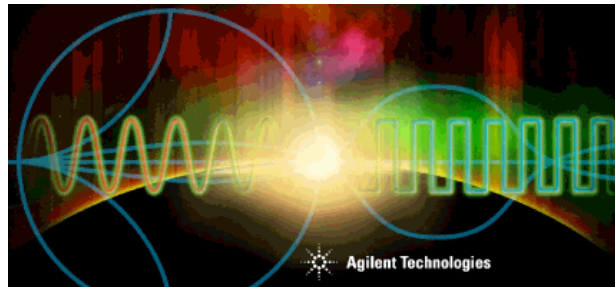


## Topic 5:

# S-parameter Simulation and Optimization



# S-parameters are Ratios

Usually given in dB as 20 log of the voltage ratios of the waves at the ports: incident, reflected, or transmitted.

## S-parameter (ratios): out / in

- S11 - Forward Reflection (input match - impedance)
  - S22 - Reverse Reflection (output match - impedance)
  - S21 - Forward Transmission (gain or loss)
  - S12 - Reverse Transmission (leakage or isolation)
- } Best viewed on a Smith chart.
- } These are easier to understand and simply plotted.

## Results of an S-Parameter Simulation in ADS

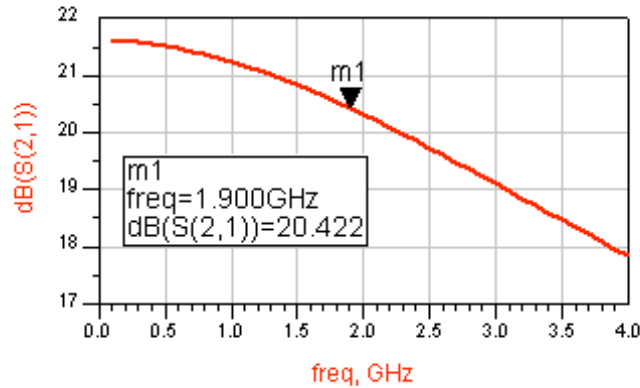
- S-matrix with all complex values at each frequency point
- Read the complex reflection coefficient (Gamma)
- Change the marker readout for Zo
- Smith chart plots for impedance matching
- Results are similar to Network Analyzer measurements

Next, ADS data

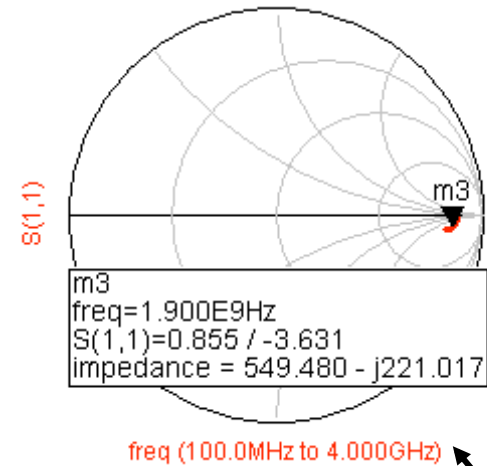


# Typical S-parameter data in ADS

**Transmission: S21 magnitude vs frequency**



**Reflection: S11 Impedance on a Smith Chart**



**Complete S-matrix with port impedance**

freq	S				freq	PortZ	
	S(1,1)	S(1,2)	S(2,1)	S(2,2)		PortZ(1)	PortZ(2)
100.0M...	0.879 / ...	2.392E-...	12.036 / ...	0.994 / ...	100.0M...	50.000 / ...	50.000 / ...
200.0M...	0.879 / ...	7.978E-...	12.020 / ...	0.993 / ...	200.0M...	50.000 / ...	50.000 / ...
300.0M...	0.878 / ...	1.729E-...	11.994 / ...	0.992 / ...	300.0M...	50.000 / ...	50.000 / ...
400.0M...	0.877 / ...	2.558E-...	10.019 / ...	0.994 / ...	400.0M...	50.000 / ...	35.000 / ...
500.0M...	0.877 / ...	4.007E-...	9.982 / ...	0.993 / ...	500.0M...	50.000 / ...	35.000 / ...
600.0M...	0.878 / ...	5.818E-...	9.939 / ...	0.991 / ...	600.0M...	50.000 / ...	35.000 / ...
700.0M...	0.875 / ...	8.013E-...	9.888 / ...	0.990 / ...	700.0M...	50.000 / ...	35.000 / ...

