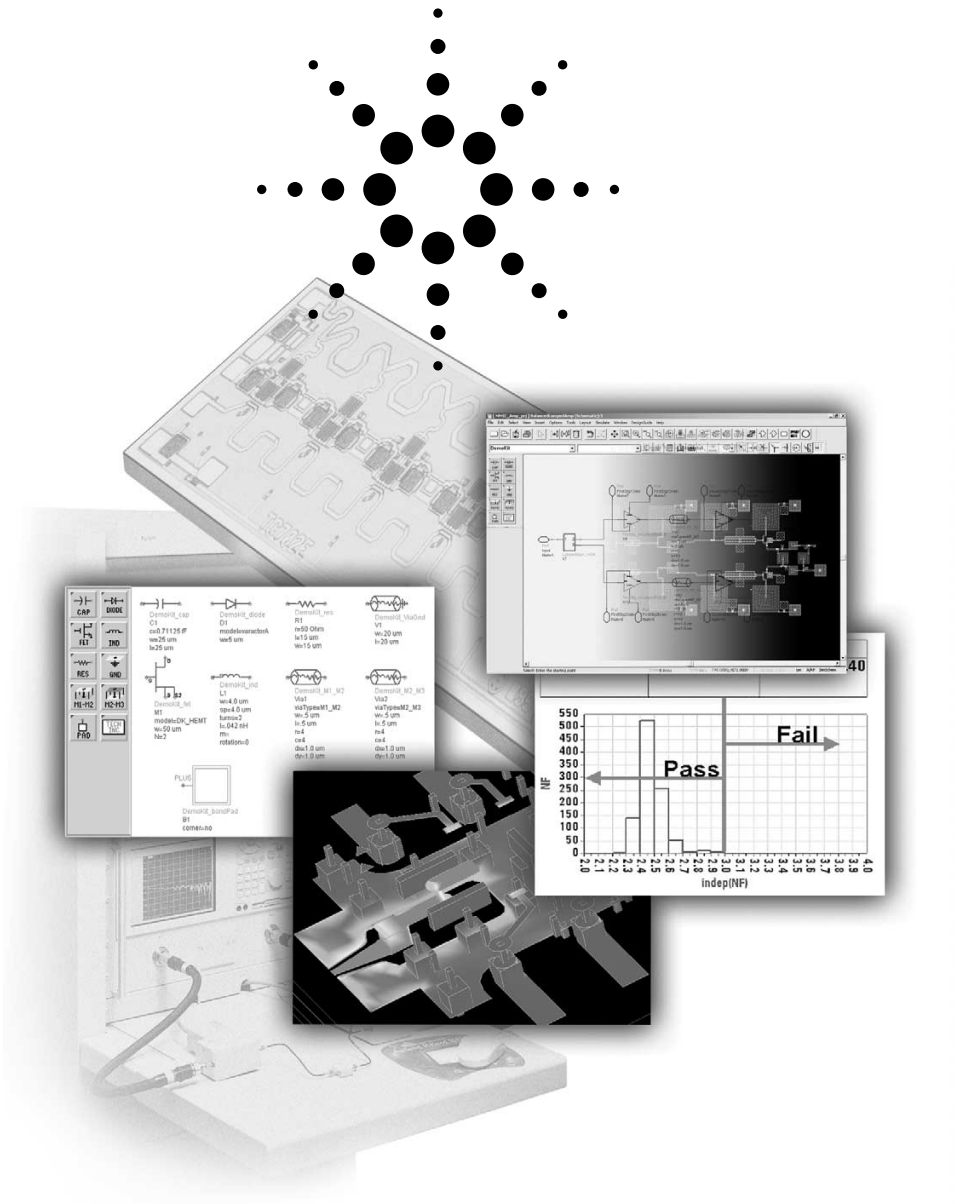
Using Advanced Design System to Design an MMIC Amplifier

Agilent EEsof EDA

Application Note Number 1462



Agilent Technologies

Contents

Appendixes

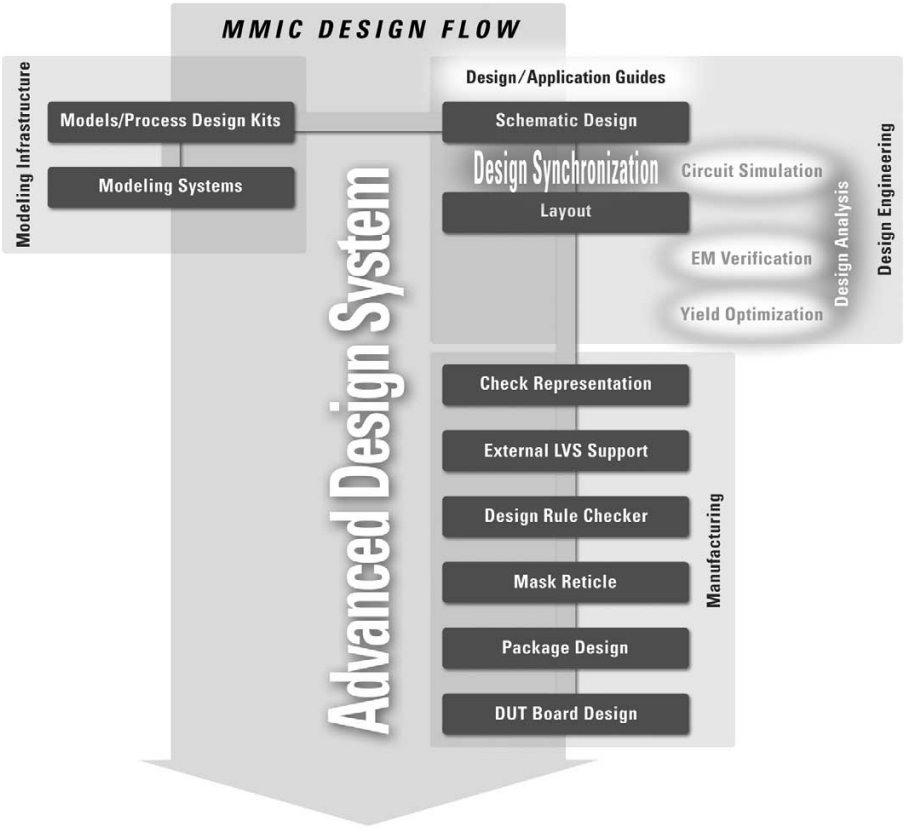
General References on Balanced Amplifiers

"Microwaves and RF Circuits: Analysis, Synthesis and Designs"
Max W. Medley, Artech House, 1993, pp 509-541.

"Foundations for Microstrip Circuit Design"
T.C. Edwards, John Wiley and Sons, 1981, pp. 242-244.

Introduction

There are many design steps required for the development and manufacture of MMIC circuits, as illustrated in the MMIC Design Flow. Advanced Design System (ADS) is a central part of the complete MMIC design flow, and is used throughout this process. This application note illustrates, through the design of an MMIC amplifier, several of the common problems faced in designing, simulating, and producing a physical layout of an MMIC circuit, as well as the validation steps that are needed to verify that the physical layout still produces the desired result. It is beyond the scope of this note to describe all possible design specifications for an MMIC circuit, but it does include enough specification and design steps to address many common design challenges.



The following sections give a step-by step description of a 0.5-Watt, 10-GHz, narrow-band amplifier on a 100-µm GaAs substrate. The example files that are used here (*$HPEESOF_DIR/examples/MW_Ckts/MMIC_Amp_prj and MMIC_AmpEM_Sims_prj*) are included with the ADS 2003A software. Design and data display file names from the examples are referenced throughout.

These example designs use components from a generic design kit that is provided with ADS 2003A (*$HPEESOF_DIR/examples/DesignKit/DemoKit*.) The models for these components do not correspond to any specific foundry process, but are representative of design kits available from many foundries.

Although the development of the generic design kit is outside the scope of this application note, it is documented in the ADS 2003A manual, titled *Design Kit Development*. This manual provides instructions that guide foundries in developing their own design kits.

The amplifier design process depends on a number of factors, including desired specifications, availability of device models, designer preference, and more. This application note describes one of many potential sequences. We assume that two stages of amplification will be required: an output stage for power and an input stage to attain sufficient gain. The design is a balanced amplifier, consisting of two parallel, two-stage amplifiers, with branch-line couplers at the input and output implemented as lumped element equivalent circuits, to split the signal at the input and recombine it at the output after amplification.

Figure 1 shows a block diagram of the balanced amplifier topology. Figure 2 outlines how the impedances to present to the devices were chosen.

The design flow begins with several simulation steps and proceeds with physical design steps, with some simulations for verification of the physical design. Three main design tasks are required to complete the amplifier: design of the first-stage (preamplifier), design of the second stage (power amplifier), and design of the branch-line couplers.

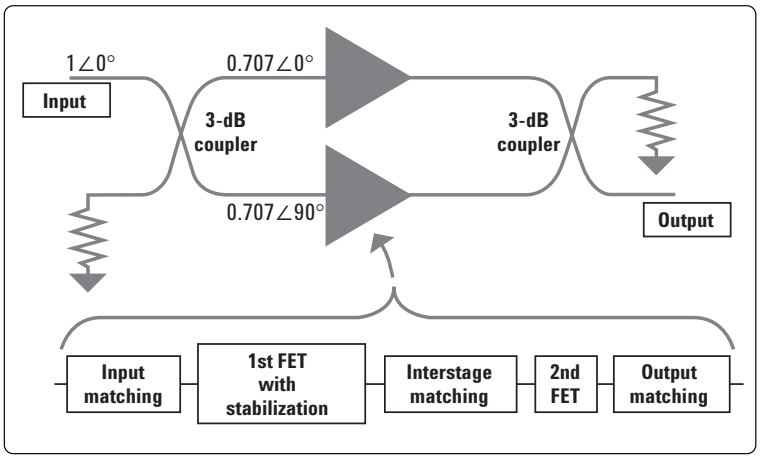


Figure 1. Balanced amplifier block diagram, utilizing two, two-stage amplifiers in parallel.

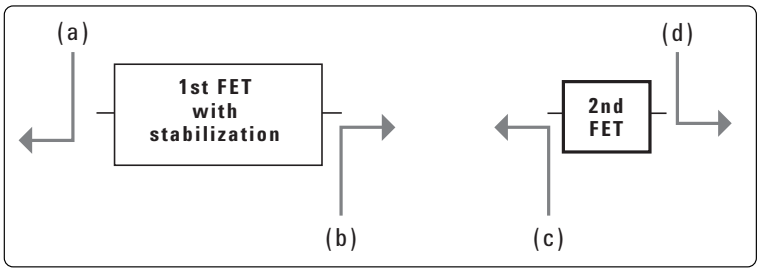


Figure 2. Choosing impedances. (a) Choose source Z for minimum noise figure, as long as gain remains reasonable. (b) Choose load Z for conjugate matching, after choosing source Z (although a slight mismatch was found to give a higher 1-dB gain compression output power). (c) Choose source Z for conjugate matching, after choosing load Z. (d) Choose load Z for maximum power delivered.

Two-Stage MMIC Amplifier Design

This section outlines the general steps for two-stage MMIC amplifier design.

1. **Select an active device.**
   This will depend on the specifications you are attempting to meet (such as frequency, power, and noise), and the devices offered by a particular foundry. The DemoKit has only one device, a HEMT (high electron mobility transistor), so a selection process is not required.

2. **Ideal bias – first-stage**
   Choose a bias point to maximize the transconductance, Gm, which should also maximize the gain of the first-stage device. The FET_Gm_Calcs schematic, shown in Figure 3, simulates the I-V curves of the device and calculates Gm at each bias point as a function of the slope of the IDS-versus-VGS curve.

   The plot in Figure 4 indicates that biasing VGS to about –0.15 V should maximize Gm. The Amplifier DesignGuide has an updated version of this simulation setup, under *DesignGuide > Amplifier > DC and Bias Point Simulations > FET I-V Curves, Class A Power, Eff., Load, Gm vs. Bias*, that calculates Gm versus DC bias point, using an AC simulation at one frequency to determine Gm.

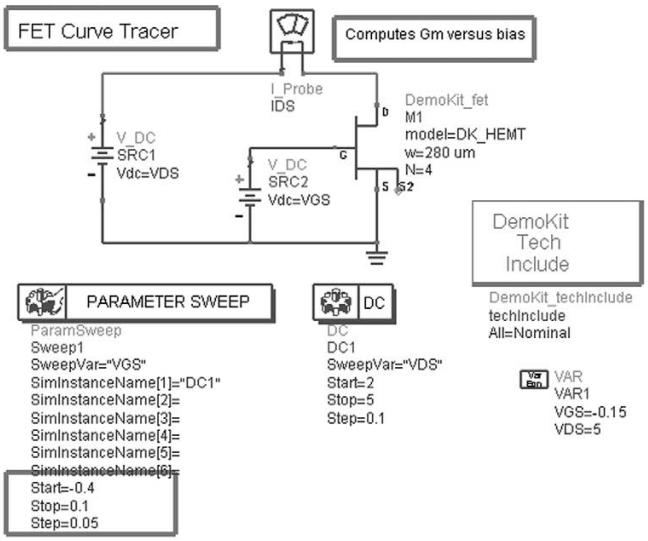


Figure 3. The FET_Gm_Calcs schematic, for simulating a device's transconductance versus bias.

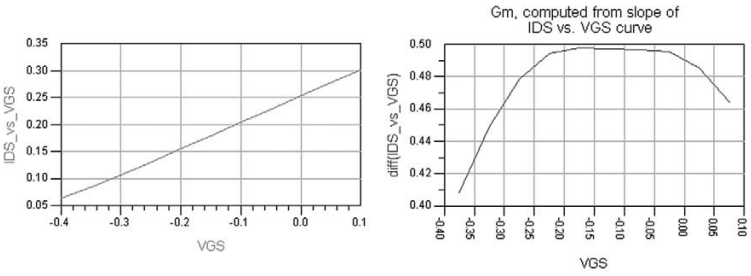


Figure 4. Plot of drain current, IDS, versus gate voltage, VGS, and transconductance, Gm.

3. Impedance matching with ideal elements – first-stage

Determine the optimal source and load reflection coefficients to present to the first-stage device, based on noise figure and gain. If noise is not important, then just design for gain. The simulation setup from the Amplifier DesignGuide (from a schematic, *DesignGuide > Amplifier > DC and Bias Point Simulations > FET NoiseFig., S-Params, Gain, Stability, and Circles vs. Bias*) is shown in Figure 5.

In this setup, the gate and drain voltages are swept, and the S-parameters and noise parameters of the first stage device are simulated at 10 GHz, at each bias point. The corresponding data display shows noise, gain, and stability circles, which are all updated, depending on the bias point you select with a marker, as shown in Figure 6. About a 1-dB noise figure and > 16 dB of gain should be achievable, but the device is potentially unstable, as indicated by the source stability circle being well within the unit Smith chart.

