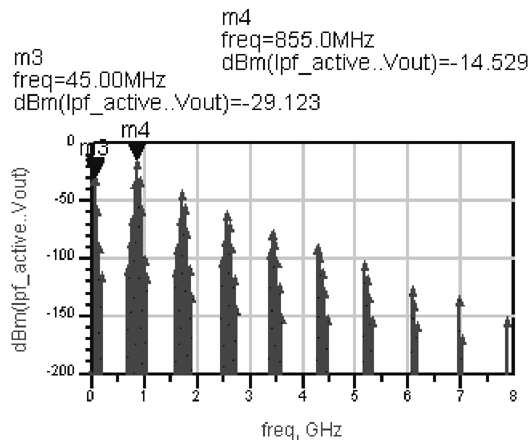
15. Simulate with the filter shorted



a. Simulate with the dataset name: **lpf_shorted**. After the simulation, plot Vout. Put a marker on the IF and the LO.

b. Position the data display so that you can see it along with the schematic. The LO should be about 11 or 12 dBm. This plot will be used as the reference.

16. Simulate with the filter active



a. Activate the filter and delete the wire.

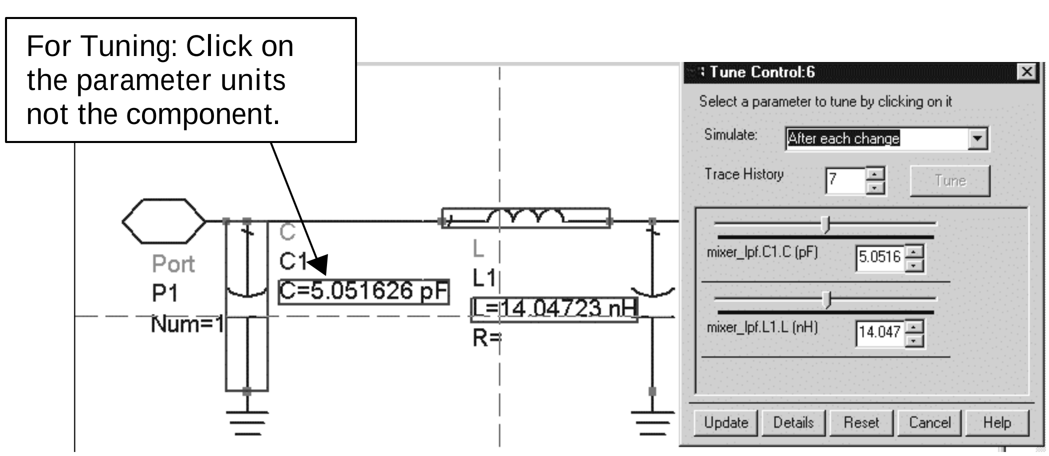
b. Simulate with the dataset name: **lpf_active**.

c. Plot the response and compare. The LO should be about 3dB lower with the filter. But this can be improved.