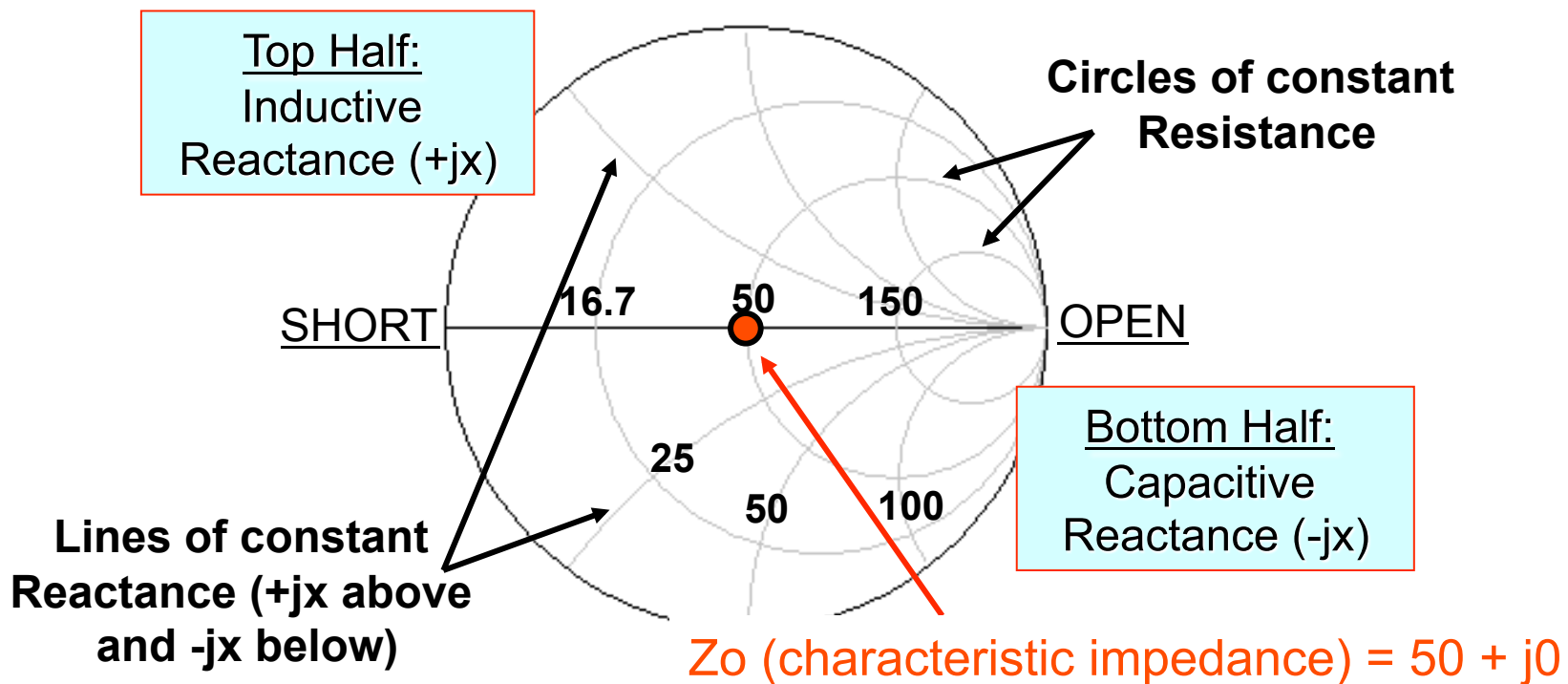
**Note: Smith marker impedance readout is changed to  $Z_0 = 50$  ohms.**

Smith chart basics...