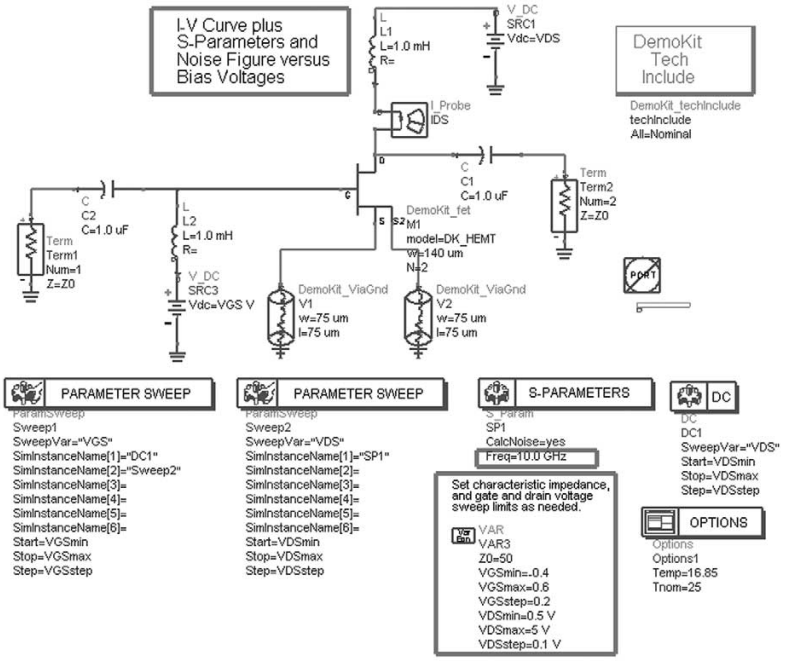


Figure 5. The FET_SP_NF_Match_Circ schematic, for simulating a device's S-parameters, gain, noise figure, and stability versus bias.

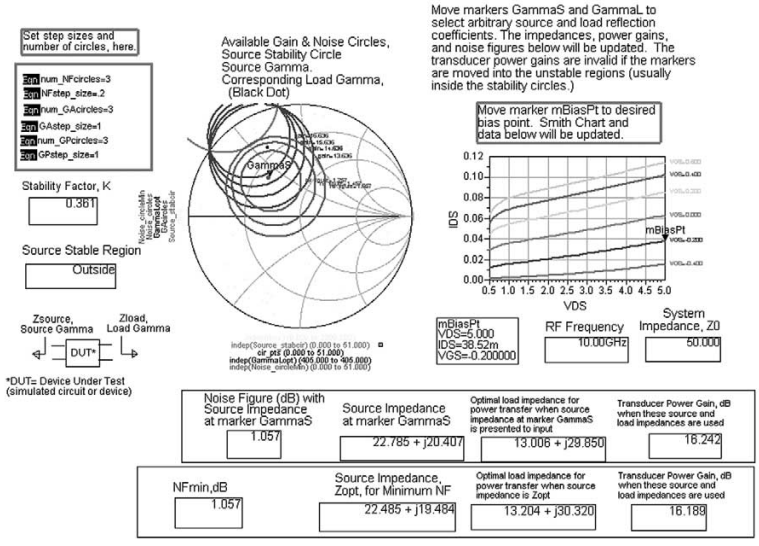


Figure 6. The gain, noise, and stability circles are plotted for the bias point selected by marker mBiasPt.

4. Attaining stability with ideal elements – first-stage

Feedback elements are added between the gate and ground and between the gate and drain of the first-stage FET to improve stability. Optimize the stability circuits using the Gain_and_Stab_opt schematic from the Amplifier DesignGuide (from a schematic, *DesignGuide > Amplifier > S-Parameter Simulations > Feedback Network Optimization to Attain Stability*), as shown in Figure 7.

This simulation includes goals to force the geometric source and load stability factors, mu_source and mu_load, respectively, to be >1 over a broad frequency range. [Ref. M. L. Edwards and J. H. Sinsky, "A new criterion for linear 2-port stability using geometrically derived parameters", IEEE Transactions on Microwave Theory and Techniques, Vol. 40, No. 12, pp. 2303-2311, Dec. 1992.]

If these stability factors are >1, then neither the source nor load stability circle intersects the unit Smith chart. Minimum noise figure and gain are included as optimization goals, otherwise performance might be degraded too much to attain stability. The results, shown in Figure 8, show good stability performance and reasonably good gain and minimum noise figure, but with ideal lumped elements.

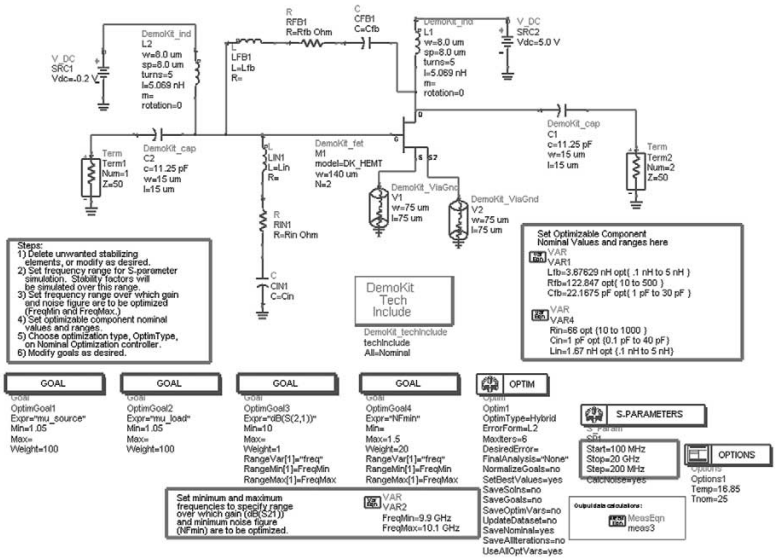


Figure 7. Optimization of feedback and shunt R, L, and C values to attain stability without degrading noise figure and gain too much.

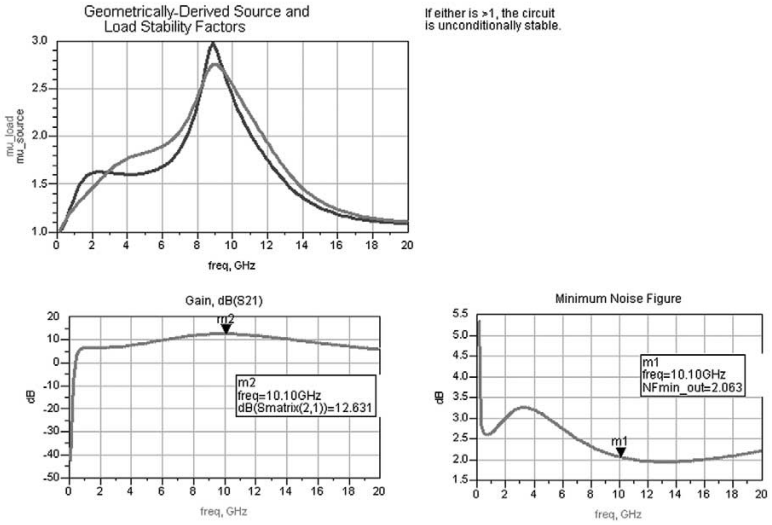


Figure 8. Gain and stability optimization results.

5. **Replace ideal elements with design kit elements – first-stage**
Replacing the ideal elements in the stabilization network with design kit elements shows a degradation in stability. We run a discrete-value optimization to adjust the design kit elements to attain better stability. Discrete-value optimization is necessary if some of the parameters to be optimized may have only discrete values, such as the number of turns of a spiral inductor. The result of the optimization is shown in Figure 9.
To improve stability near DC, we added a 10-Ohm resistor in series with the inductor biasing the gate of the device, at a later step in the design. Discrete-value optimization can be quite time-consuming, since it carries out an exhaustive search of all possible combinations of parameter values. It is recommended to first run a continuous optimization to get ideal element values as a starting point, and then run a discrete-value optimization, allowing the parameter values to vary over only a limited range. Certain continuous optimization types (mainly random and its variations) will handle both continuous and discrete optimizable variables.

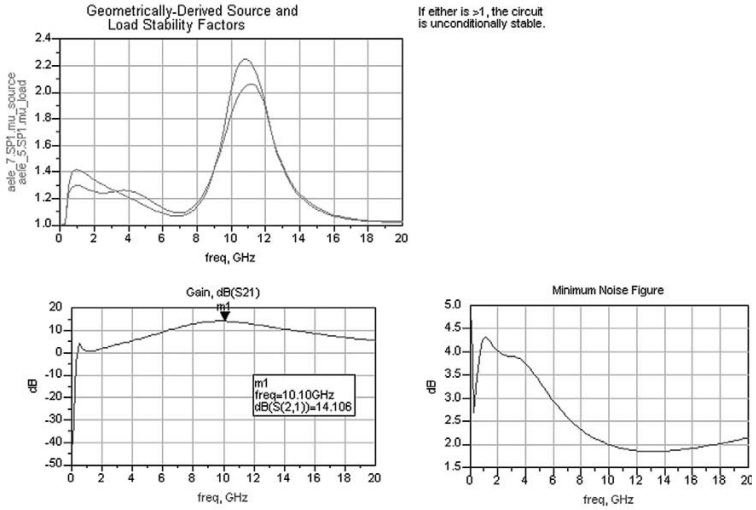


Figure 9. Discrete value optimization results, using components from the DemoKit.

6. **Impedance matching – first-stage with stabilization network**
Determine the optimal source and load impedances to present to the stabilized FET, via S-parameter and noise figure simulations, using the SP_NF_GainMatchK schematic from the Amplifier DesignGuide (from the schematic *DesignGuide > Amplifier > S-Parameter Simulations > S-Params, Gain, NF, Stability, Group Delay vs. Swept Parameters*, shown in Figure 10).

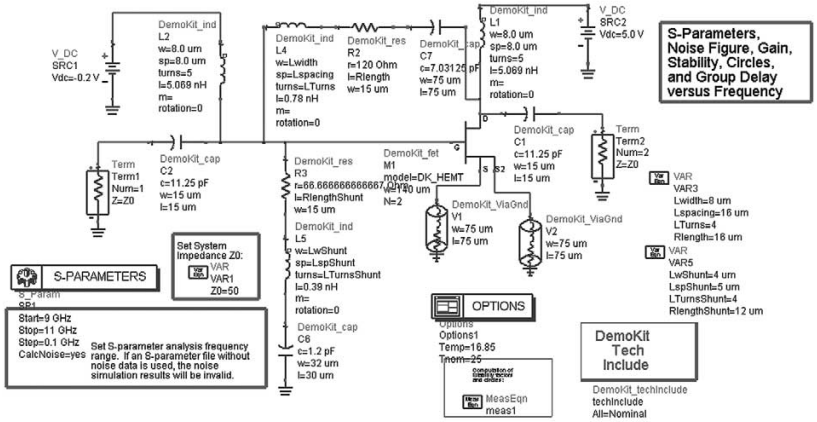


Figure 10. Simulation to determine the optimal source and load impedances for gain or minimum noise figure for the first stage device with stabilization network.

The data display, shown in Figure 11, shows that with a source impedance of 21.3 +j*3.7 ohms, the noise figure is about 2.0 dB. With this source impedance, the corresponding optimal load impedance is 65.1 +j*38.5 ohms, which should give a transducer power gain of 13.1 dB. It was later discovered experimentally that generating a load impedance of 39.5 +j*52.9 ohms gives a higher one-dB gain compression output power for the two-stage amplifier, at the expense of lower gain, so this impedance was used instead. If gain is more important than noise, then a source impedance to maximize gain could be chosen.

7. Load pull – second stage device

For the second stage we want to generate more output power, so we experiment with the device size. A device size four times as large as the first stage device was selected, although a larger device should give more output power. A load pull simulation, HB1Tone_LoadPullMagPh, copied from the examples/RF_Board/LoadPull_prj, showed 26.7 dBm power delivered, with a load of 7.76 +j*9.7 ohms, as shown in Figure 12. (Additional load pull utilities are available in the load pull application, under DesignGuide > Loadpull, from a schematic window.)

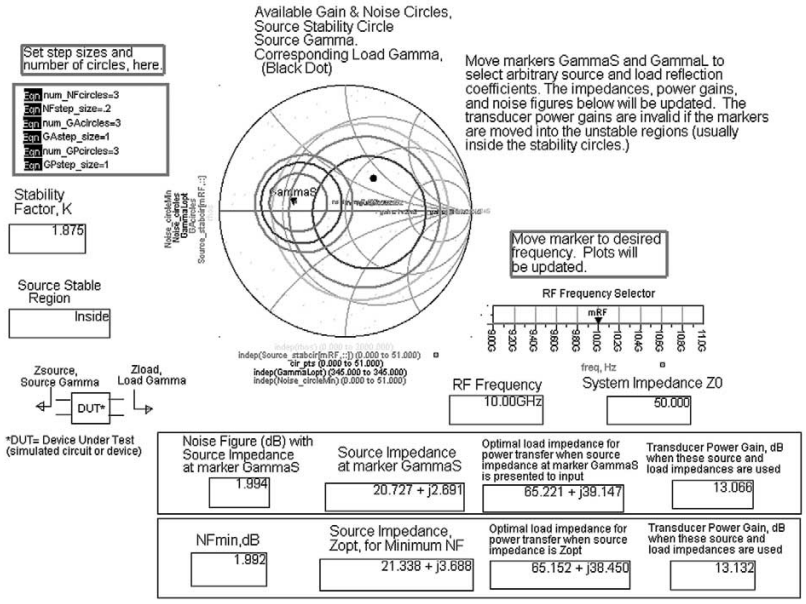


Figure 11. Gain and noise circles and optimal source and load impedances for minimum noise figure.

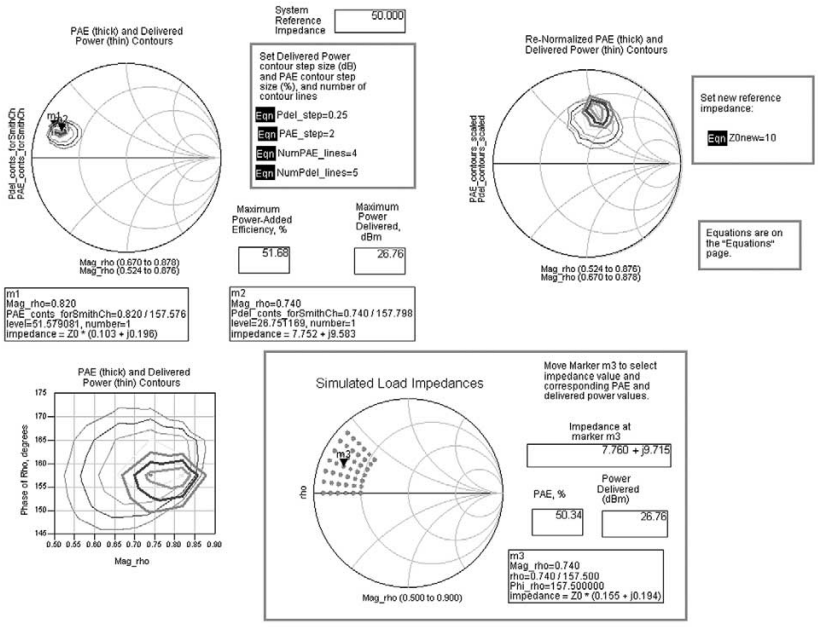


Figure 12. Load pull simulation results.

8. Source pull – second stage

A source pull simulation, *HB1Tone_SourcePull*, from the Amplifier DesignGuide, indicates that the power delivered to the load does not depend much on the source impedance. So the interstage matching network is designed to provide the complex conjugate as the source impedance to present to the second-stage FET, while this FET is terminated in the optimal load impedance determined from the load-pull simulation.

