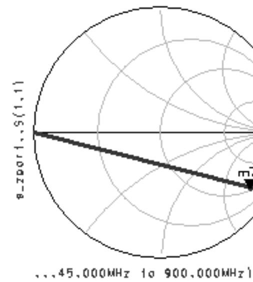
5

This chapter shows various ways of creating matching networks by sweeping values and using optimization.

Lab 5: Matching & Optimization

OBJECTIVES

- Create an input match to the RF and an output match to the IF
- Tune and Optimize to achieve matching goals



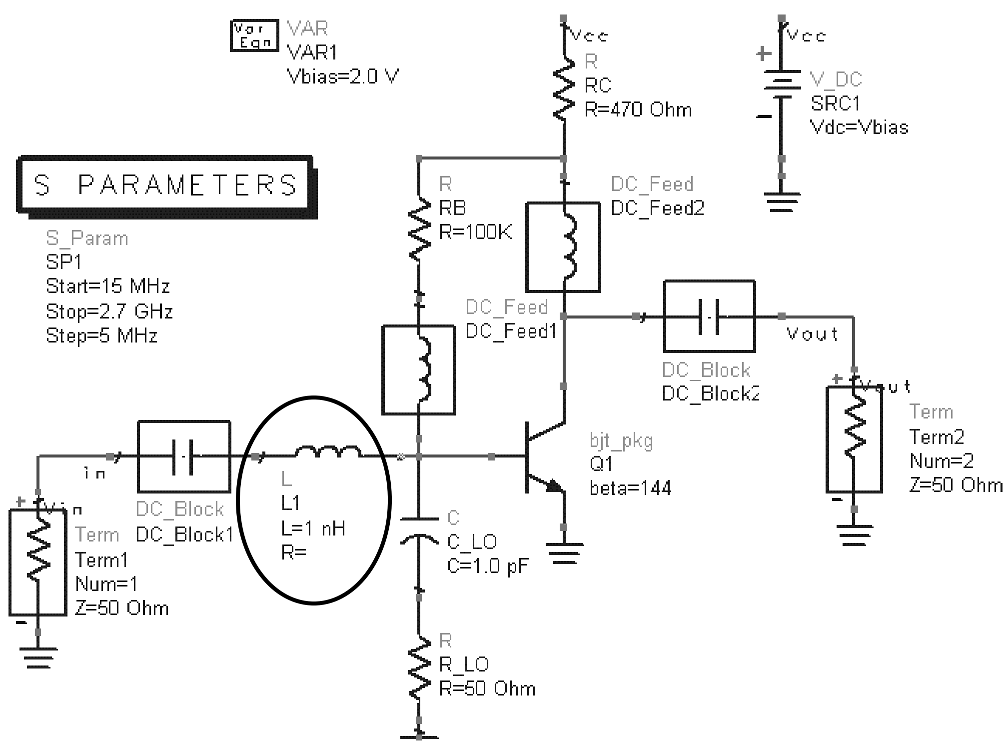
Mixer Design Note: From the Smith Chart S-11 results in the last lab, it appears that a series inductor can be added to the input as a first step in moving toward the center of the Smith chart for the RF match at 900 MHz. However, this does not take into consideration the other L and C components. But as a first step, it is reasonable to add the series inductor and see the effects of tuning as ideal components are replaced with real values.

PROCEDURE

1. Create a new schematic design for the input match.

a. Use the s_params design (last lab) and save it as: **s_match**.

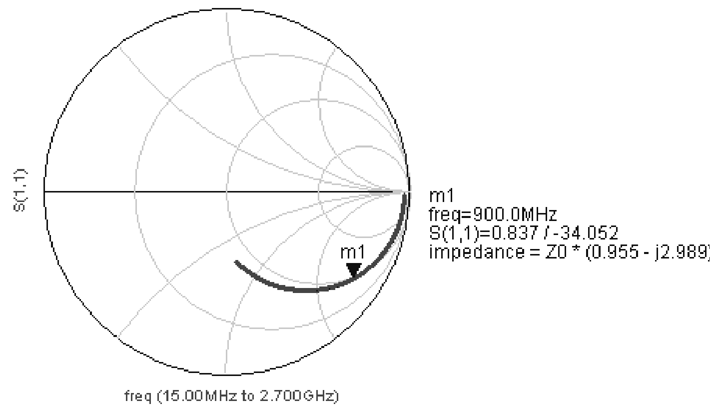
b. Insert an **inductor L in series** to the input, as shown. Your circuit should look like the one here where the Sweep Plan and Z-ports are removed and set the S-parameter controller to sweep 15 MHz to 2.7 GHz – this will simulate most of the frequencies that will result when the LO is added.



c. Check the sub-circuit to be sure there is no capacitor across the base-collector (from the last lab).

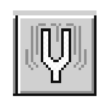
d. **Simulate** and **display S-11** in a new data display window. Position the dds window next to the schematic so you can see both at the same time. The default dataset should be the same name as the schematic: **s_match**. The results of the swept analysis should look like the plot here where a marker is added to show the value of S-11 at 900 MHz:

Use the keyboard arrow keys and the mouse to position the marker



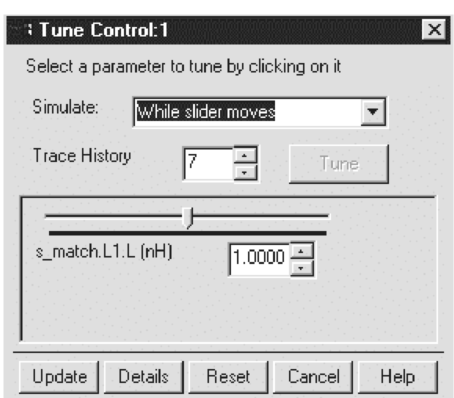
2. Start tuning the inductor

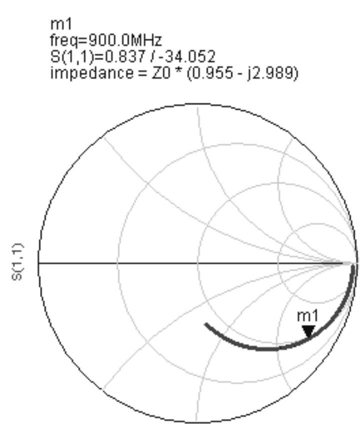
a. Select the **inductor** and **start the tuning mode**.