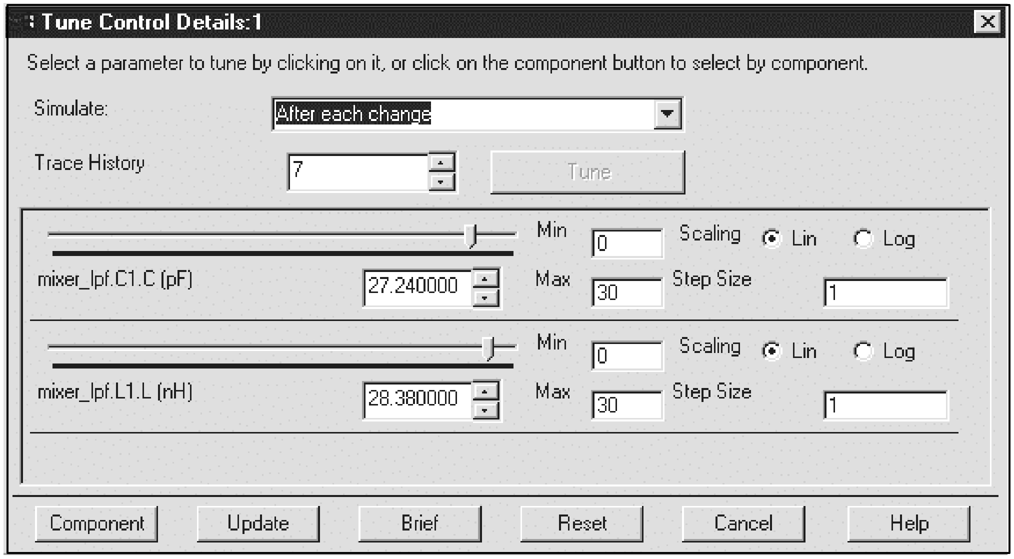
17. Tune the lpf

a. In the schematic, select the filter (click on it). Then start the tune mode. You must start the tuning feature from the schematic where the simulation has been set up.

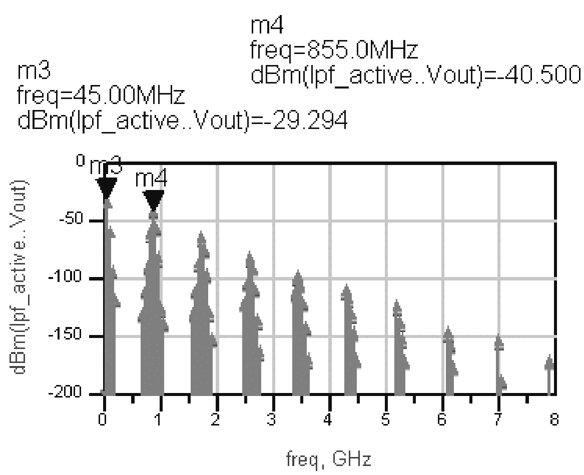
b. After the Tune Control dialog appears, push into the filter subcircuit and select (click on) the C1 and L1 parameters as shown here. Then they will be written into the Tune Control dialog.



c. Position the data display and the Tune Control so you can see them. Also, move the schematic window aside or below but do not minimize it or close it. In the Tune Control, use the **Details** button to get more range and set the step size. Then tune the filter to lower the LO signal.



Tuned lowpass filter moves the LO down 10dB below the IF. Note that the IF remains very close to its previous level. This is a big improvement .



18. After experimenting, close the data display and schematic. These will not be used for the next lab.

EXTRA EXERCISES:

1. Try writing an equation for conversion gain if the system is not 50 ohms. For example, if you are driving a high impedance you would use the following measurement equation syntax for the dBm function where the load is 1K ohms + j200 ohms:

```
MeasEqn
conv_gain
if_pwr=dBm(mix(Vout,{-1,1}),1K+j*200 )
conv_gain=if_pwr-(-40)
```

> NOTE: Your matching network also needs to be adjusted for the load.

2. Use the **pspec** function to calculate power gain to the load. To do this, first look at the Help for pspec. Then insert a current probe at the Vout node of the mixer. Simulate and then write the following equations in the data display and list the values:

```
Eqn if_pwr_watts=pspec(mix(Vout,{-1,1}),0,mix(I_Probe1.i,{-1,1}))
Eqn if_pwr_dbm=10*log(if_pwr_watts)+30
Eqn pspec_gain=if_pwr_dbm-(-40)
```

freq	if_pwr_watts	if_pwr_dbm	pspec_gain
45.00MHz	8.704E-7	-30.603	9.397

Above, if_pwr_watts uses the pspec function to calculate output power in watts (high voltage, low voltage, current at 45 MHz). The if_pwr_dbm equation gives the value in dBm and pspec_gain is the accurate conversion gain. This way of calculating is very accurate for any load impedance.

3. Sweep the LO power +/- 10 dbm (around -10 dbm level) and see if the circuit still meets the conversion gain specification.

4. Set up a temperature sweep of the circuit. To do this, sweep the device temperature parameter.

5. Determine the amount of IF leakage at the input and the amount of RF or LO leakage at the output.