9. Designing the input matching network

There are three matching networks to be designed: the input to the first stage, the interstage between the first and second stage, and the output of the second stage. Things to consider in choosing these networks include the size of the passive elements, incorporating DC blocking capacitors, and making some of the networks high-pass and others low-pass, so the overall response is band-pass. Because of the relatively low operating frequency, distributed-element matching would require too much space, so we use lumped elements instead. Because this impedance matching is at a single frequency only, two-element, lumped matching may be realized quite simply.

The Passive Circuit DesignGuide was used to generate a simple, lumped-element matching network to generate the desired source impedance, as shown in Figure 13, and the resulting network is a simple shunt-C, series-L network. (Note that this same match can be found in the Matching utility or the Smith Chart utility.)

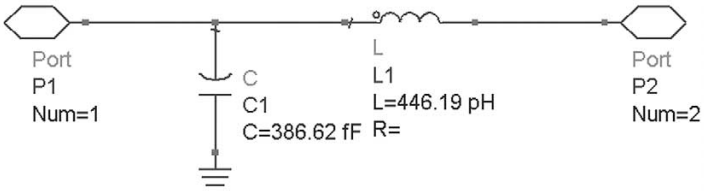


Figure 13. The input matching network.

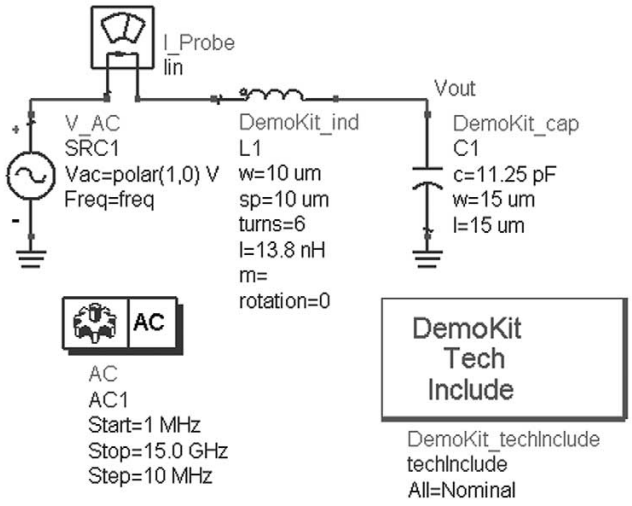


Figure 14. Simulating the impedance of a bias network.

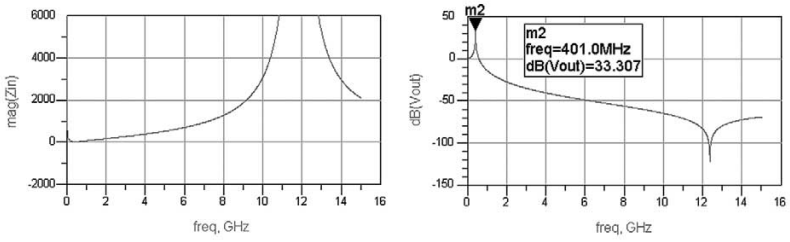


Figure 15. Bias network impedance simulation results.

10. Replace ideal elements with design kit elements

The network with ideal elements must be replaced with design kit elements, which have parasitics. The parasitics vary with the size of each component. You want the DC-bias inductor to be large enough to provide a high impedance at 10 GHz, but not so large that its parasitic capacitance to ground causes a self-resonance to occur below this frequency. Figure 14 shows the setup for simulation of the impedance of a simple bias network.

The results are shown in Figure 15.

The input matching circuit with design kit elements and a DC bias network is shown in Figure 16.

The corresponding layout is shown in Figure 17. Since this circuit is used to bias the gate of the first-stage device, there should be little or no bias current drawn from the supply, so a resistor could be used instead of the inductor. This has the additional benefit of saving GaAs real estate.

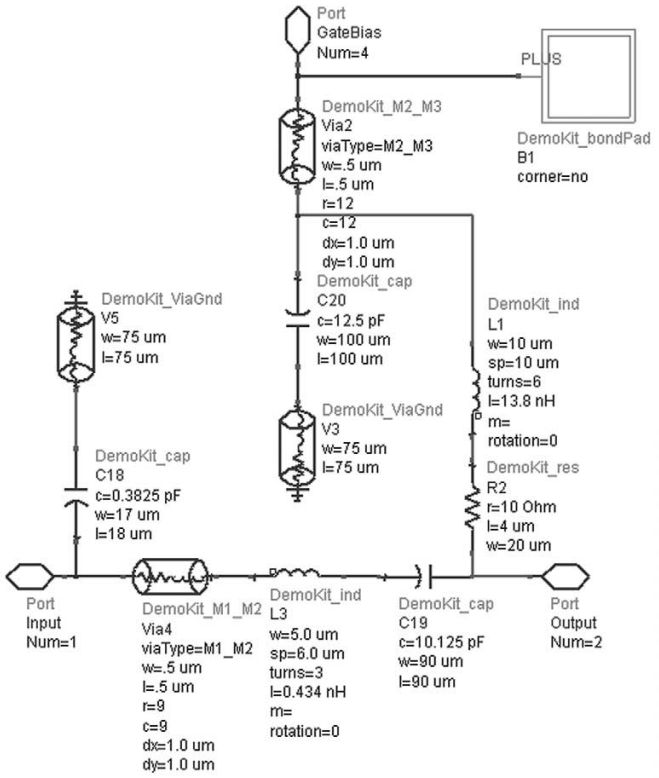


Figure 16. Input matching circuit schematic, with design kit elements and a DC bias network.

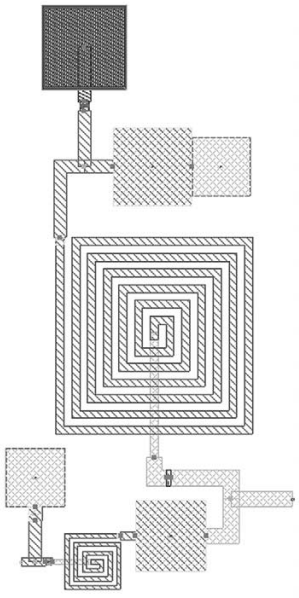


Figure 17. Corresponding input matching circuit layout, with design kit elements and a DC bias network.

11. Interstage match

The interstage network transforms the input impedance of the second-stage device to the optimal load impedance to present to the first-stage device. The Matching utility was used to design the simple shunt-C, series-L matching network (or the Passive Circuit DesignGuide could be used). The InterstageMatch_wBias network is shown in Figure 18. It shows the interstage matching network including design kit elements and DC bias inductors.

The layout, shown in Figure 19, has the drain bias inductor of the first stage very close to the gate bias inductor of the second stage, so coupling is likely to occur. The amount of coupling and to what degree it degrades circuit performance can be determined from a Momentum (electromagnetic) simulation, although we did not perform one for this example.

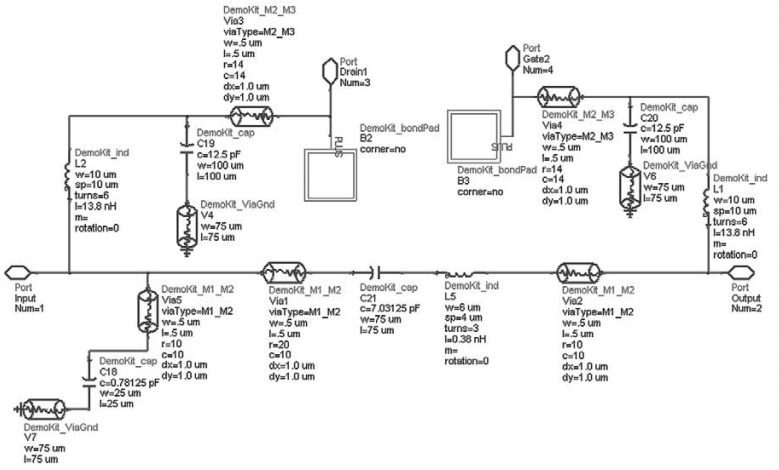


Figure 18. The interstage matching network, including design kit elements and DC bias inductors.

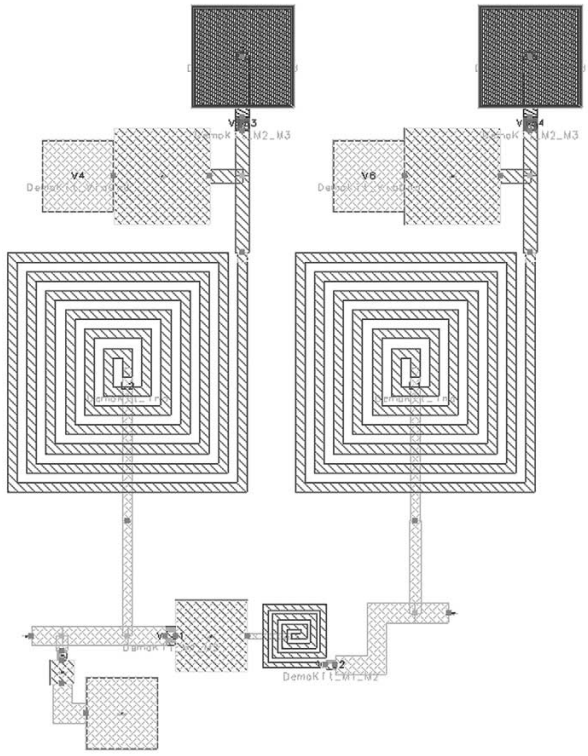


Figure 19. Interstage matching network layout.

12. Output match – second stage

The output matching network is used to transform 50 Ohms to the optimal load impedance (7.76 +j*9.7 Ohms) to present to the output of the second stage device. An ideal, shunt-L, series-C network is created using the OutputMatch1 schematic in the example, which came from the Amplifier DesignGuide (another option for generating impedance matching networks). The OutputMatch_wBias schematic (in the example file, and not shown here) uses design kit elements instead of ideal elements, and includes a DC bias inductor.

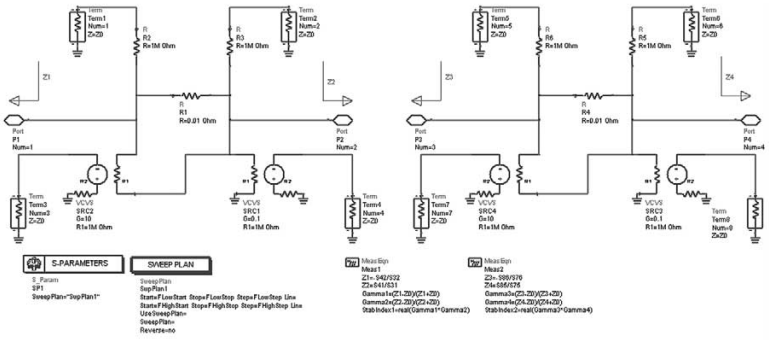


Figure 20. S-probe pair schematic.

13. Interstage match and stability verification – S-probe

The next step is to verify that when we connect the matching networks, input device with its stabilization network, and output device that we are generating the desired source and load impedances at each device. Also, we need to verify that the stability conditions are satisfied at the input and output planes of each device. An S-probe is used to determine the source and load impedances at the input and output of each device.

The S-probe is an element that you can insert anywhere into a circuit without loading it. It will determine the impedances and reflection coefficients looking in both directions. From these reflection coefficients, we can determine whether the small-signal stability conditions are satisfied or not. The S-probe pair schematic used in this example is shown in Figure 20.

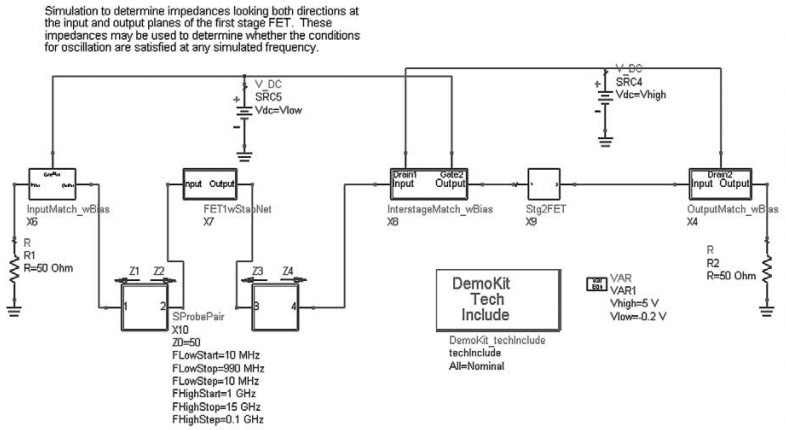


Figure 21. Determining the impedances looking both directions, at the input and output of the first stage FET.

14. Input match verification

The TwoStgAmpInZ_TB, shown in Figure 21, determines the source and load reflection coefficients presented to the first stage device.

The data display in Figure 22 shows that these impedances are close to the desired values at 10 GHz, and that the stability conditions are satisfied from 10 MHz to 20 GHz.

15. Output match verification

TwoStgAmpOutZ_TB (shown in Figure 23) determines the source and load reflection coefficients presented to the output device.

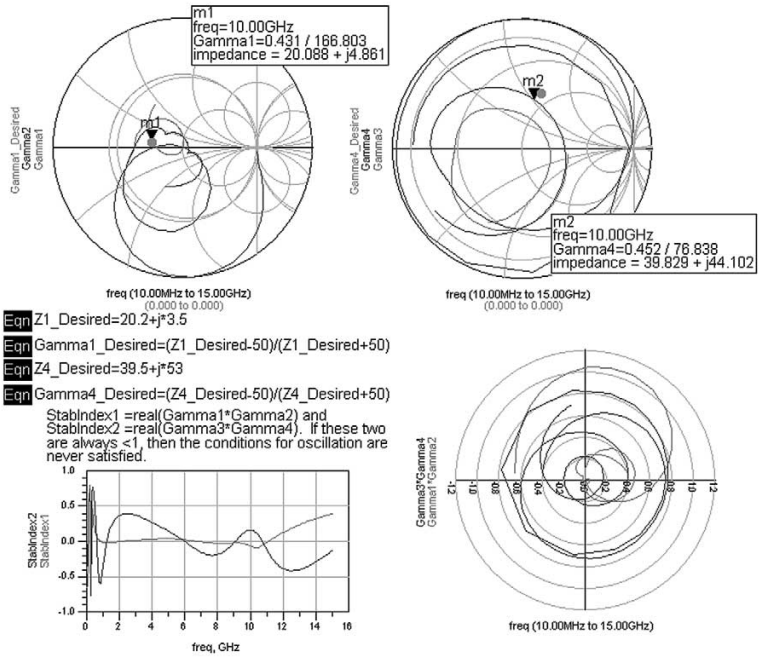


Figure 22. Source and load impedances close to the desired values are being generated by the matching networks.

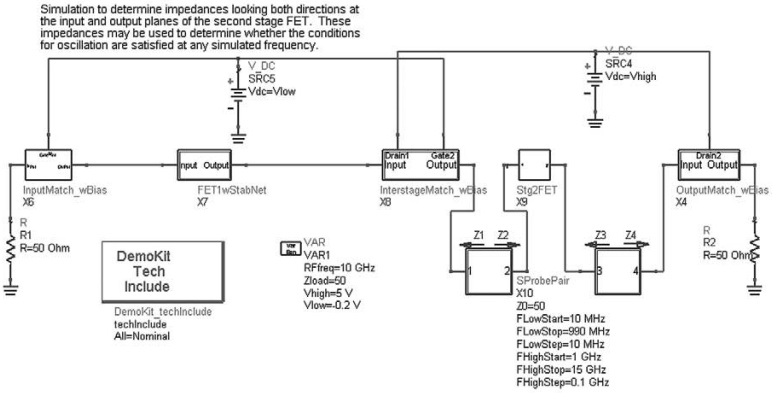


Figure 23. Two-stage amplifier output schematic.