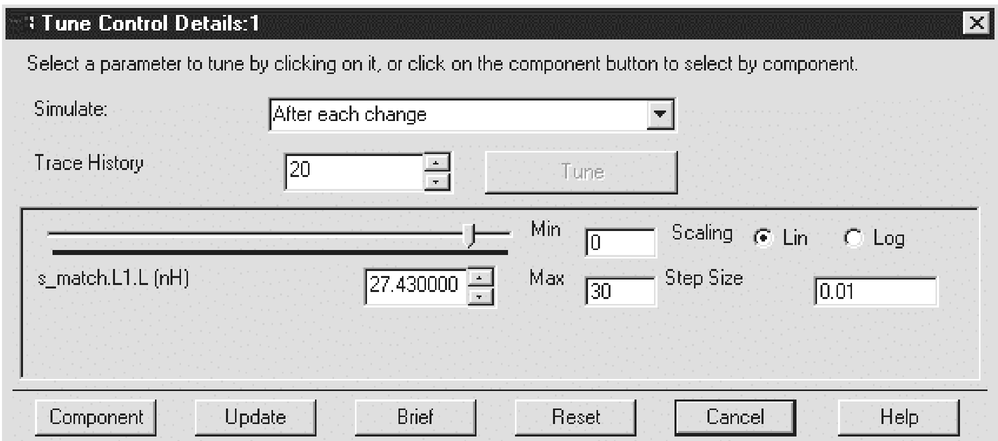
b. After the tuning dialog and status appear, open and position a **new data display window** near the tune control so you can see them both – move the schematic aside if necessary. Notice that the default dataset name **s_match** will appear (same as the schematic). Insert a Smith chart with S11 data and put a marker at 900 MHz. Notice that the S-11 trace is now changed with the real values of C and L.

c. Now, set the tune control to **slider** mode and move the slider back and forth between the ends. Notice that the value of S-11 changes very little because the range of inductance is too narrow.

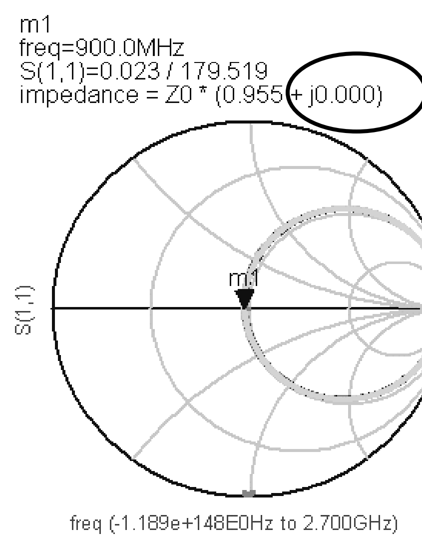




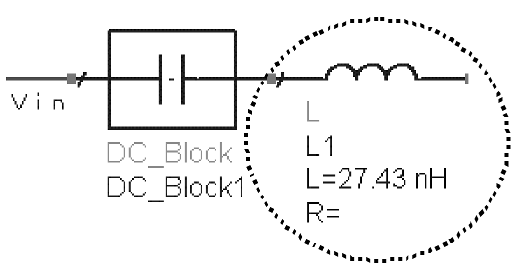
d. Increase the tuning range: click the **Details** button and the more detailed tune control appears. Increase the range from 0 to **30** by typing over the existing value. Based on the imaginary part of the impedance (-j3.1), the conjugate value of inductance of **30 nH** is close enough. Also, set the resolution Step Size to step to something small such as **0.1** or **0.01** and increase Trace History to **20**.



e. You should now be able to carefully move the slider and click the step buttons until you reach the impedance of **j0.000** as shown by the marker on the last trace. You can use this technique for determining the sensitivity of any component.



f. Click the **Update** button on the tune control and the value of L will appear on the component:



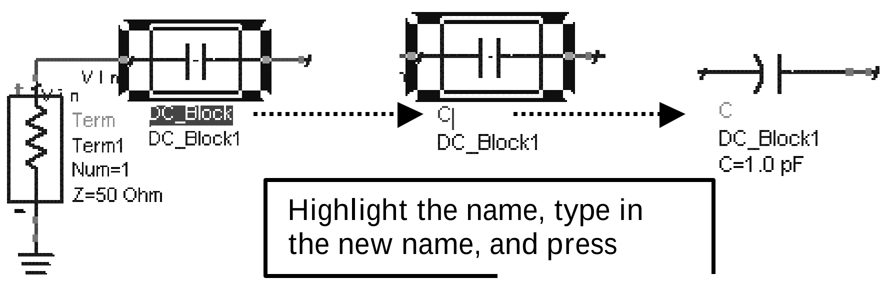
g. Save the data display as **s_match**.

3. Build a new input matching network (new configuration)

CIRCUIT DESIGN NOTE: At this point, the addition of the series inductor is only a first approximation. The remaining ideal components (DC feeds and blocks) must be replaced by realistic values and this may require a completely different topology other than just adding a series inductance. Also, a shunt capacitor needs to be added to the input to remove the IF signal that may appear there. Therefore, instead of continuing to add components in an attempt to create a match, you will use the following configuration that will solve all the matching problems for the input. This will speed up the lab exercise.

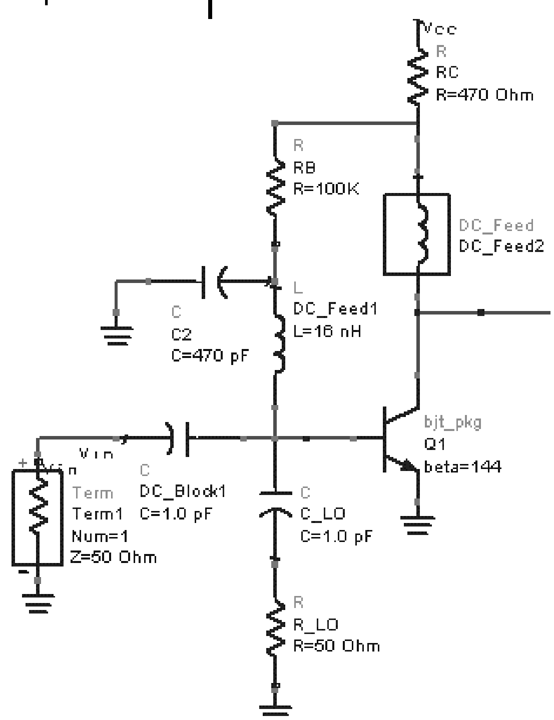
a. On the input, remove the series inductor you just tuned. It will be replaced by a network which will achieve the desired RF match and also provide the filtering.

b. Change the DC_Blocker to a real capacitor by **highlighting the component name** (see drawing - DC_Block) and typing in the new component name **C** and pressing Enter on the keyboard. The DC Block will automatically become a lumped capacitor:

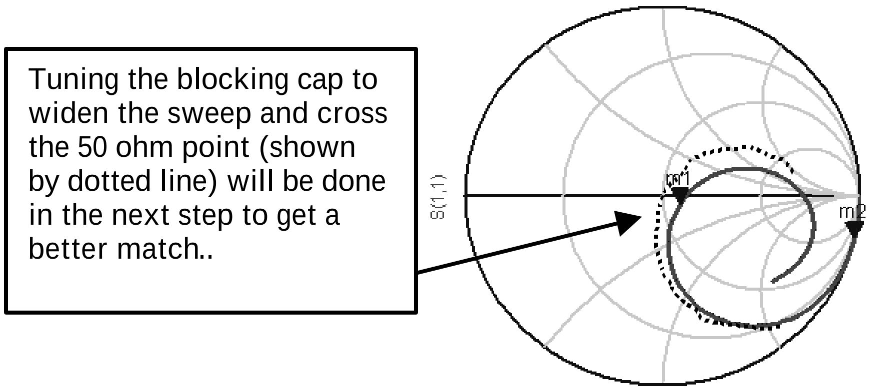


c. Continue modifying the input topology: Insert **C=470 pF** to shunt the IF (470 pF is a short to 45 MHz). Also, change the DC_Feed1 to **L=16 nH** to allow the dc to flow but it will block (choke) the RF. Lastly, be sure the **Z-ports** have been removed**.**

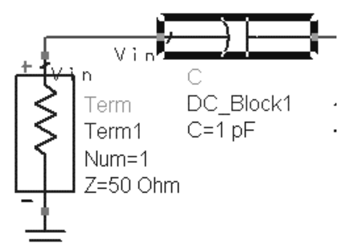
d. **Simulate** the new input network with a new dataset name: **s_match_in**.



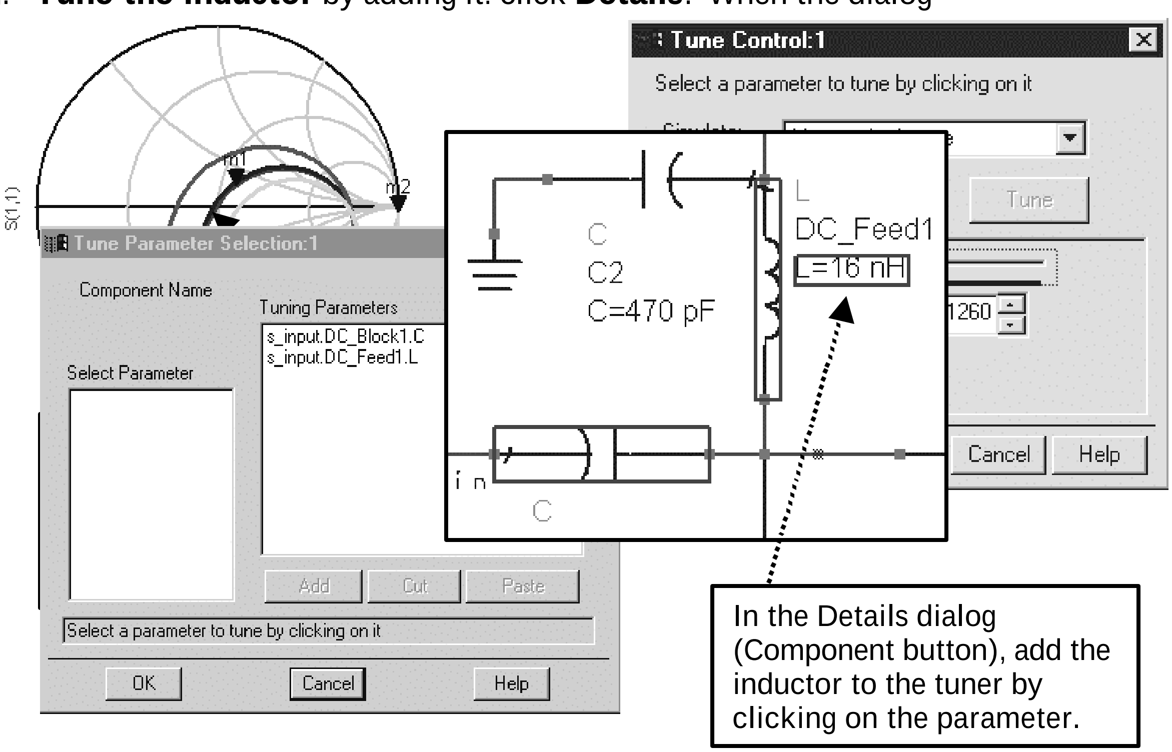
e. Plot the results and you should see a response like the one shown here where marker 1 is at the RF and marker 2 is the IF (almost an open). However, the response can be more finely tuned (next steps) so that the trace crosses directly through the 50 ohm point.



f. **Select the blocking capacitor** and start **tune** mode. Adjust the value of capacitance until the trace cuts though the center of the Smith chart. The next step will be done to adjust the inductor so that 900 MHz is directly in the center.

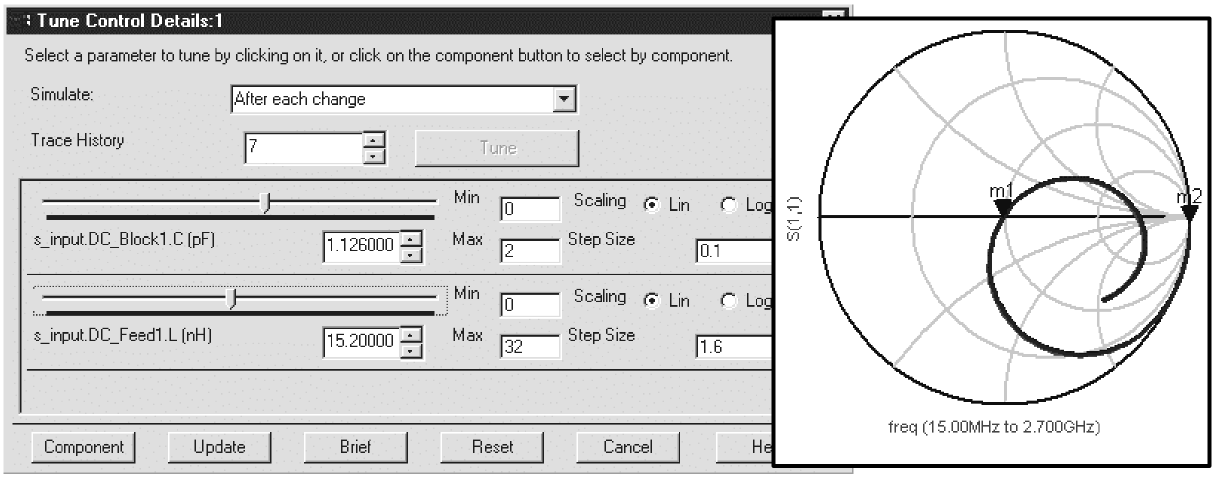


g. **Tune the inductor** by adding it: click **Details**. When the dialog



appears, select the **Component Button** and add the inductor by clicking on the parameter value (not the component) L=16 nH.

h. Adjust the inductance and you should get an almost perfect match at 900 MHz. In addition, the matching network is very efficient because it uses a minimum of components to block the dc, choke the RF, and shunt the unwanted IF frequency to ground. Click the **Update** button and the values will be updated on the schematic.