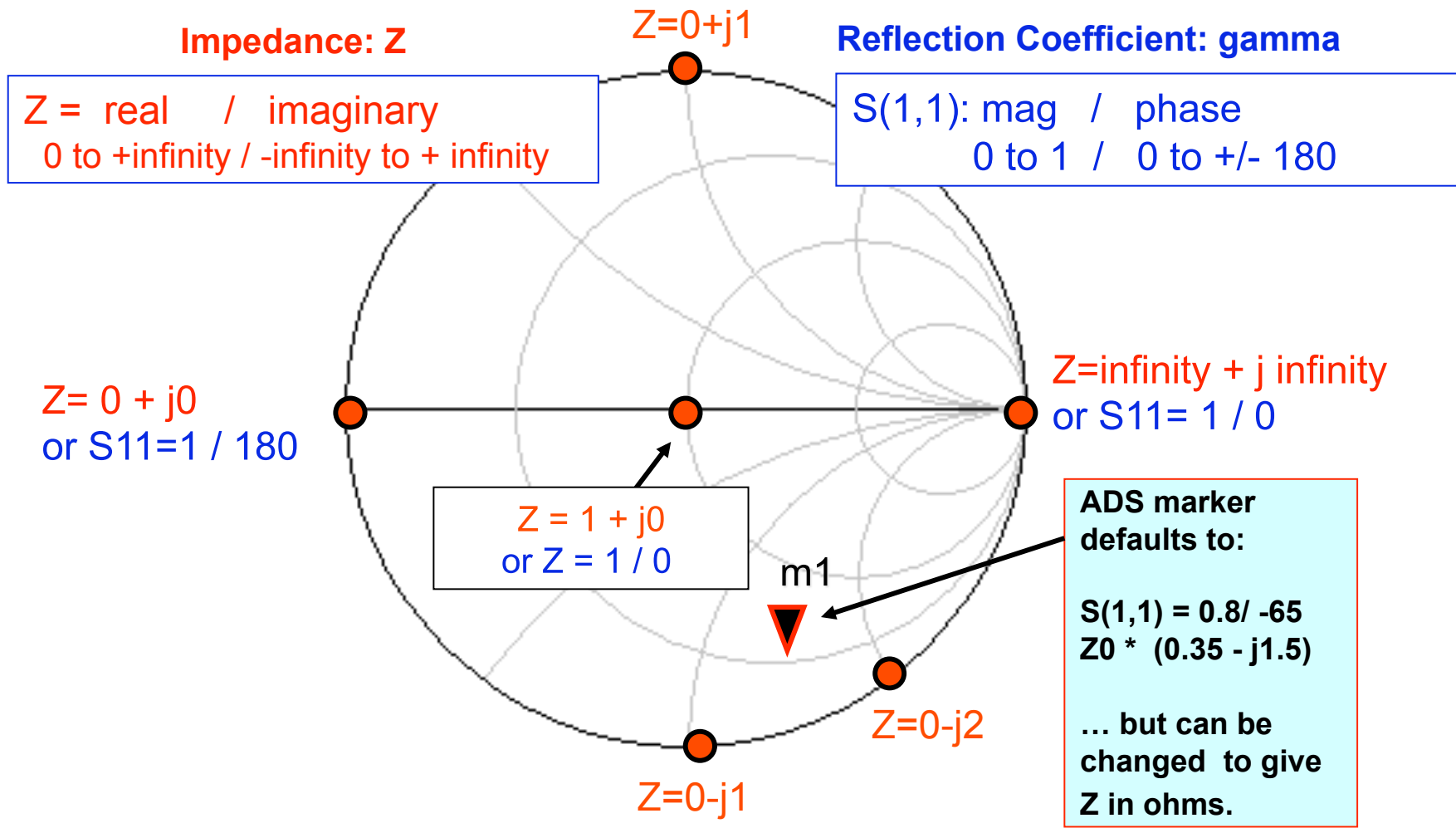
# The Impedance Smith Chart simplified...

This is an impedance chart transformed from rectangular  $Z$ . Normalized to 50 ohms, the center =  $R50+J0$  or  $Z_0$  (perfect match). For  $S_{11}$  or  $S_{22}$  (two-port), you get the **complex impedance**.



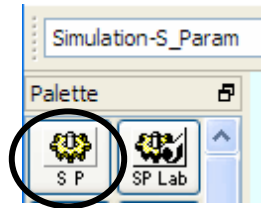
More Smith chart... 

# The Smith chart in ADS Data Display



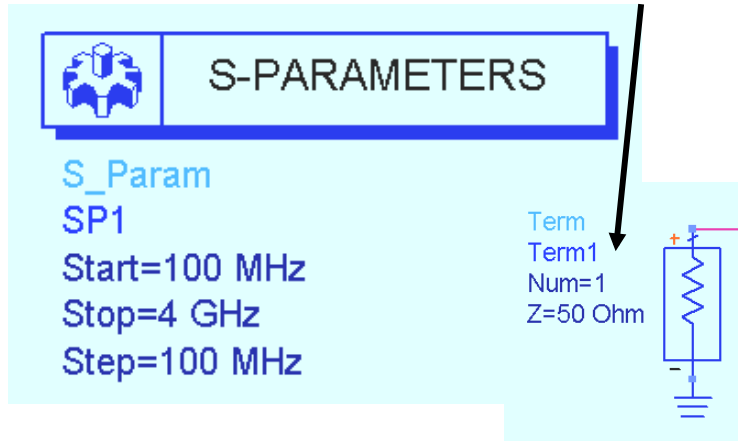
S-parameter controller... 