The data display in Figure 24 shows that the load impedance is close to the desired value at 10 GHz, and that the stability conditions are satisfied from 10 MHz to 20 GHz. (It is necessary to check stability conditions over a broad frequency range, beyond the operating band of interest, to check for undesired potential oscillations.) Also, the source impedance presented to the input of the device is close to the complex conjugate of the device's input impedance at 10 GHz.

16. Two-stage amplifier gain compression

Next, a swept-power simulation of the two-stage amplifier with matching networks was carried out to determine the maximum output power that could be supplied, power-added efficiency, 1-dB compression point, etc. This simulation is TwoStgAmp_TB, as shown in Figure 25, and indicates a maximum output power of about 26.6 dBm and an output power at the 1-dB gain compression point of about 25 dBm. This simulation setup and data display are from the Amplifier DesignGuide (*DesignGuide > Amplifier > 1-Tone Nonlinear Simulations > Spectrum, Gain, Harmonic Distortion vs. Power (w/PAE)*). There are many other simulation setups in the Amplifier DesignGuide, so you could look at things like intermodulation distortion, 1-dB gain compression, frequency response, and responses versus swept parameters as well.

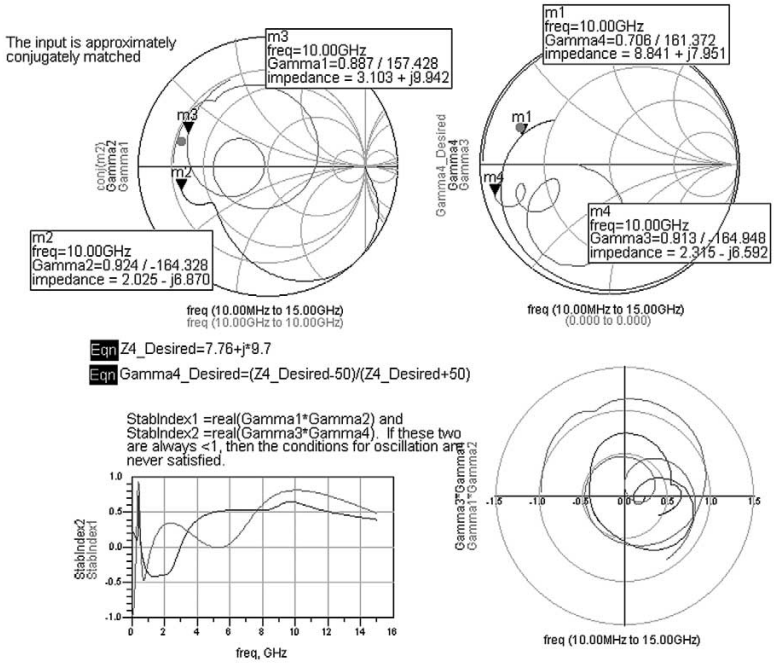


Figure 24. Source and load impedances close to the desired values are being generated for the second-stage device, and the stability conditions are satisfied.

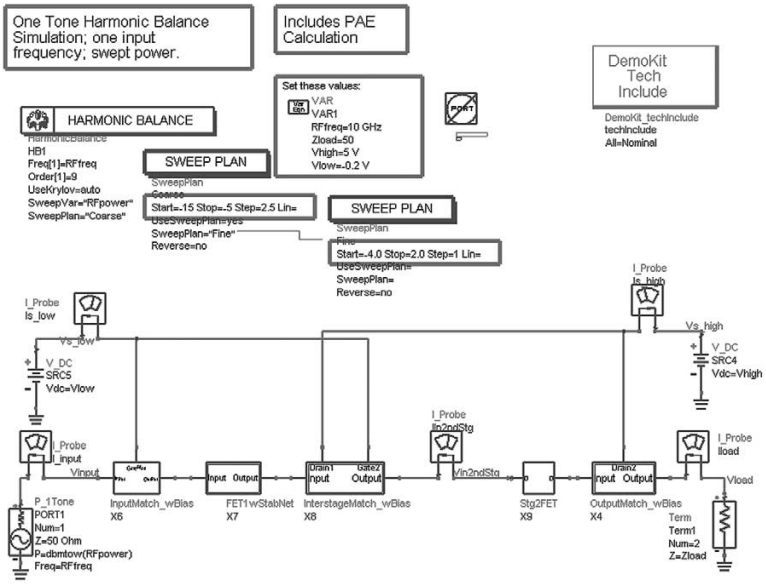


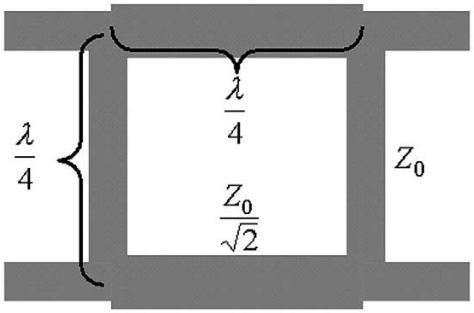
Figure 25. Simulating the gain compression and power-added efficiency of the two-stage amplifier.

Branch-Line Coupler Design

Branch-line couplers are used at the input and output, to split the signal to be sent through two parallel, identical two-stage amplifiers and then to recombine the signals at the output. One of the advantages of this approach is that the input and output matches of the overall amplifier are good, even though the two-stage amplifier may be mismatched at the input or output. Also, you potentially can obtain 3 dB higher output power than a single amplifier could provide by itself.

Branch-line couplers may be implemented via quarter-wavelength transmission lines as shown in Figure 26. But at 10 GHz, these lines would be 2-3 millimeters long. So instead, the transmission lines are replaced by C-L-C pi networks, as shown in Figure 26. The values for the inductors and capacitors are given by the equations in the figure.

Exact values for the Ls and Cs are computed on the BLC_LumpedIdeal schematic, shown in Figure 27. (It is useful to have an ideal branch-line coupler, because it can be used to determine the best performance that the amplifier is capable of achieving, and to determine whether time and effort should be expended on improving a physical branch-line coupler design or on the two-stage amplifier.) These Ls and Cs were converted to design kit components, with a resulting degradation in performance.



For Z_0 sections:
$L = Z_0 / \omega$,
$C = 1/(\omega Z_0)$

For $Z_0/\sqrt{2}$ sections:
$L = Z_0 / (\omega\sqrt{2})$,
$C = \sqrt{2}/(\omega Z_0)$

At 10 GHz, λ/4 is too long (2-3 mm.) So use lumped element pi network in place of each λ/4 section.

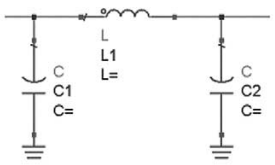


Figure 26. The Branch-line coupler, implemented as transmission lines, can be implemented using a π-network equivalent circuit for each $\lambda/4$ section, using the equations shown [Reference, "Foundations for Microstrip Circuit Design," T.C. Edwards, John Wiley and Sons, 1981, pg. 10.]

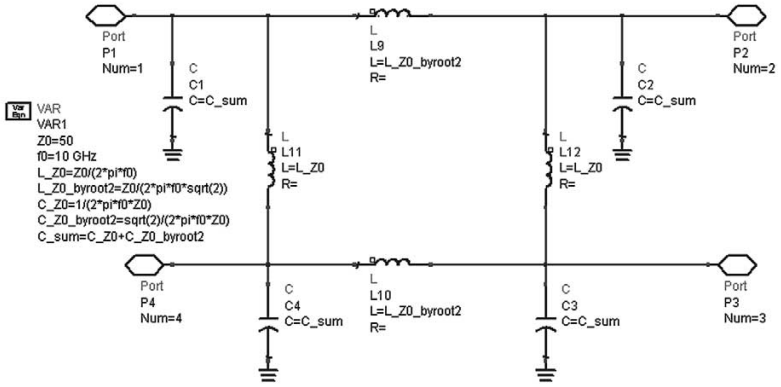


Figure 27. A branch-line coupler implemented using ideal, lumped elements.

A discrete-value optimization was run (setup is BranchLineCoupDiscOpt) as shown in Figure 28, to improve the performance of the branch-line coupler circuit implemented with design kit components.

The simulation results are shown in the BLC_Lumped_TB data display, as shown in Figure 29.

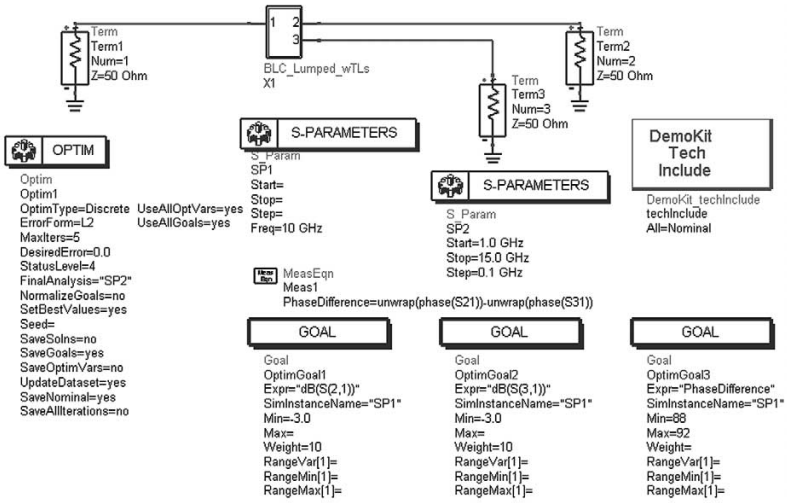


Figure 28. Setup for optimizing the branch-line coupler performance. The optimizable variables are defined in the subcircuit, as shown in Figure 30.

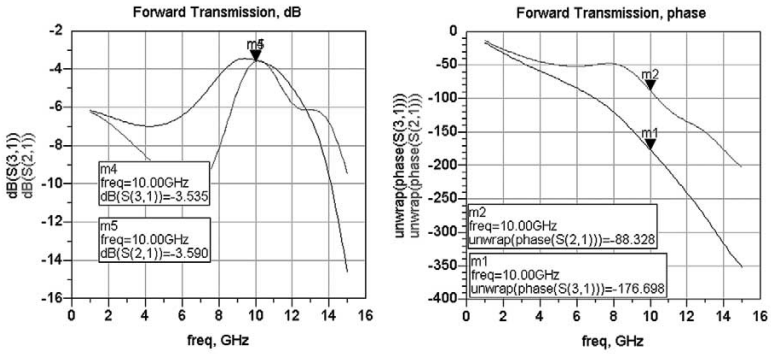


Figure 29. Optimized branch-line coupler performance.

The optimized parameter values are shown on the BLC_Lumped schematic, shown in Figure 30. The BLC_LumpedBk_to_Bk_TB is used to determine the insertion loss as well as the overall frequency response of two branch-line couplers connected back to back. Ideally, this loss would be 0 dB, but the actual loss will reduce the ideal 3-dB increase in output power that this balanced amplifier configuration would achieve if these were lossless. You can compensate for loss due to the input branch-line coupler by increasing the input signal power, but you cannot make up for the loss due to the output coupler.

Preliminary balanced amplifier performance (without including interconnect parasitics)

Combining the two-stage amplifiers and lumped-element branch-line couplers together, we get a balanced amplifier. This is simulated in BalancedLumpedAmp_TB, and shown in Figure 31. This is the same simulation setup from the Amplifier DesignGuide that was used to evaluate the two-stage amplifier. The results show a saturated output power of about 29 dBm, and an output power at the 1-dB gain compression point of about 26.5 dBm. These results are with design kit components, but without including any transmission line effects.

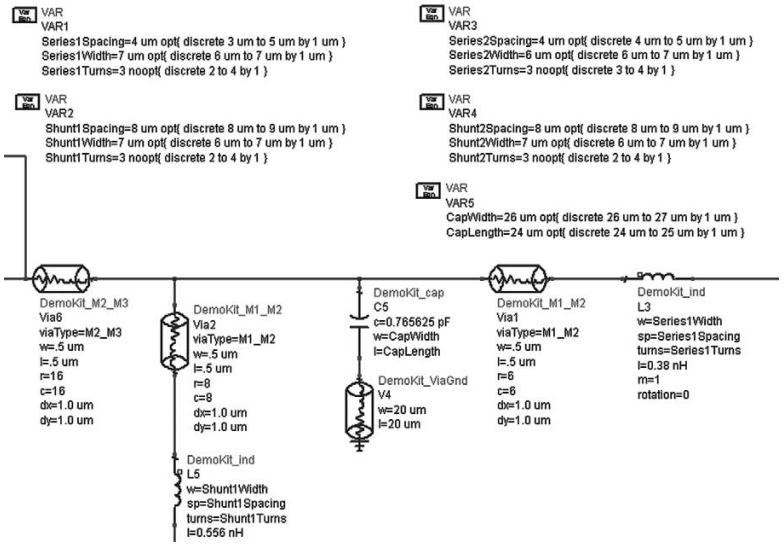


Figure 30. Optimized branch-line coupler parameter values.

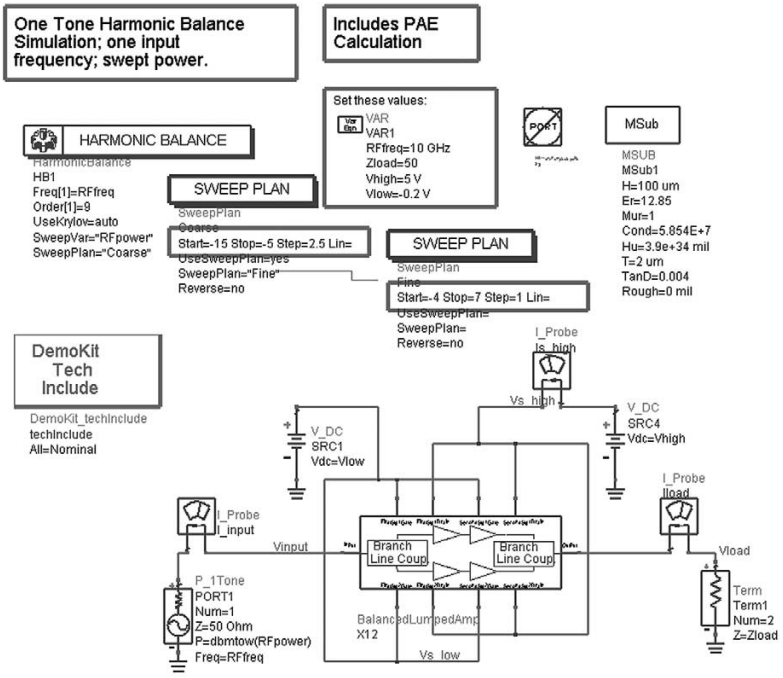


Figure 31. Simulating the gain, power, and power-added efficiency of the preliminary balanced amplifier.