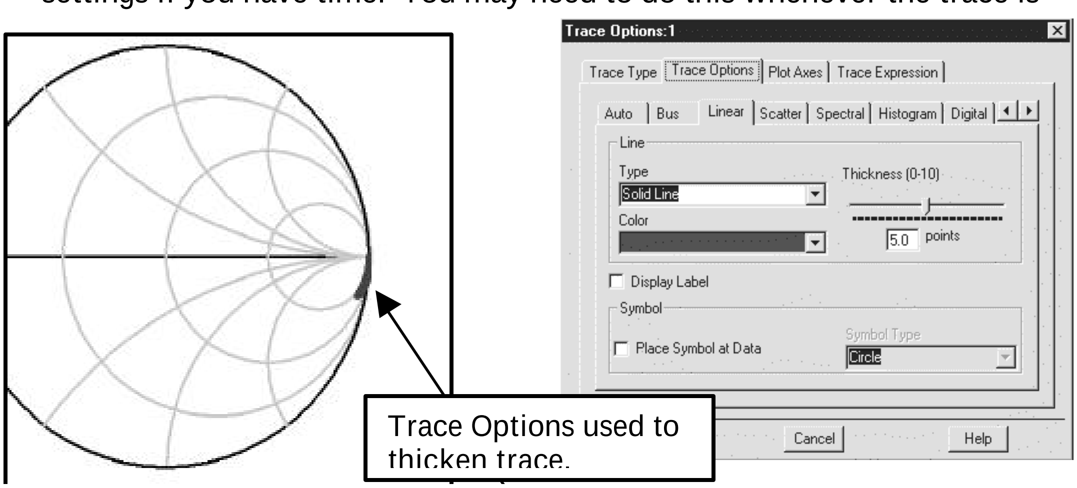


Design Note – L and C values: The tuned values of L and C will vary depending upon how finely you tune. However, C should be just about 1 pF and L should be between 15 and 16 nH for the following steps.

4. Examine the S-22 data

a. In the data display, insert a plot of S-22 from the last tuning simulation. You should see that S-22 is close to an open circuit over the frequency range.

b. Zoom into the trace area and double click on the trace. When **the Trace Options** dialog appears, thicken the trace and try using the other settings if you have time. You may need to do this whenever the trace is



difficult to see or when it is in a very narrow range. Build the output circuit.

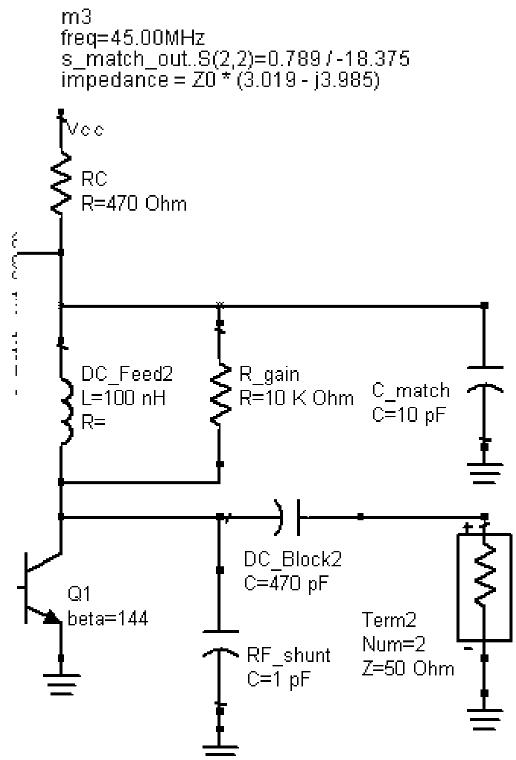
Output Match Design Note: For the next part of the lab exercise, you will use the optimizer to achieve the output match with a given topology.

5. Build the IF output matching network

Build the output to look like the one shown here. The DC feed is a 100 nH inductor in parallel with R_gain resistor (10K) which controls conversion gain. The capacitor (RF_shunt = 1 pF) will help short higher frequencies. Looking into the transistor from the 50 ohm load are two other capacitors for blocking (470 pF is a short to the IF) and C_match for matching.

6. Simulate and plot the S-22 results

Simulate (dataset name= s_match_out) and then note your results. The trace should be similar to the one shown here. S-22 at 45 MHz (shown by marker 3) is not matched to the characteristic impedance of 50 ohms. While you could use the tuner to try and achieve a match, the optimizer can also achieve the same goals.

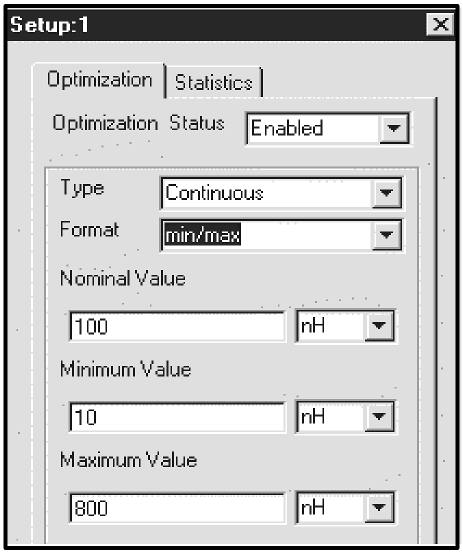


Optimization NOTE: The following steps show how to set up an optimization in three steps: 1) Enabling the components to be optimized, 2) Defining the Goals, and 3) setting up the Optimization control.