# S-Parameter Simulation Controller



Default sweep variable = **freq**

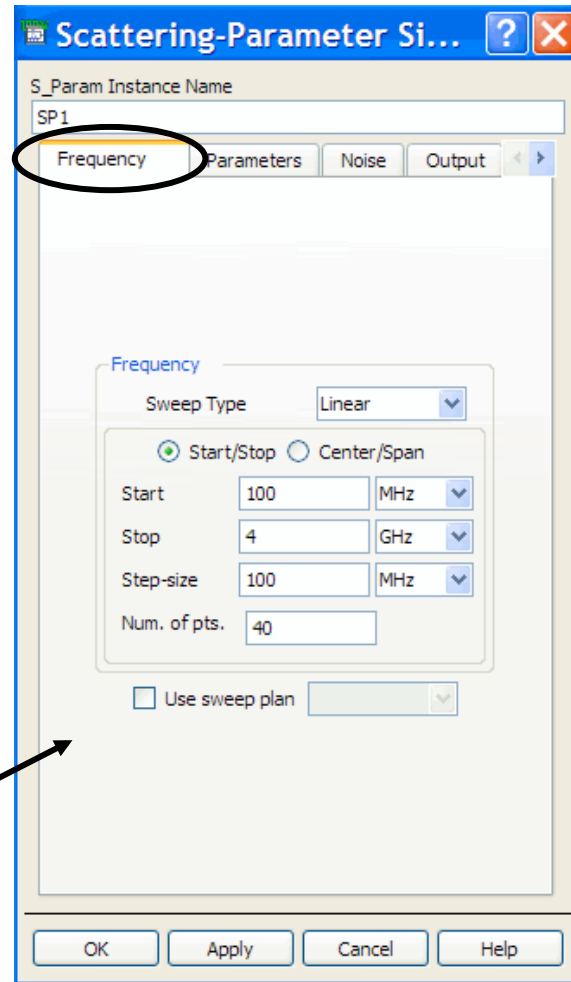
The simulator requires a port termination Num =



S-PARAMETERS

S\_Param  
SP1  
Start=100 MHz  
Stop=4 GHz  
Step=100 MHz

Term  
Term1  
Num=1  
Z=50 Ohm



Scattering-Parameter Si...

S\_Param Instance Name  
SP1

Frequency Parameters Noise Output

Frequency

Sweep Type: Linear

Start/Stop Center/Span

Start: 100 MHz

Stop: 4 GHz

Step-size: 100 MHz

Num. of pts.: 40

Use sweep plan

OK Apply Cancel Help

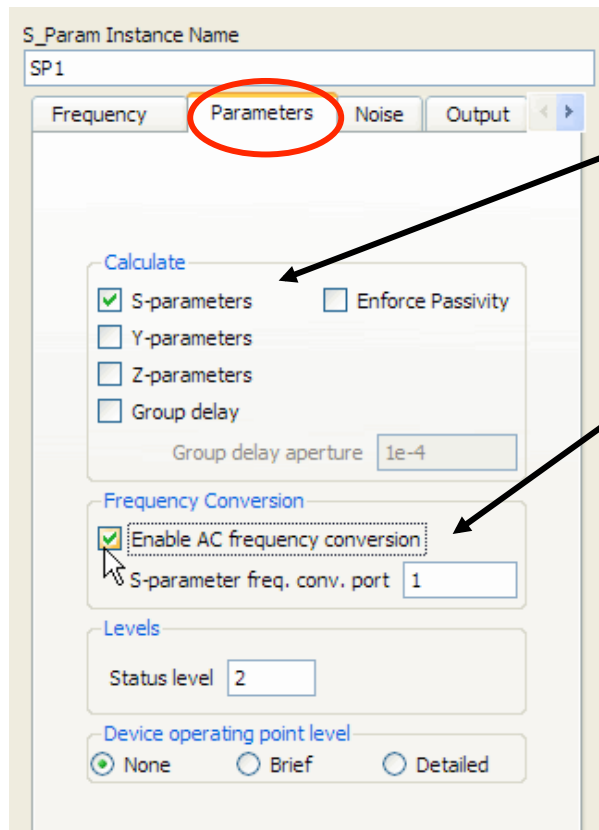
Sweep plan can also be used (see next slide). Either way, simulation data results in an S matrix in the data set for the specified range and points.