Creating the layout

The layout of each subcircuit was done by initially placing design kit elements in the schematic, then using the *Layout > Place Components From Schem To Layout* command to manually place the components into the layout. A faster alternative is the *Layout > Generate/Update Layout* command, which will automatically place all of the schematic components into the layout. After components were placed in the layout, traces were inserted (*Insert > Trace* command, or selecting the toolbar icon) to connect them together.

LineCalc was used to determine that a 70 µm width was needed for a 50 Ohm line on 100 µm GaAs, and that a 20-µm wide line is about 77 Ohms. For RF interconnects, these transmission line lengths are kept as short as possible to minimize parasitics.

In the layout, vias were inserted where it was necessary to change from one metal layer to another. These were then placed in the schematic via the *Schematic > Place Components From Layout To Schem* command.

The *Tools > Check Representation* command, shown in Figure 32, is quite useful for verifying that all components have been placed in both the layout and the schematic and that their parameter values all match.

Errors are reported as shown in Figure 33. When placing the subcircuits in the top-level layout, the *Edit > Edit In Place > Push Into* command is quite useful for aligning the pins of different subcircuits for final connections as well as for adjusting the placement of components to eliminate overlaps and minimize wasted space.

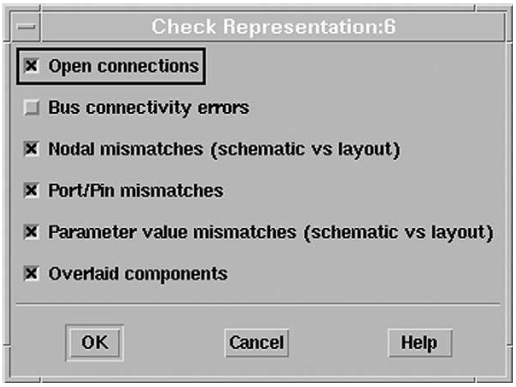


Figure 32. Check representation dialog box.

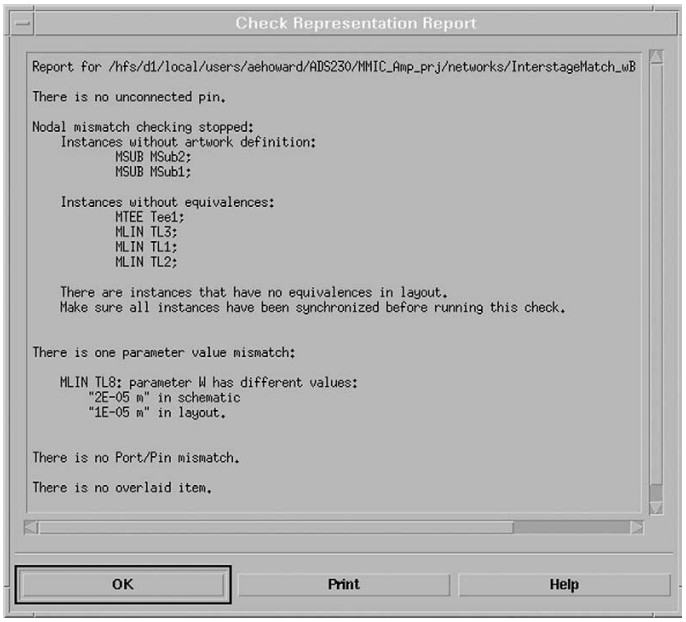


Figure 33. Check representation error display.

An initial layout of the complete amplifier is shown in BalancedLumpedAmp layout, Figure 34. Since the effects of transmission lines have not been included in the simulations yet, it is expected that some adjustments to this layout will be necessary.

Modeling transmission line effects – converting traces to transmission line elements

To include the effects of transmission lines, the traces in the layout (which are simulated as short circuits) must be converted to transmission line elements. To do this, a copy of each subcircuit was created, adding the suffix "wTLs" to the design name (although this is not necessary, it makes it easier to compare the circuit performance with and without including transmission line effects.) An MSUB component for each metal layer on which transmission line effects were to be included was placed into the schematic. Each MSUB component would have a different metal layer name for its Cond1 parameter. In this design, all RF traces are on either Metal1 or Metal2, so two MSUB components have been inserted. (If traces on Metal3 are to be modeled as transmission lines, also, then a third MSUB component is necessary.) Some real design kits might have pre-defined MSUB components; one for each metal layer.

In each layout, to convert all traces on a particular layer, do a *Select > Select All On Layer...* Then do an *Edit > Path/Trace/Wire > Convert Traces...* In the Trace Control dialog box, shown in Figure 35, set Convert Trace to Transmission line elements and enter the MSUB Element ID (from the schematic) under Substrate References. When this command is executed, the traces in the layout are converted to transmission line elements.

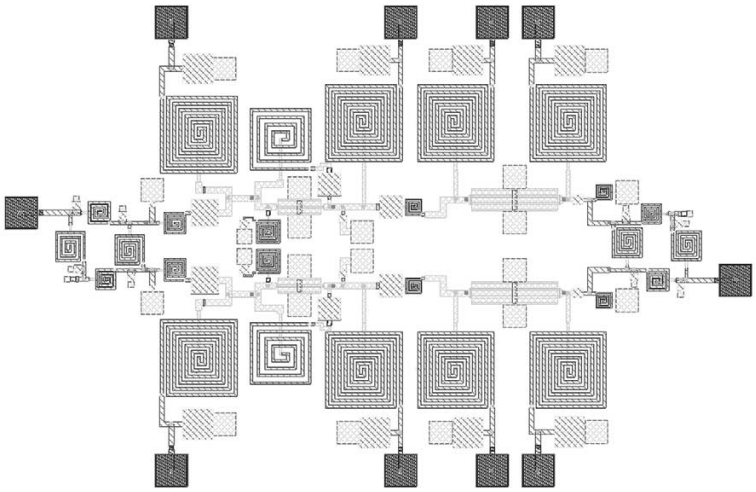


Figure 34. The BalancedLumpedAmp layout.

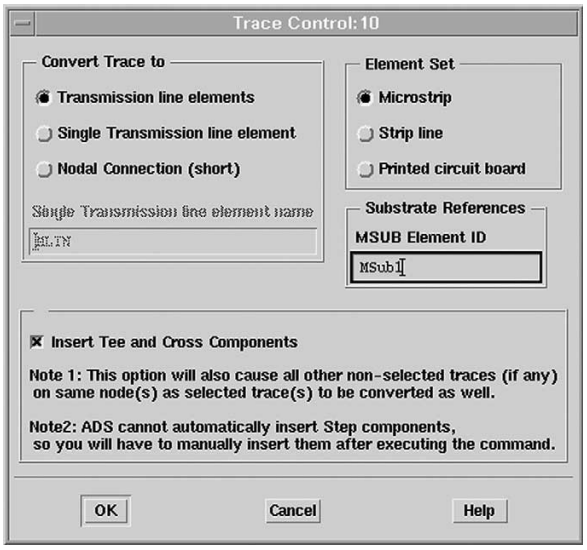


Figure 35. Trace control dialog box.

Figure 36 shows a section of a layout after traces have been converted to transmission lines. These transmission lines can then be placed in the schematic via the S*chematic > Place Components From Layout To Schem* command. There are a few ways to do this. First, you can manually place the components to create a schematic that looks similar to your layout, and it enables you to find key transmission lines and test the effects of varying them on performance. Another option is to use layout look-alike components to create schematic symbols that look the same as the passive parts of your layout. With this method, a Momentum simulation is automatically launched to generate a model for each look-alike component, but the Momentum simulation only has to be run once, as long as you don't modify the layout. This should give the most accurate simulation results.

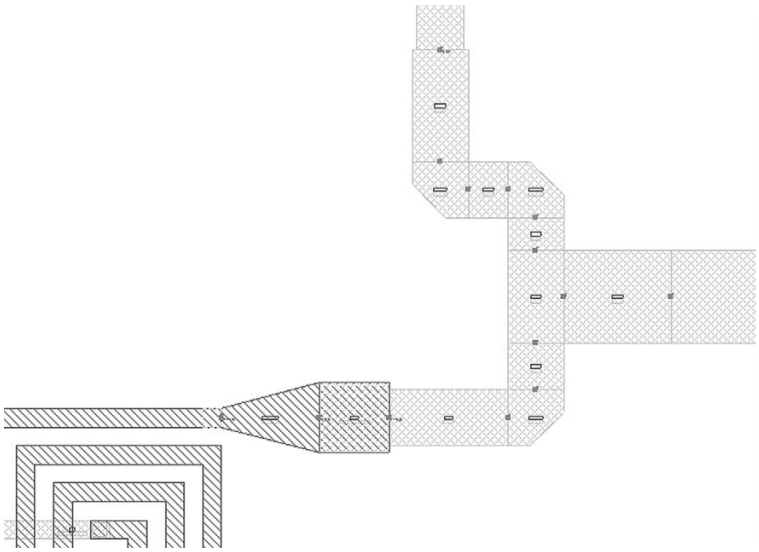


Figure 36. Part of a layout after converting traces to transmission lines.

Alternatively, you can skip placing the transmission line components onto the schematic and instead select Simulate from Layout in the Design Parameters dialog (*File > Design Parameters*), as shown in Figure 37.

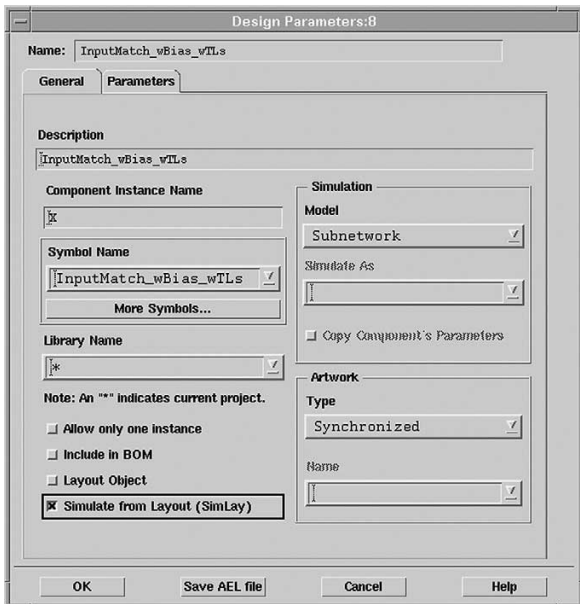


Figure 37. Design Parameters dialog box, allowing you to specify that the layout representation be used for the simulation.

Compensating for transmission line effects

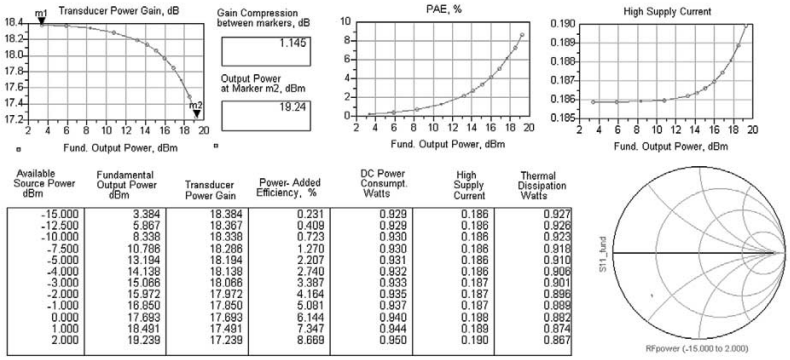
After the transmission lines have been included in the various subcircuits, it is necessary to check source and load impedances presented to each FET and re-run simulations to check for performance degradation. Initially, the overall performance of the two-stage amplifier degraded substantially, with output power dropping to less than 20 dBm, as shown in Figure 38.

For the input matching network, with transmission line effects included, reducing the design kit series inductance from 0.424 nH to 0.39 nH and reducing the shunt capacitance from 0.3825 pF to 0.32 pF brought the source impedance back to the desired value.

In the InterstageMatch_wTLs schematic, reducing the series inductance from 0.38 nH to 0.32 nH, to compensate for the inductance of the transmission lines brought the source and load impedances back to the desired values. (It was discovered experimentally that matching for gain will not necessarily give you the maximum 1-dB gain compression output power, so the load presented to the first stage device is not optimal for gain.) Also, increasing the widths of several of the transmission lines reduced the amount of parasitic inductance they contribute. The output matching circuit was adjusted similarly. This sort of "tuning" to compensate for parasitics could be much more complicated in a broadband design, and optimization would be a good tool to apply in that case.

The branch-line coupler circuit was re-optimized with the interconnect traces modeled as transmission lines, using the BranchLineCoupDiscOpt discrete value optimization setup. The optimizer found that for optimal performance, the inductors in the two series arms were unequal and the inductors in the two shunt arms were unequal as well.

The results of the optimization are shown in Figure 39.



Available Source Power dBm	Fundamental Output Power dBm	Transducer Power Gain	Power- Added Efficiency, %	DC Power Consumpt. Watts	High Supply Current	Thermal Dissipation Watts
-15.000	3.384	18.384	0.231	0.928	0.186	0.927
-12.500	5.867	18.367	0.409	0.929	0.186	0.926
-10.000	8.338	18.338	0.723	0.930	0.186	0.923
-7.500	10.786	18.286	1.270	0.930	0.186	0.918
-5.000	13.194	18.194	2.207	0.931	0.186	0.910
-4.000	14.138	18.138	2.740	0.932	0.186	0.906
-3.000	15.066	18.066	3.387	0.933	0.187	0.901
-2.000	15.972	17.972	4.164	0.935	0.187	0.896
-1.000	16.850	17.850	5.081	0.937	0.187	0.889
0.000	17.693	17.693	6.144	0.940	0.188	0.882
1.000	18.491	17.491	7.347	0.944	0.189	0.874
2.000	19.239	17.239	8.669	0.950	0.190	0.867

Figure 38. The two-stage amplifier performance results, with interconnect traces modeled as transmission lines.

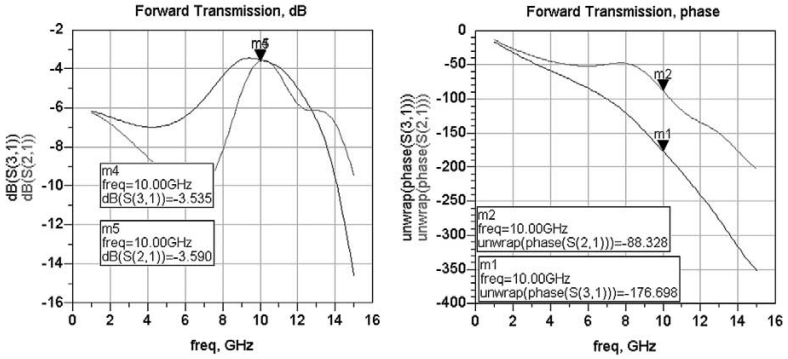
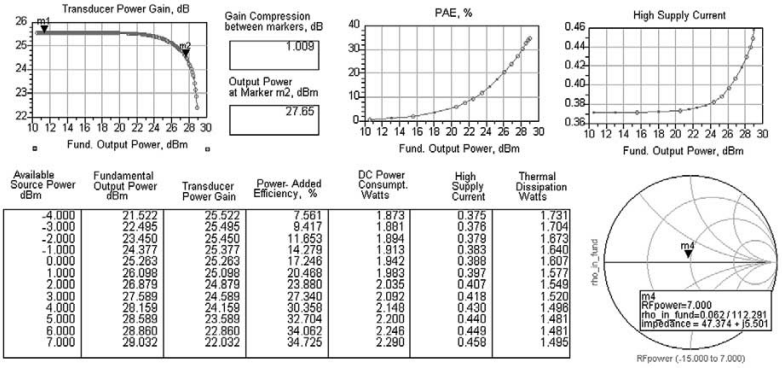


Figure 39. Branch-line coupler re-optimization results, including transmission line effects.