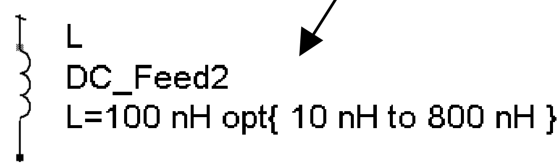
7. Enable the components to be optimized

a. Edit (double click) **the DC_Feed2** inductor and click the Optimization/Statistics Setup button.

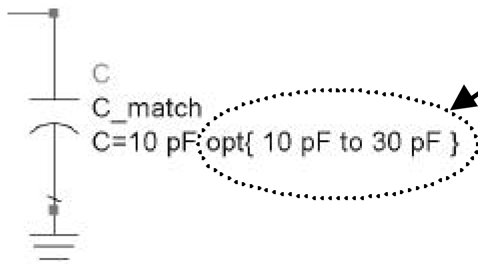
b. In the dialog, enable the dc feed inductor component for optimization, type, and range as shown. For this step, you will use Continuous optimization with min/max values: 10 to 800 nH. Click OK as needed.



The enabled component will show the nominal value and opt range. Use the F5 key to move the schematic component text so you can see it.



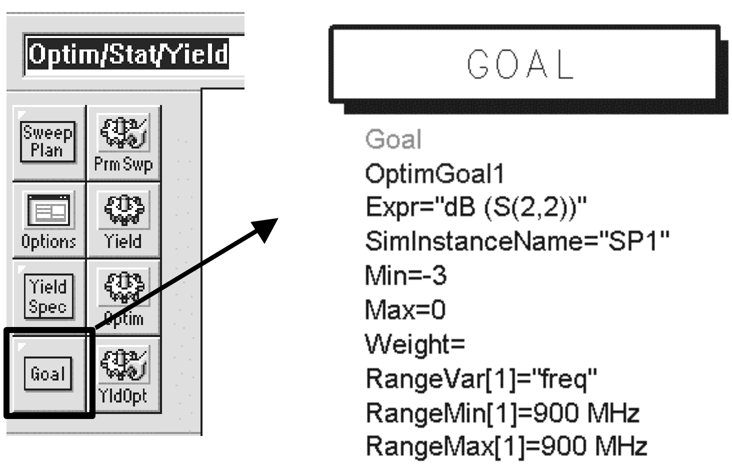
c. **Enable the C_match capacitor** for continuous min/max optimization also over the range of **10 to 30 pF**. Edit the component, using the dialog box to do this - after a component is enabled for optimization, the annotation will appear. Or, you can edit it directly on the screen by typing in the **opt** function and range as shown here.



Components can be enabled for optimization by on-screen editing using the **opt** function and the range in curly braces.

8. Define optimization goals

a. Insert the first optimization goal from the Optim/Stat/Yield palette. Goals are required (named) in the optimization component. Set up the goal as shown using the steps here:



NOTES: You can also edit the goal by double clicking on it.

The 900 MHz **range** is required by the simulator.

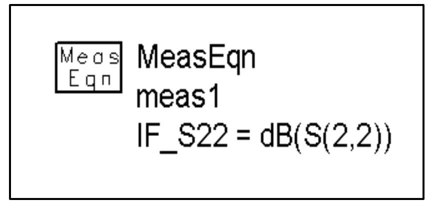
b. Enter the **Expr**, which is return loss: **"dB S(2,2))"**

c. Type in the SimInstanceName - the name of the S-parameter simulation controller: "**SP1"**.

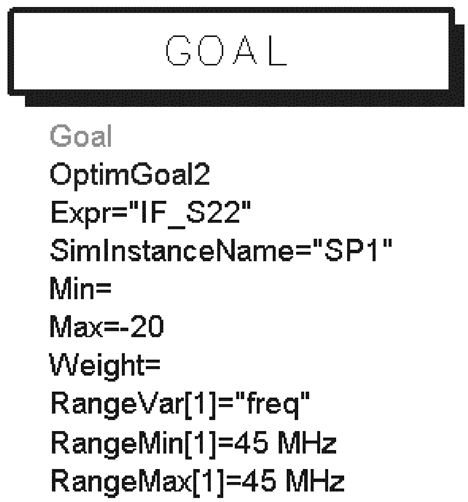
d. Type in the **Expr min/max range**: **–3 dB to 0 dB of return loss**

e. Type in the Range Variable: use the global variable **"freq"** and set the range which will be at one frequency: **900 MHz**.

f. **Insert a measurement equation to be used in the second goal.** Measurement equations are found in all simulation palettes. This goal will be available in the dataset. Type in the equation as shown where IF_S22 (or some name of your choice) will be the expression for achieving the IF return loss goal:



g. Insert the second **optimization goal** for the IF and type in the expression name as shown here. Enter the max goal value of –20. There is no need to set min or you can set it to –1000).

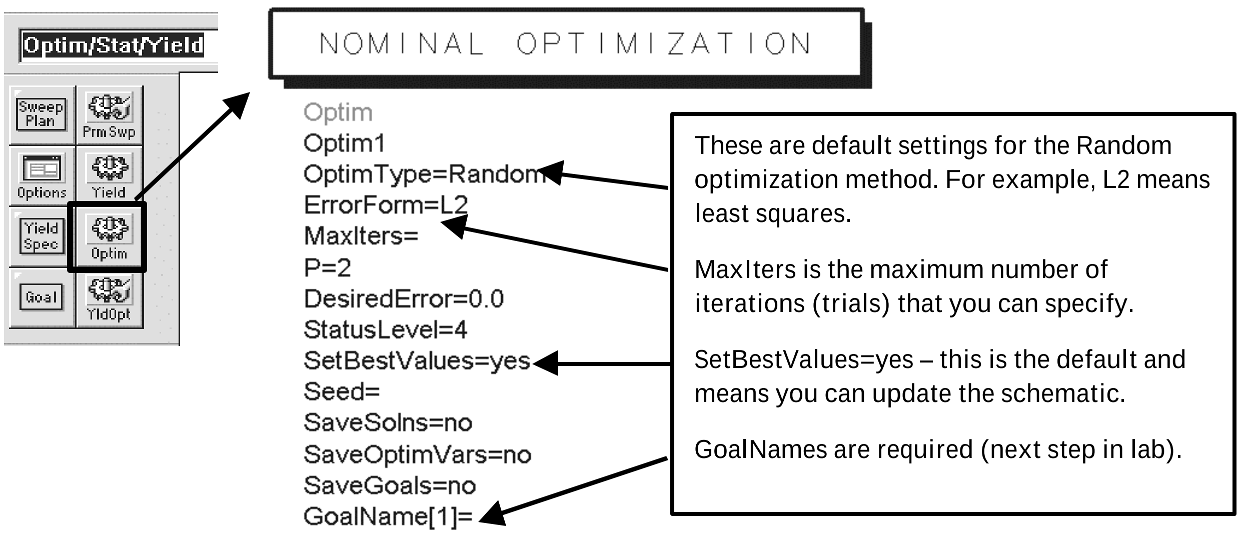


Review of Opt Goals: Goals must refer to the simulation controller name: “SP1” (similar to a parameter sweep). The ***expression*** usually refers to the measurement (data in array form). By specifying a min and max range for the expression, you are specifying what goal you want to achieve. Here, the goal is to have an IF match of at least -20 dB (no min is required) and an RF match between 0 and -3 db. In simple terms, you want a good match at 45 MHz at the output and a bad match on the output at 900 MHz.