Amplifier performance including transmission line effects

The simulation results of the BalancedLumpedAmp_wTLs are shown in Figure 40. The amplifier has a saturated output power of > 29 dBm, and an output power at the 1-dB gain compression point of 27.7 dBm. The maximum power-added efficiency is about 35%.

Figure 41 shows the frequency response of the amplifier near 10 GHz. It has about a 2.8 dB noise figure at 10 GHz.



Available Source Power dBm	Fundamental Output Power dBm	Transducer Power Gain	Power- Added Efficiency, %	DC Power Consumpt. Watts	High Supply Current	Thermal Dissipation Watts
-4.000	21.522	25.522	7.561	1.873	0.375	1.731
-3.000	22.495	25.495	9.417	1.881	0.376	1.704
-2.000	23.450	25.450	11.653	1.894	0.379	1.673
-1.000	24.377	25.377	14.279	1.913	0.383	1.640
0.000	25.263	25.263	17.246	1.942	0.388	1.607
1.000	26.098	25.098	20.468	1.983	0.397	1.577
2.000	26.879	24.879	23.880	2.035	0.407	1.549
3.000	27.589	24.589	27.340	2.092	0.418	1.520
4.000	28.159	24.159	30.356	2.148	0.430	1.498
5.000	28.589	23.589	32.704	2.200	0.440	1.481
6.000	28.860	22.860	34.062	2.246	0.449	1.481
7.000	29.032	22.032	34.725	2.290	0.458	1.495

Figure 40. BalancedLumpedAmp_wTLs simulation results.

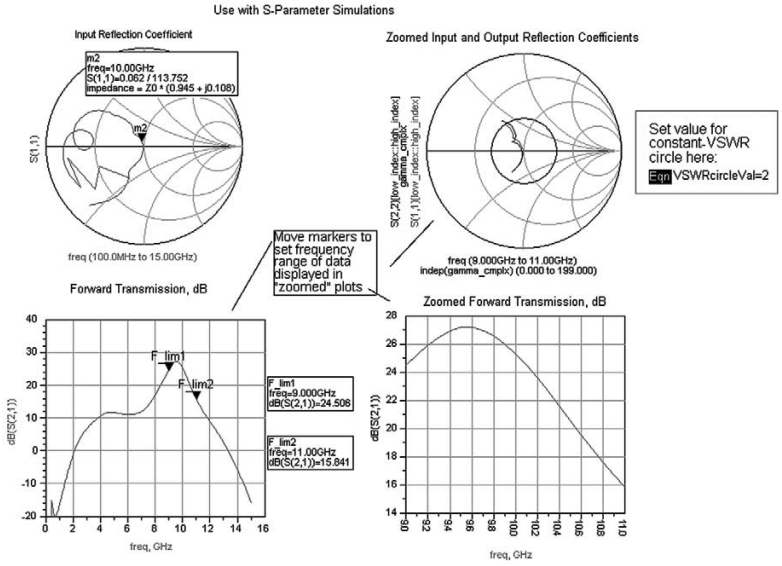


Figure 41. Amplifier frequency response near 10 GHz.

A more accurate simulation using 2.5-D electromagnetic simulation (Momentum)

Electromagnetic simulation can be used in the design process in several ways. One way is to verify that the equation-based models of components in the design kit are accurate. Actually, electromagnetic simulation could be used to create these models in the first place. Appendix A shows how to generate a library of more accurate inductor models, using the Advanced Model Composer.

Another use of EM simulation is to determine if undesired coupling between traces or components is degrading performance. In particular, the inductors in the branch-line coupler are very close together (closer to each other than they are to the ground plane) and coupling is likely to be significant.

Comparing the design kit inductor model with an EM simulation

The $HPEESOF_DIR/examples/MW_Ckts/MMIC_AmpEM_Sims_prj example (included in the ADS 2003A release) includes simulations of several design kit inductors with Momentum and compares the results with circuit simulations using the equation-based models from the design kit. The inductor was simulated in a series configuration, as shown in Figure 42.

The layout, after meshing for a Momentum simulation, is shown in Figure 43. The equations for S21 and its phase are:

$$S21 = \frac{2Z_0}{jwL + 2Z_0}$$

and

$$phase(S21) = \tan^{-1}\left(\frac{-wL}{2Z_0}\right)$$

[Reference: Guillermo Gonzalez, "Microwave Transistor Amplifiers, Analysis and Design," Prentice-Hall, 1984, pg. 19.]

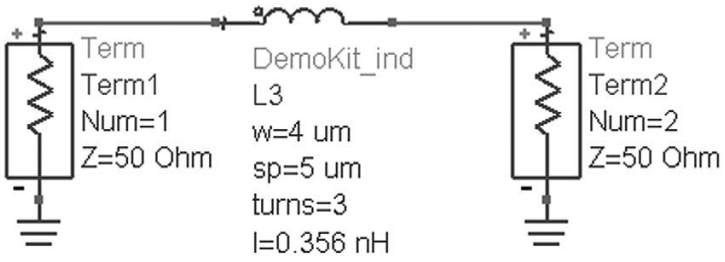


Figure 42. Simulating the S-parameters of a DemoKit inductor.

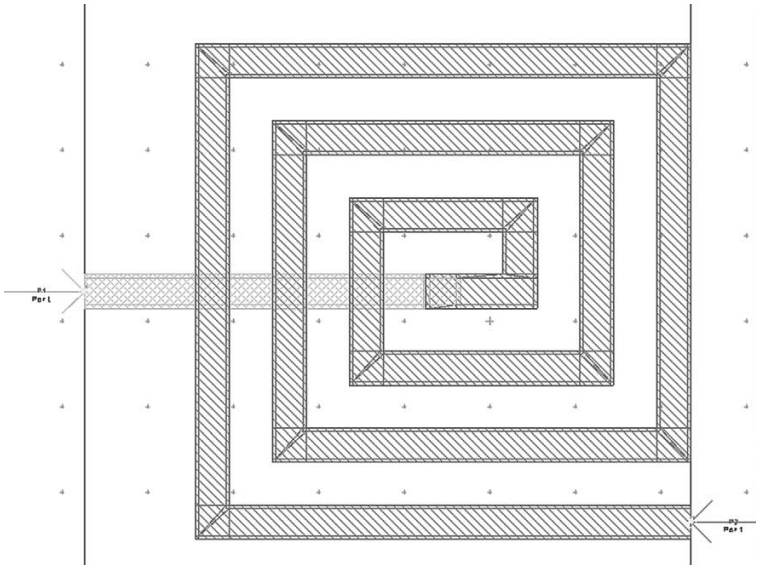


Figure 43. DemoKit inductor layout, including mesh for Momentum simulation.

Figure 44 compares the phases of an ideal 0.356 nH inductor with the equation-based model and a Momentum simulation. The plot indicates that the design kit's equation-based model actually behaves like a 0.394 nH inductor, and the Momentum simulation indicates that the inductor behaves like a 0.454 nH inductor, at least at 10 GHz. If this Demo design kit were to be used for designs on a real process, then it would be desirable to create more accurate models for critical components, particularly the inductors. A quick method of improving a particular circuit model would be to add parasitic elements until its response agrees with its Momentum simulation.

Using Momentum to simulate the branch-line coupler

Because the inductors in the branch-line coupler are so close together, degradation in performance due to unwanted coupling is expected. Momentum is used initially, to determine whether the degradation is acceptable or not. The capacitors in the Demo design kit are between the Metal1 and Metal2 layers. In a real process, there would probably be some intermediate metal between these two layers, to reduce the distance between the plates. Otherwise some high dielectric constant material would be needed to realize the capacitors. To simplify the Momentum simulation process, the layout without the capacitors is simulated with Momentum, a "look-alike" component is generated, and then a Circuit/EM co-simulation is run (re-using the Momentum results) to include the capacitors.

Figure 45 shows the layout that was simulated with Momentum.

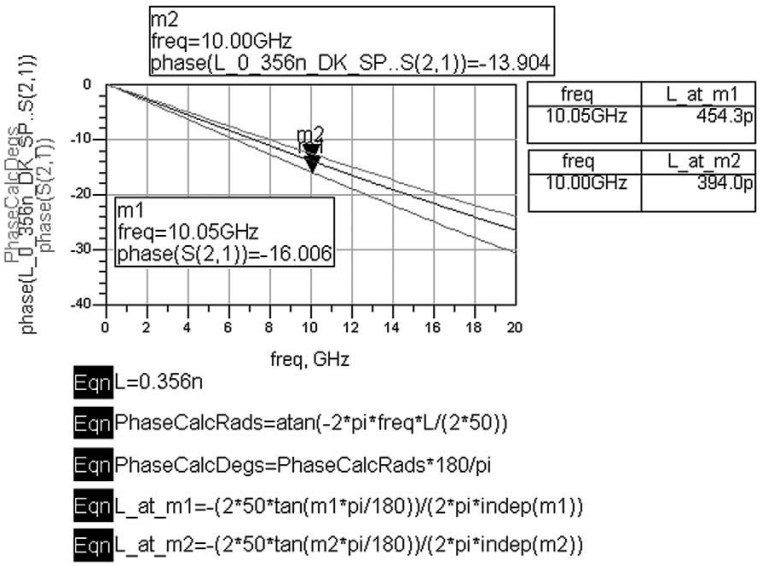


Figure 44. Comparing the phase shift of an ideal, DemoKit (equation-based), and Momentum-simulated inductors.

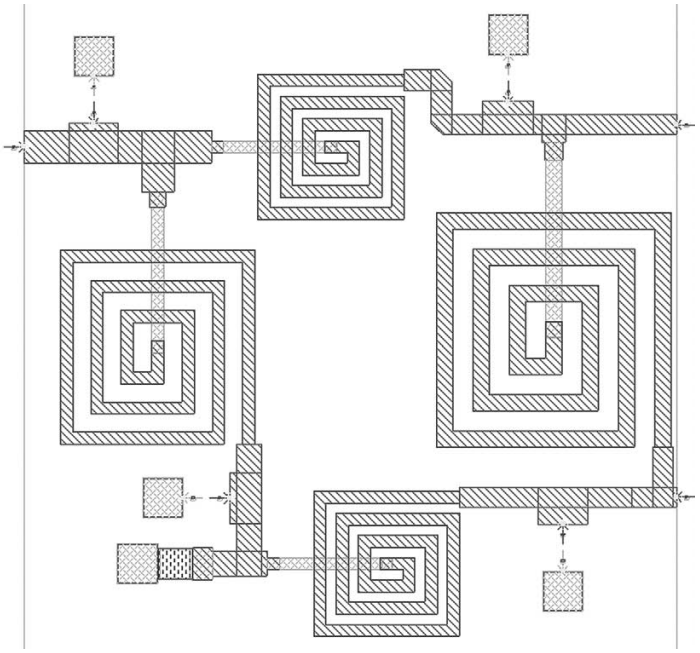


Figure 45. Branch-line coupler simulated using Momentum, after removing capacitors.

It has external ports for input and output signals and internal ports where the capacitors are to be connected. Besides removing the capacitors, the only other modification to the layout was to replace the Metal1-to-Metal2 via matrices with sheet vias, to improve the meshing efficiency. This is shown in Figure 46, with the original via shown on the left, and the modified via shown on the right.

Figure 47 shows the circuit simulation setup. It has a look-alike component for the branch-line coupler, which was simulated using Momentum RF, as well as equation-based models for the capacitors.

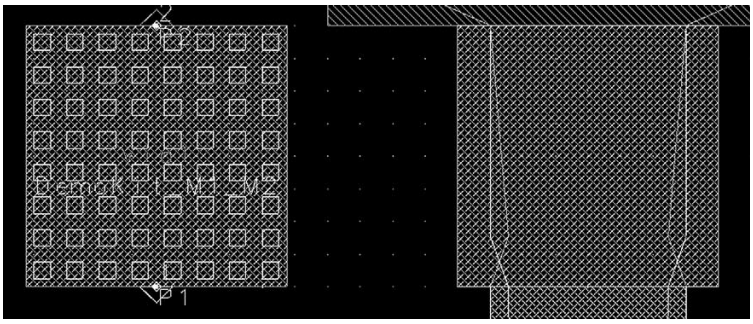


Figure 46. Original Metal1-to-Metal2 via structure at left, and after simplification for more efficient meshing, at right.

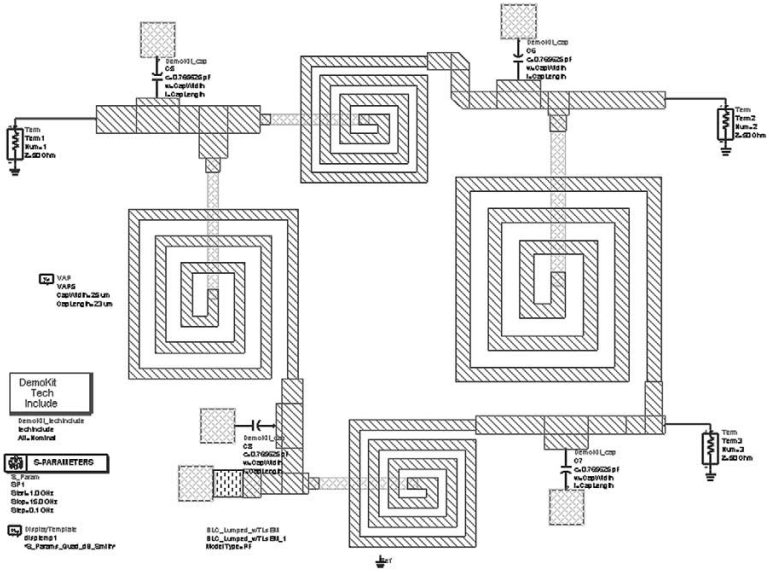


Figure 47. Reusing the Momentum simulation results in a circuit simulation of the branch-line coupler, which includes DemoKit capacitors.

Figure 48 shows that the Momentum-based simulation of the branch-line coupler has significantly higher insertion loss than the simulation shown in Figure 39 that used the Demo design kit inductor models (that are equation based.) This discrepancy in the simulation results is due to the equation-based inductor models not agreeing well with Momentum simulations of the same inductors.

With this EM-based model of the branch-line coupler, the balanced amplifier performance is degraded, with the 1-dB gain compression output power at only 25.5 dBm.

To improve the performance of the branch-line coupler, the Advanced Model Composer was run (as described in Appendix A) on the demo kit inductors to create a parameterized, more accurate, EM-based model. The parameters were width, spacing, and number of turns. Then, a circuit optimization, similar to what was described earlier, was run on the branch-line coupler, to improve its performance. This time, during the optimization, goals were added to minimize S(2,2) and S(3,3). The layout of the optimized branch-line coupler is shown in Figure 49.

Its performance is shown in Figure 50.

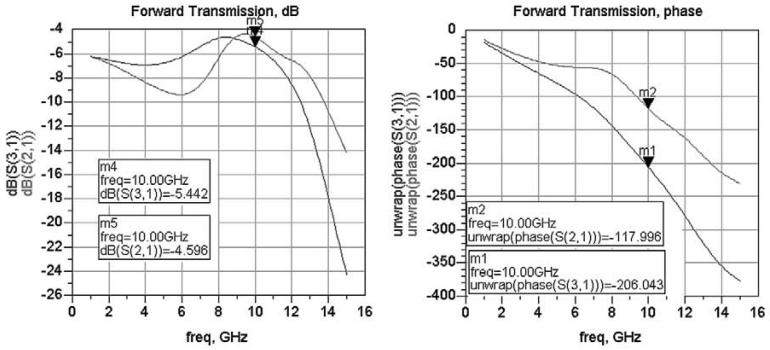


Figure 48. Momentum simulation results of the first branch-line coupler.

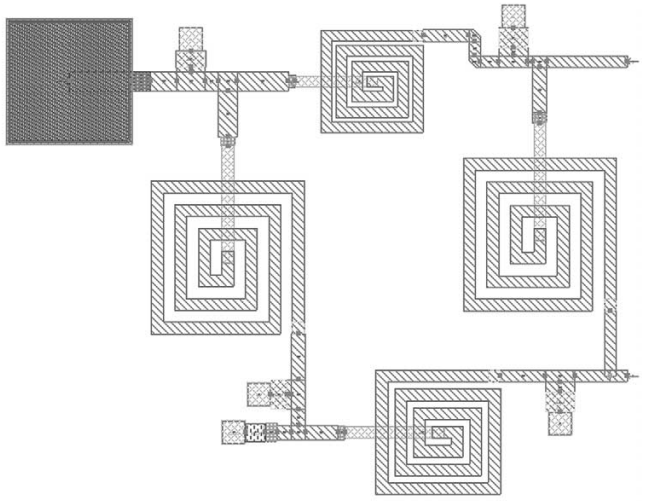


Figure 49. Layout of revised branch-line coupler, after optimizing inductor and capacitor parameter values, using a parameterized inductor model generated using the Advanced Model Composer.

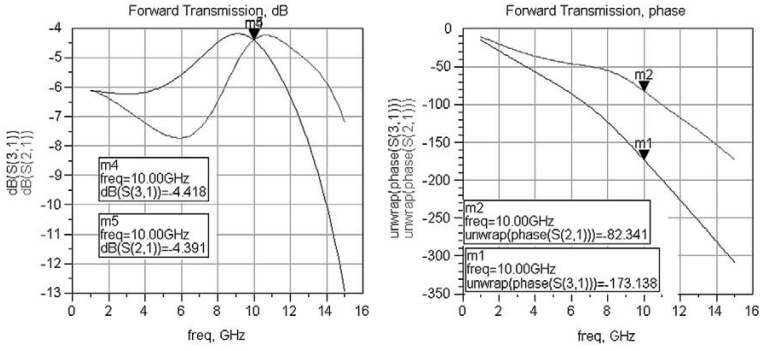


Figure 50. Simulation results of the revised branch-line coupler, after optimization.

The results from the Momentum simulation of this improved branch-line coupler are shown in Figure 51. The agreement between this simulation and the optimization results is now quite good.

The results of a simulation of the amplifier with the re-optimized branch-line coupler are shown in Figure 52. This used the layout look-alike component and Momentum simulation results for the branch-line coupler. The output power at the 1-dB gain compression point is about 27 dBm, and the saturated output power is about 28.3 dBm. While the output power at the 1-dB gain compression point meets the 0.5 Watt specification, there is no margin, so this specification would not be met over all temperatures and over variations in the manufacturing process. Changes to the design, such as using a larger device, a different bias point, a less lossy power combination circuit, or different impedance matching circuits would be necessary to achieve a design that could be manufactured with high yield.

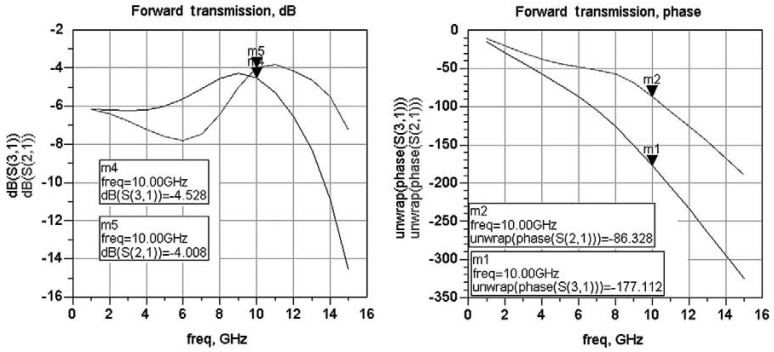
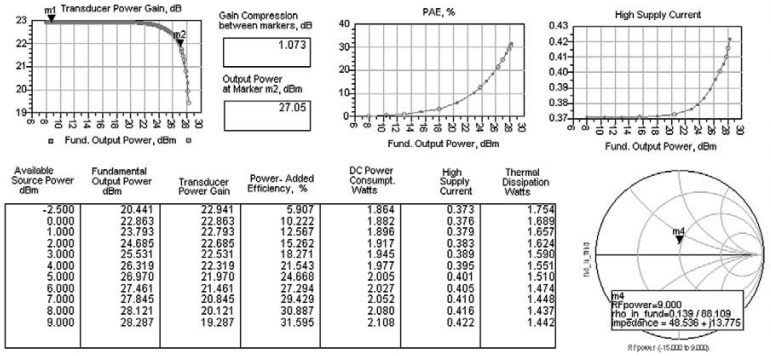


Figure 51. Momentum simulation results of the revised branch-line coupler.



Available Source Power dBm	Fundamental Output Power dBm	Transducer Power Gain	Power- Added Efficiency, %	DC Power Consumpt. Watts	High Supply Current	Thermal Dissipation Watts
-2.500	20.441	22.941	5.907	1.864	0.373	1.754
0.000	22.863	22.863	10.222	1.882	0.376	1.689
1.000	23.793	22.793	12.567	1.896	0.379	1.657
2.000	24.685	22.685	15.262	1.917	0.383	1.624
3.000	25.531	22.531	18.271	1.945	0.389	1.590
4.000	26.319	22.319	21.543	1.977	0.395	1.551
5.000	26.970	21.970	24.668	2.005	0.401	1.510
6.000	27.461	21.461	27.294	2.027	0.405	1.474
7.000	27.845	20.845	29.429	2.052	0.410	1.448
8.000	28.121	20.121	30.887	2.080	0.416	1.437
9.000	28.287	19.287	31.595	2.108	0.422	1.442

Figure 52. Performance of the balanced amplifier, including the revised branch-line coupler.

Conclusion

This application note demonstrates the use of ADS in a complete, front-to-back design process. ADS has numerous capabilities to facilitate MMIC design, including:

- Device evaluation (Gm versus bias)
- Load pull
- Impedance matching tools
- Tests for stability
- Linear and nonlinear simulations, including DesignGuide setups to simulate gain compression, power-added efficiency, intermodulation distortion, and numerous other characteristics of interest (only a few are shown in this application note)
- Discrete and continuous optimization
- Yield optimization (although not included in this application note)
- Use of foundry design kits
- Use of Advanced Model Composer to create models that can be simulated with the speed of standard circuit simulators yet provide the accuracy of EM simulation
- Physical design tools including a design rule checker and check representation, that checks for differences between a schematic and layout representation of a circuit
- Different ways of including parasitic transmission lines
- Electromagnetic simulation-to-model coupling

Appendix A – Using the Advanced Model Composer to Create a Library of Inductor Models

The Advanced Model Composer is a tool that allows you to create a library of multi-dimensional, parameterized, passive planar models for simulating arbitrary-shaped structures. Momentum, the 2.5-D electromagnetic simulator, is run repeatedly to generate these models. The models provide EM-level accuracy with the speed of an equation-based model. (The Model Composer, which is similar to the Advanced Model Composer, may be used for generating EM-based models of common, pre-defined shapes such as tees, bends, gaps, etc.) The Advanced Model Composer is covered in detail in the ADS manuals, under Momentum, "Chapter 6: Layout Components and Advanced Model Composer."

This appendix shows how to use the Advanced Model Composer to generate more accurate EM-based models of the demo design kit spiral inductors. It is easy to apply the Advanced Model Composer to parameterized layouts such as this spiral inductor, which has metal width, spacing, and number of turns as layout (and simulation) parameters.

Using the Advanced Model Composer on your own inductors

You can apply the Advanced Model Composer to your own spiral inductors, if they have parameterized layouts. If you don't have a parameterized layout, you can use one of the standard spiral models from the ADS layout library (TLines – Microstrip.) Or you can download a library of spiral inductor macros from the Agilent EEsof EDA technical support website (**www.agilent.com/find/eesof-support**, and search for spiral and transformer artwork macros.) Alternatively, you may use the ADS Layout Macro editor to build custom spiral inductors (refer to the Graphical Cell Compiler manual for step-by-step instructions.)

Creating a model of the DemoKit spiral inductor

We will create a new component by running a number of Momentum simulations, starting from a DemoKit spiral inductor layout. We will run the simulations over a defined range of parameter values.

Start ADS, open a project, and open a layout window.

Install the demo design kit, if it isn't already installed, so that you can place components from the DemoKit palette into the layout. Figure A-1 shows a spiral inductor from the demo design kit, with ports added.

We assign different parameters by selecting *Momentum* > *Component* > *Parameters...* Figure A-2 shows the dialog box after defining parameters W, SP, and N, the width, spacing, and number of turns, respectively. The Type has been set to Subnetwork, not Nominal/Perturbed, because all of the vertices change as these parameter values are changed. You must set each parameter's default value equal to the value used to create the layout as drawn for proper mapping between the defined parameter and the physical component.

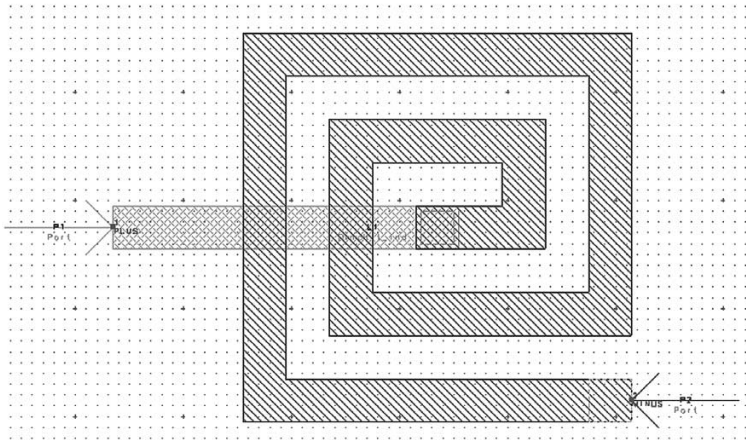


Figure A-1. A spiral inductor from the demo design kit.

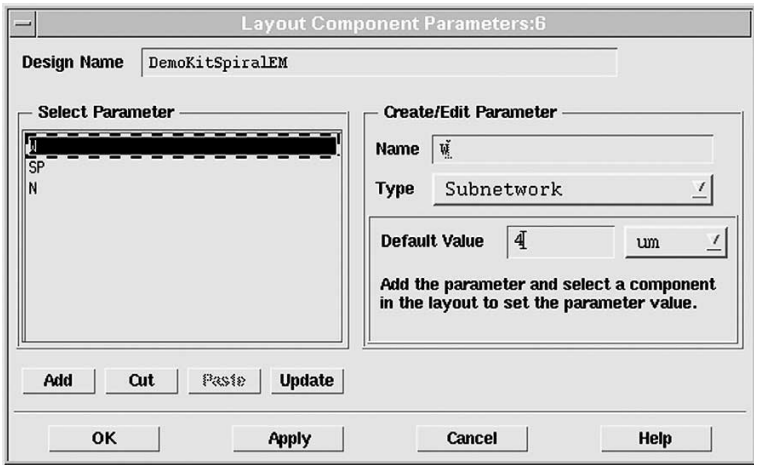


Figure A-2. Layout Component Parameters dialog box.

After defining the new component's parameters, you need to edit the component instance's parameter box and replace the initial, fixed parameter values with the new parameter names you have just defined. This is necessary to complete the mapping of the newly-defined parameters to those of the component's physical representation/compiled model. Figure A-3 shows the component's parameter dialog box (select the inductor, then do an *Edit* > *Component* > *Edit Component Parameters...*) after replacing the original parameter values with the newly-defined parameters. For example, the width, w, was originally set equal to 4 µm.

Now we are ready to generate a layout component, which is necessary to use the Advanced Model Composer. To do this, select *Momentum* > *Component* > *Create/Update...* The Create Layout Component dialog box is shown in Figure A-4. Set the simulation parameters in this dialog box as desired, and click OK. You should see a message window saying that the layout component was created successfully.

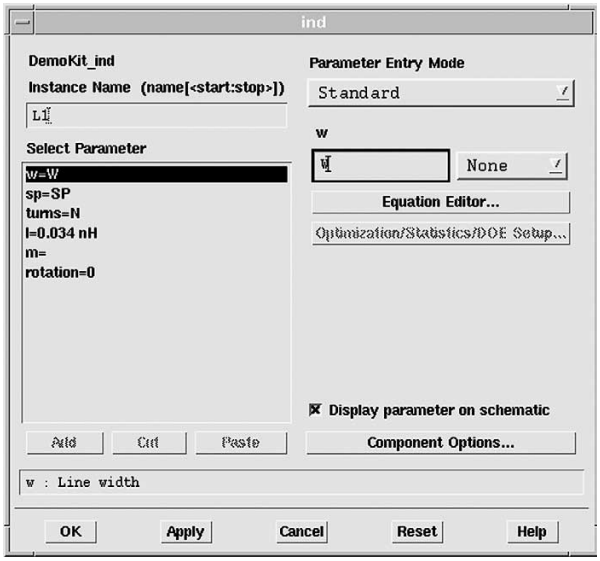


Figure A-3. Component parameter dialog box after replacing the original, fixed parameter values with newly-defined parameter variables.

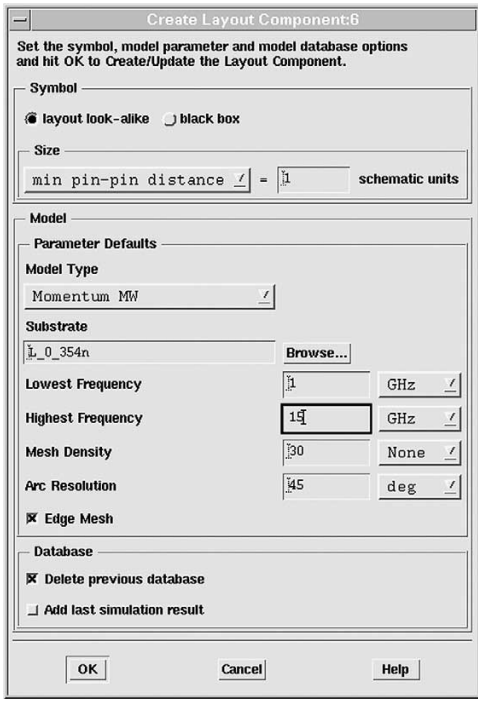


Figure A-4. Create Layout Component dialog box.

After assigning parameters and generating a new layout component, we can now specify the range of parameter values over which the Advanced Model Composer model will be generated, and start the process. Select *Momentum > Component > Advanced Model Composer > Create Model...* In the Advanced Model Composer –Create Model dialog box that appears, as shown in Figure A-5.