9. Set up the Optimization control

The optimization component controls the simulation by receiving data and testing the data until the goals are reached or the maximum number of iterations has expired.

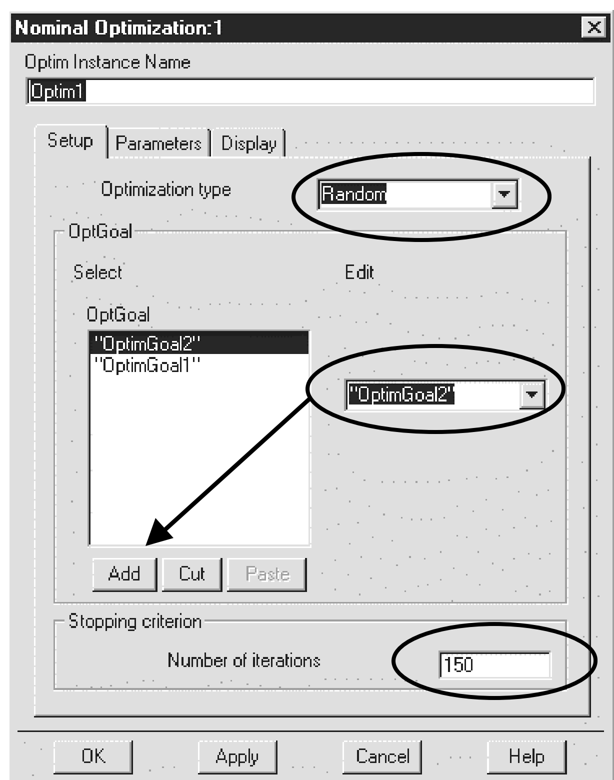
a. Select **Optim/Stat/Yield** in the schematic window palette and insert the Nominal Optimization controller (Optim).



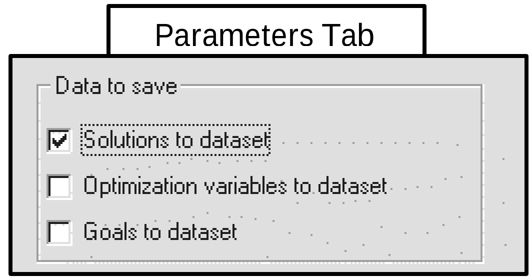
b. Edit (double click) the Optimizer control cmponent and add the two goals (**OptGoal**) by clicking their names. If you do not select specific goals, the default is to run all the goals.

c. Be sure to select and use **Random** optimization (most common).

d. Use **150** iterations. For Random optimization, one iteration is a successful simulation and may or may not get closer to the goal.



e. In the Parameters tab, check the box for Solutions to dataset. This will put the S parameters in the dataset. Also, always be sure the **Set best values...** box is checked (yes on display). This allows the optimized component values to be updated on the schematic.



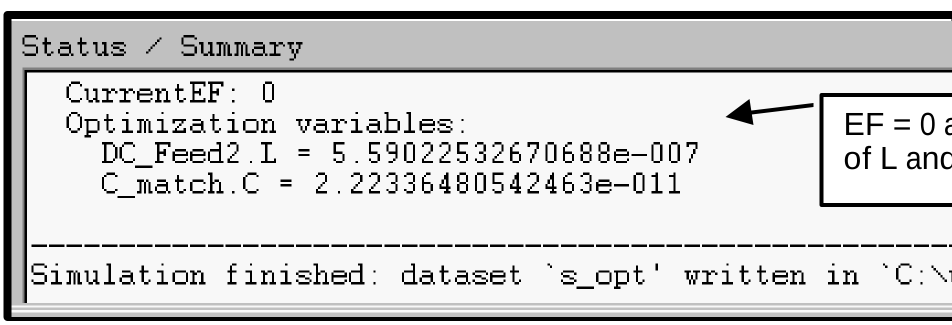
Parameters Tab Note: The ***Data to save*** selections can create large datasets that you may not need. To avoid this, do not check any boxes and, if you achieve the goal (EF=0), update the component values, deactivate the optimizer and do a regular simulation. However, for this lab, you will use the Solutions to dataset.

f. In the **Display** tab, set only the things you want to be displayed – this is a good practice for keeping organized schematics and simulations.

10. Optimize

a. Use a new dataset name (such as **s_opt**) and **Simulate (F7)** with the simulation set 15 MHz to 2 GHz with 5 MHz steps to land on RF and IF.

b. Watch the **Status Window** for the results of the optimization. Use the scroll bar if necessary to read it. If the optimization is successful, you should see a message that the EF (error function) = 0. If not, check your work, or try another type such as Gradient, or adjust the ranges.



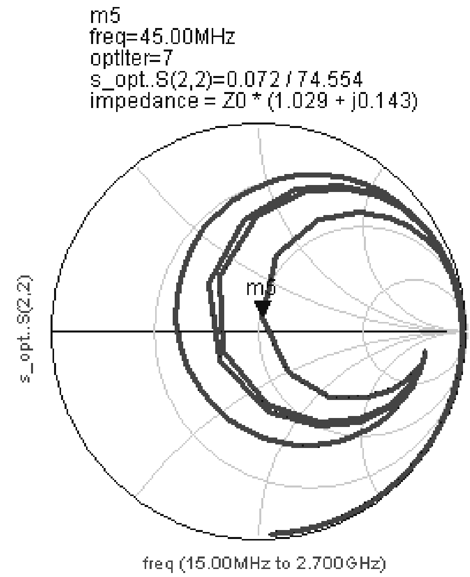
EF = 0 and the values of L and C are given.

c. If the EF is 0, go to the schematic and click **Simulate > Update Optimization Values**. The optimized values of L and C will appear as exact values but you can round them off. Here, C is about 22 pF and L is about 560 nH (your answer may vary slightly).

C C_match C=2.223365e+001 pF opt{ 10 pF to 30 pF }	L DC_Feed2 L=5.590225e+002 nH opt{ 10 nH to 800 nH }

11. Plot the S22 data.

It will be similar to the plot shown here where all the successful iterations are traced. Notice that one of the traces is near the center of the Smith Chart (marker). That trace represents the last optimization iteration where the goals were met.



12. List the *meas eqn* data

a. Insert a list of your equation: **IF_S22** that was used in the goal. The equation will be in the same dataset as the S-parameters (s_opt). You should see the value of the equation at 45 MHz which represents the optimized goal.

freq	s_opt..IF_S22
15.00MHz	-2.096
20.00MHz	-2.493
25.00MHz	-3.176
30.00MHz	-4.417
35.00MHz	-6.931
40.00MHz	-13.368
45.00MHz	-20.102
50.00MHz	-7.794
55.00MHz	-3.972
60.00MHz	-2.274
65.00MHz	-1.429
70.00MHz	-0.970

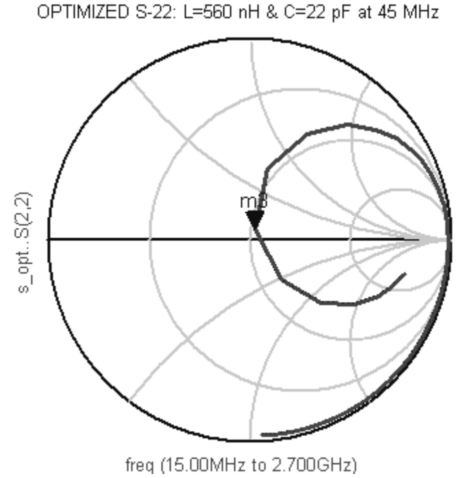
Your measurement equation: IF_S22 = dB (S(2,2)), from the schematic is shown for the 45 MHz IF as reaching –20 dB of return loss using the optimized values of L and C.

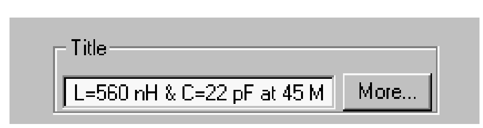
b. **Deactivate** the Optimizer and edit the component values on screen by highlighting and deleting the unwanted values and typing in the values of L and C as: **L = 560 nH and C = 22 pF**.

C=22 pF ~~opt{ 10 pF to 30 pF }~~

c. **Simulate** and your plot of S-22 will now have only one trace similar to the one shown here. Also, edit the plot and use the **Plot Options** to title the plot.

OPTIMIZED S-22: L=560 nH & C=22 pF at 45 MHz





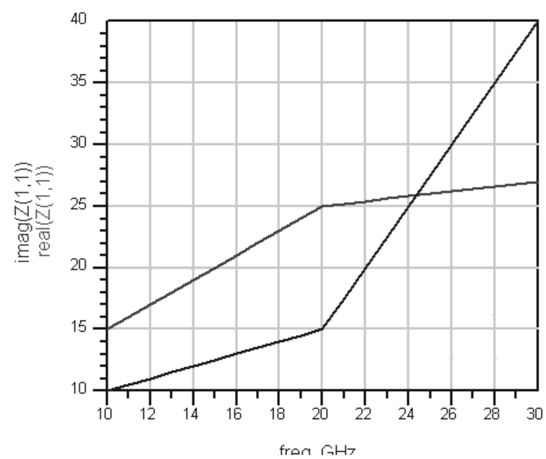
At this point the mixer has good input and output matching networks. Of course, you could refine the output match with the tuner but it is not necessary.

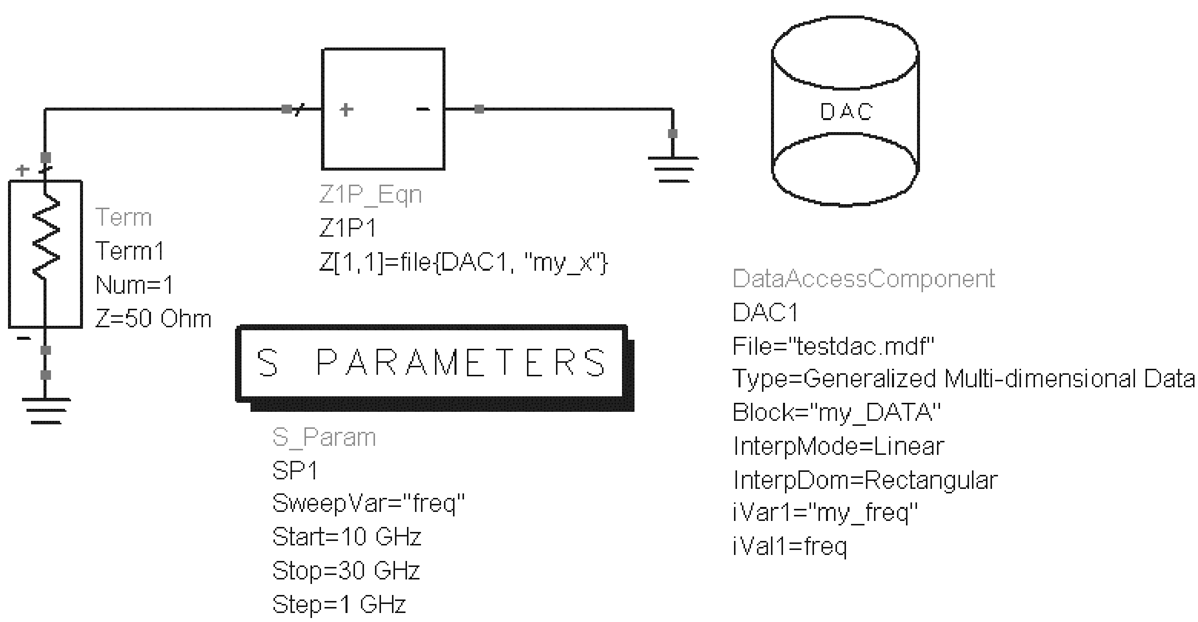
NOTE on the *opt* and *noopt* function: Refer to the schematic where the optimized component value had annotation such as: C=7.95462189+001 pF opt{ range]. If you type noopt instead of opt, that component (noopt) will not be optimized. This is easier than editing the component in the dialog box.

EXTRA EXERCISES:

1. Optimize again using gradient method instead of random or try to optimize to better goals: S-22 = -25 or better dB at IF. To do this, try using another optimization type such as genetic.