Enter simulation parameters as desired. Then select the *Layout Parameters* tab. Figure A-6 shows the settings after defining W, SP, and N each to be a discrete list of values. The width and spacing parameters could be defined as continuous variables, but the 1-um steps will keep the component on the layout grid, and this provides sufficient resolution.

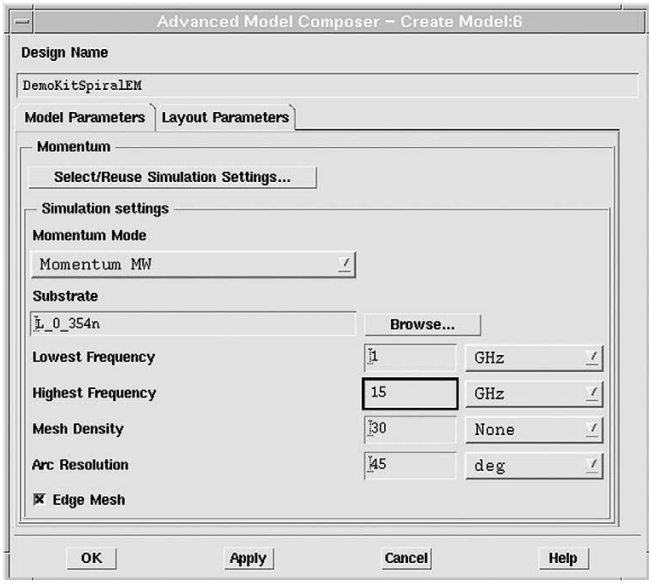


Figure A-5. Enter simulation parameters in the Advanced Model Composer – Create Model dialog box.

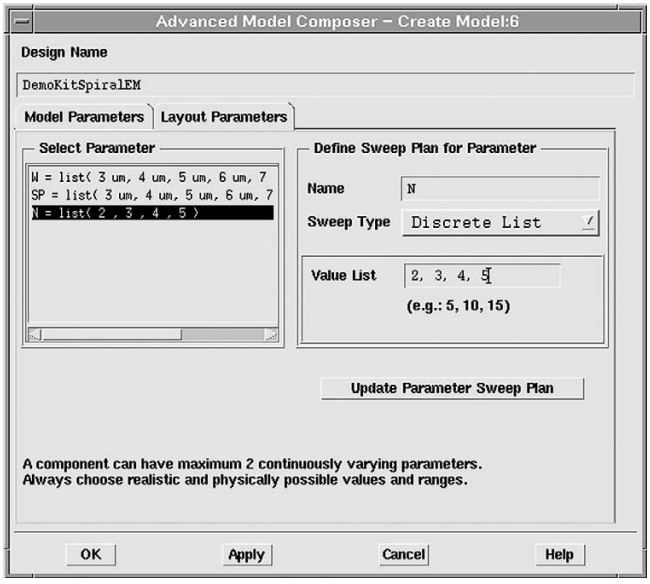


Figure A-6. Defining discrete lists of values for the width, spacing, and number of turns.

When you click *OK*, you will see the Start Model Composer dialog box, as shown in Figure A-7. The dialog explains that the model generation process will launch a separate ADS session in the background that may run for a while.

You can see the progress of the model generation process by selecting *Momentum* > *Component* > *Advanced Model Composer* > *Status/Control...* The window that appears, as shown in Figure A-8, shows that 1 hour, 8 minutes and 21 seconds were required to run the simulations and generate the model for a reduced range of parameter values.

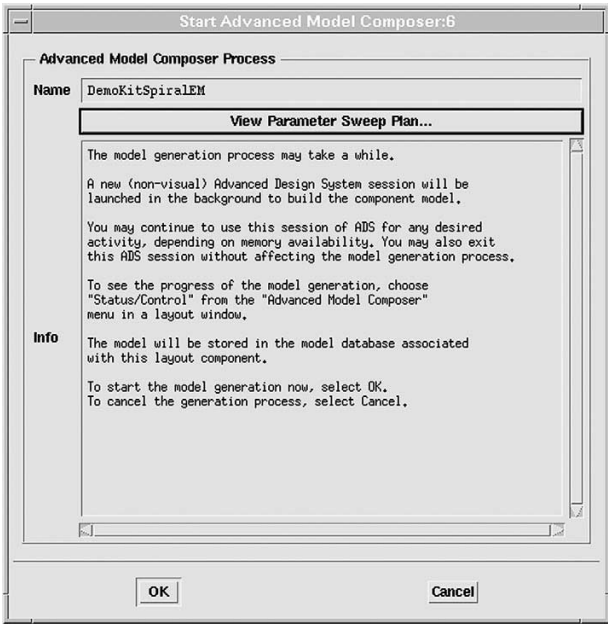


Figure A-7. Dialog box that appears just before you start the model generation process.

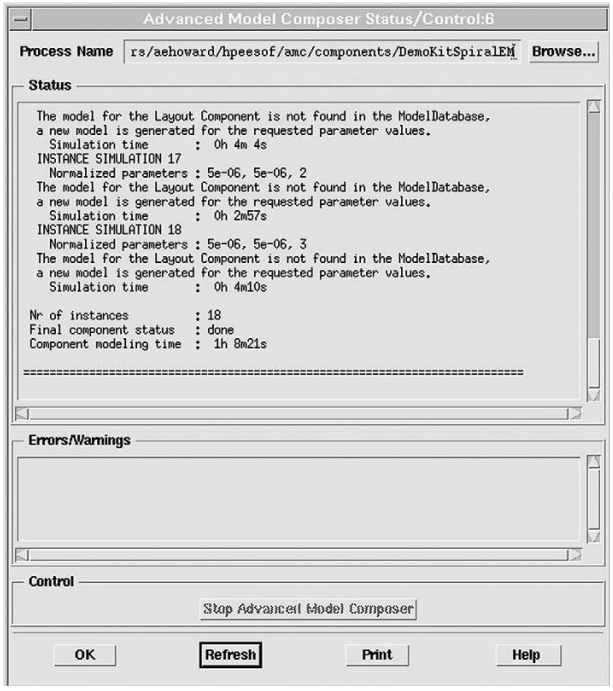


Figure A-8. Advanced Model Composer status dialog box.

If you select *Momentum > Component > Model Database...* you will see a list of files in the model database, as shown in Figure A-9. There is one .rat file for each combination of parameter values, and the .pml file is the final, compact model file. When a model file is selected, a description of the simulation settings appears under the Description field.

Now we can package the spiral inductor model into a new or existing design kit, in order to utilize the model in circuit simulations. Select *Momentum > Component > Advanced Model Composer > Design Kit...* to bring up the Create Layout Component Design Kit dialog box, as shown in Figure A-10. You may enter the component name and the design kit name as desired. Nothing will appear under the Components field until the design kit has been created. The design kit is stored in your $HOME/hpeesof/amc/design_kit directory.

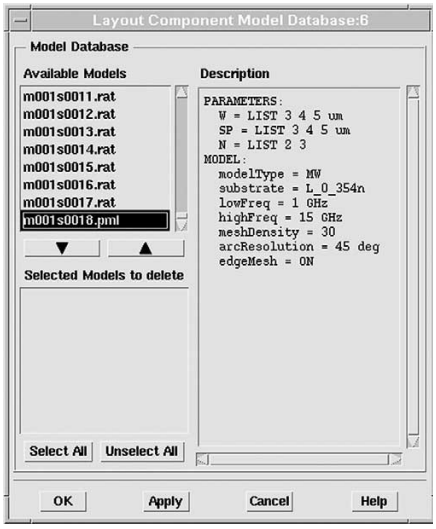


Figure A-9. Files in the model database.

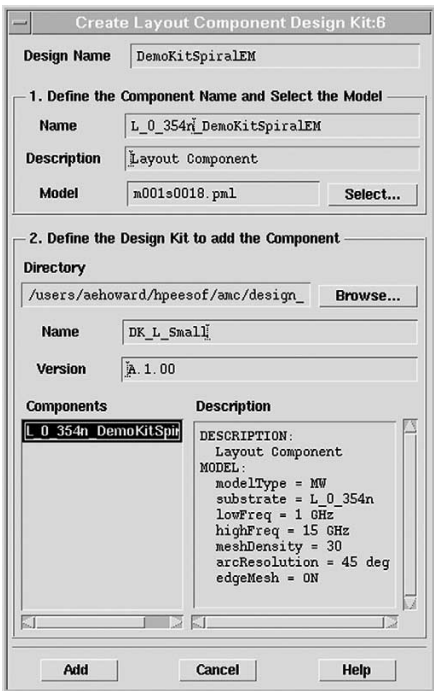


Figure A-10. Packaging the model into a design kit.

After the design kit is created, you must install it so you can use its models in circuit simulations. From the ADS main window, select *DesignKit* > *Install Design Kits...* Select the Browse... button and find the design kit in your $HOME/hpeesof/amc/design_kit directory. The filled dialog box is shown in Figure A-11. You will have to close your current project and reopen it for the design kit to become active.

To insert components from your newly-created design kit, you must use the library browser, as shown in Figure A-12.

Figure A-11. Dialog box for installing a design kit.

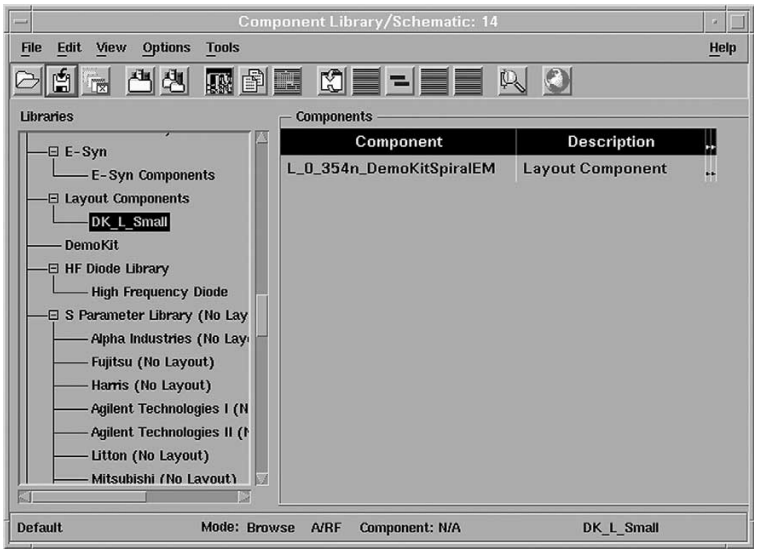


Figure A-12. Insert components from the design kit using the library browser.

Appendix B – Running the Design Rule Checker

The Demo design kit has a set of layout design rules, and they may be used to check layouts for violations. This appendix briefly shows the steps required to run the design rule checker. More detailed information about running the DRC, writing design rules, etc. is in the ADS Design Rule Checker manual.

Running the design rule checker

1. Copy the DRC rules from the Demo design kit (*$HPEESOF_DIR/examples/DesignKit/DemoKit/drc/rules/completedrc.ael*) into the project directory, under *verification/rules/*.

2. From a layout window, select *Tools* > *DRC: Custom Rules*. Figure B-1 shows the Custom DRC dialog box after browsing to find the rule file within the project directory.

3. Select Apply, and the rules file will be compiled. After they are compiled, the message window shown in Figure B-2 appears.

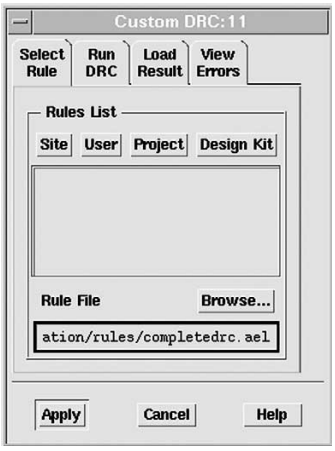


Figure B-1. Dialog box for running the DRC, after selecting a rules file.

Figure B-2. Dialog box, showing that the rules file has been compiled.

4. Select the Run DRC tab, and under Check Area, select Full Design or Current View Window, and then Apply. A message window similar to the one shown in Figure B-3 appears when the DRC is finished.

5. Load results by selecting the Load Result tab and then Apply. A message window similar to the one shown in Figure B-4 appears when the program is finished loading the results.

Figure B-3. Message that appears after the DRC has been run.

Figure B-4. Message that appears after the DRC results have been loaded.

6. Select the *View Errors* tab, then select Auto Zoom and other buttons (First, Last, Prev, or Next), as shown in Figure B-5, to view errors in the layout window.

7. Figure B-6 shows one of the errors — two bond pads less than 100 µm apart.

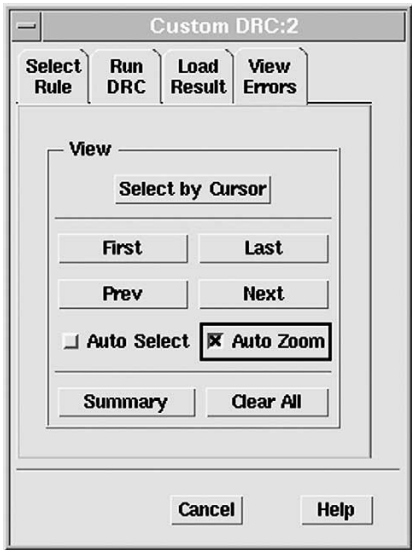


Figure B-5. Dialog box for viewing DRC errors.

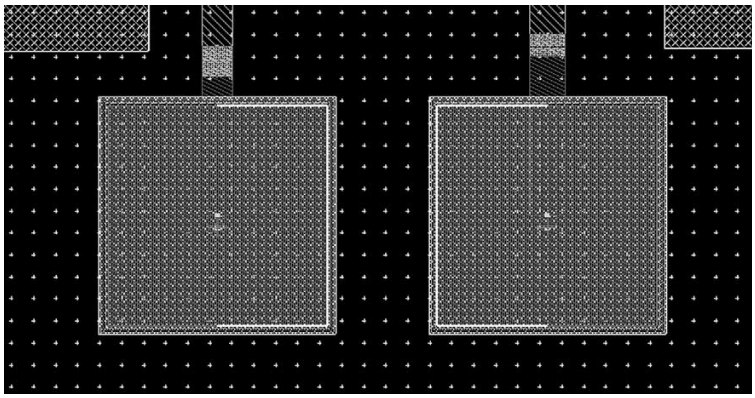


Figure B-6. A highlighted DRC error, showing two bond pads less than 100µm apart.

For more information about
Agilent EEsof EDA, visit:
www.agilent.com/find/eesof

For more assistance with your test and measurement needs, visit:
www.agilent.com/find/assist

Phone or Fax:

United States:
(tel) 1 800 452 4844

Canada:
(tel) 1 877 894 4414
(fax) (905) 282-6495

China:
(tel) 800 810 0189
(fax) 800 820 2816

Europe:
(tel) (31 20) 547 2323
(fax) (31 20) 547 2390

Japan:
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Korea:
(tel) (82-2) 2004-5004
(fax) (82-2) 2004-5115

Latin America:
(tel) (305) 269 7500
(fax) (305) 269 7599

Taiwan:
(tel) 0800 047 866
(fax) 0800 286 331

Other Asia Pacific Countries:
(tel) (65) 375-8100
(fax) (65) 6836-0252
Email: tm_asia@agilent.com

Product specifications and descriptions in this document subject to change without notice.

Printed in USA, June 17, 2003
5988-9637EN

Agilent Technologies