2. Try using a DAC component to create a frequency sensitive inductor. As the plot here shows, the real and imaginary values change with frequency. These curves are described by a file which is read by the DAC. To do this, you need to write a file for the data and build the schematic required schematic. Step by step instructions follow on the next page…



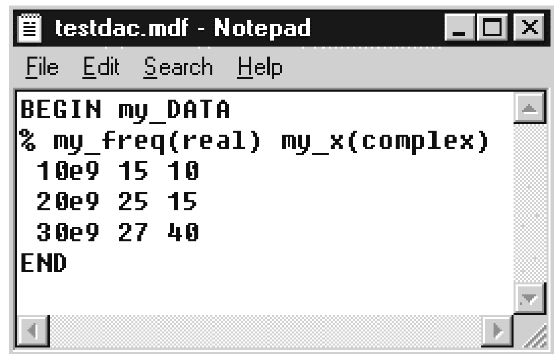


DAC instructions:

a. Open a new schematic saved it as DAC_Z. Refer to the previous circuit and insert the components in their default state:

- S-parameter controller, Termination and ground, Z1P from the equation based linear palette, and a DAC from the Data items palette.

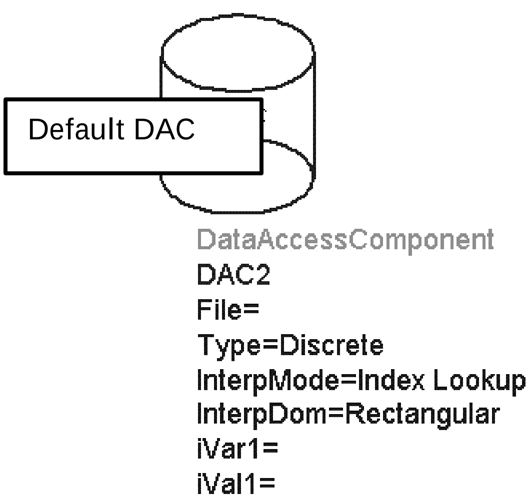
b. Write an **mdf** file using the ADS main window Options > Text Editor (use only Note pad not Word pad which has formatting - this is a must). Write the file shown here and save it in the DATA directory as: **testdac.mdf**. If necessary, you may need to use the windows file explorer to change the name if it is saved as a .txt file. Also, be careful of the syntax in the file - the first column contains 3 frequency points, the second and third columns contains the real and imaginary parts of the reactive component.

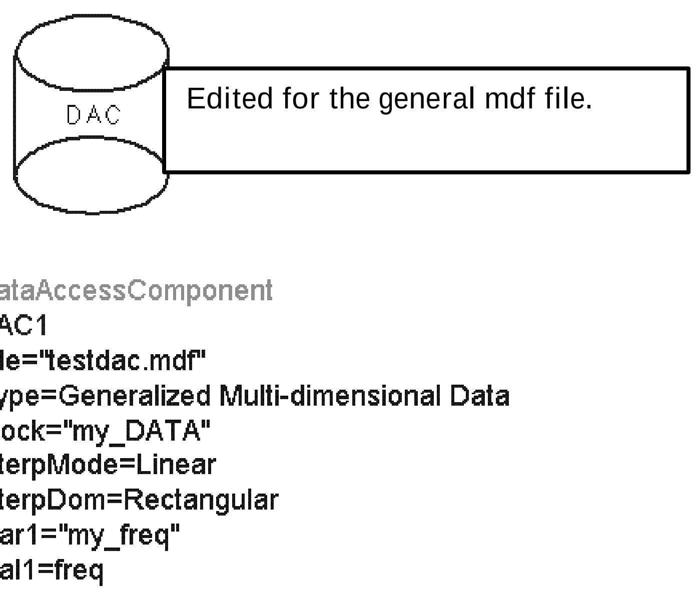


c. On schematic, edit the S-parameter controller. In **Parameters tab**, set to compute **Z parameters** not S. In the **Display ta**b, check the the Sweep Var and start, stop, set and set them as shown to sweep the global variable **"freq"** from 10 to 30 GHz in 1 GHz steps. You will get interpolated data for all the steps.

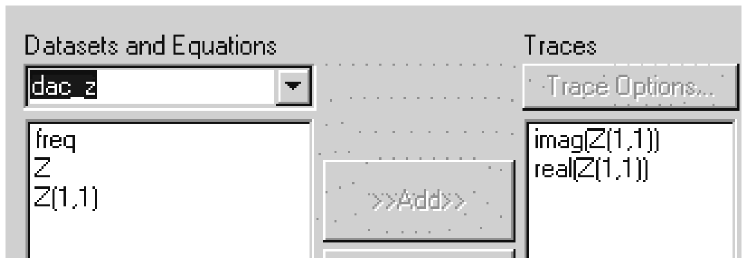
d. On schematic, set the Z1P value of **Z[1,1]= file{DAC1,"my_x"}**. The value of Z11 is the variable "my_x" in the DAC1 file. Of course, the file is testdac.mdf.

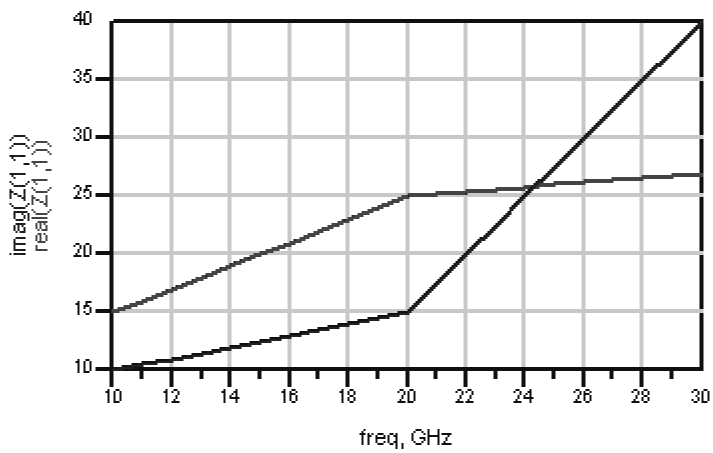
e. On schematic, edit the DAC as shown here. IVar1 is the independent variable and iVal1 is the swept variable. As "freq" is swept, "my_freq" will be indexed and the DAC will return complex values of "my_x" interpolated over the frequency range.





f. Check the circuit and simulate. Then plot two traces, real and imag, of Z(1,1) as shown where X changes with frequency. Now, the Zport can be used wherever a frequency sensitive component is required. For multiple components, simply create different files and access them as required.





THIS PAGE IS INTENTIONALLY BLANK