Next, other tabs



# Parameters and Noise tabs

## Parameters

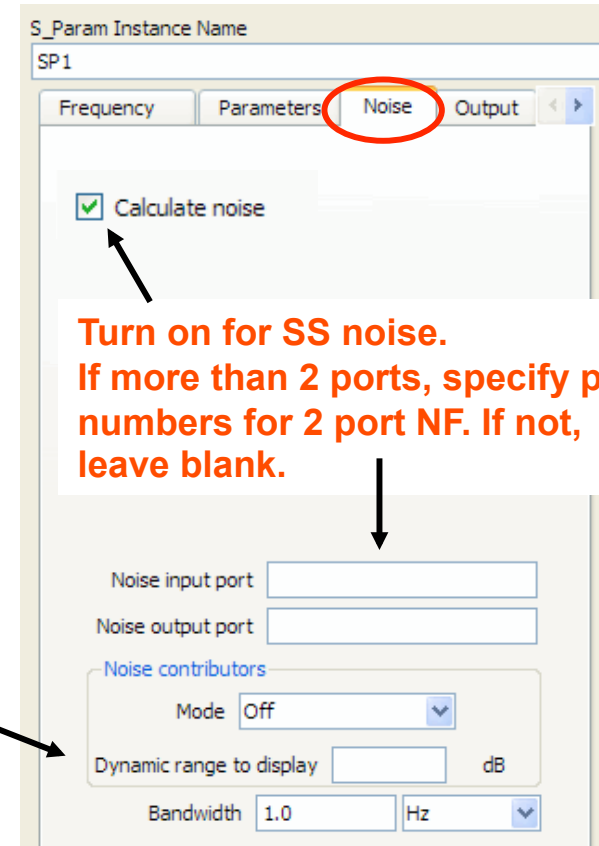


Calculate other parameters.

Enable Frequency Conversion for ADS system mixer only.

Dynamic range: Leave blank and get all values.

## Noise



Turn on for SS noise. If more than 2 ports, specify port numbers for 2 port NF. If not, leave blank.

Next, Sweep Plan...



# Sweep Plan with S-parameter simulations

Sweep Plan is for sweeping **FREQ.**  
 Otherwise, use a **Parameter Sweep**  
 for variables (Vcc, pwr, etc.)

Mixer designers: Here  
 is a plan for an RF, LO,  
 and IF.

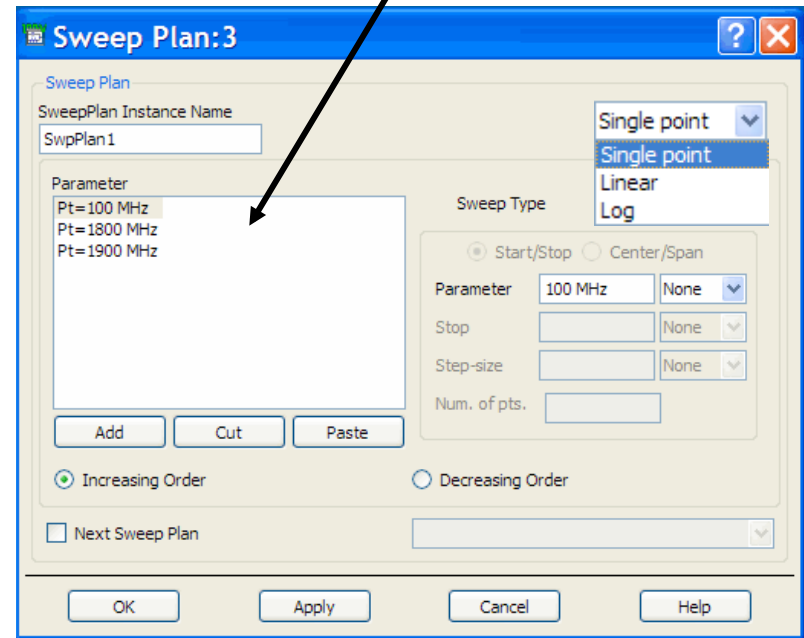
**S-PARAMETERS**

S\_Param  
 SP1  
 SweepPlan="SwpPlan1"  
 Start=100 MHz  
 Stop=4 GHz  
 Step=100 MHz

These are  
 ignored if  
 Sweep plan  
 is selected!

**SWEEP PLAN**

SweepPlan  
 SwpPlan1  
 Pt=100MHz  
 Pt=1800MHz  
 Pt=1900MHz  
 UseSweepPlan=  
 SweepPlan=  
 Reverse=no



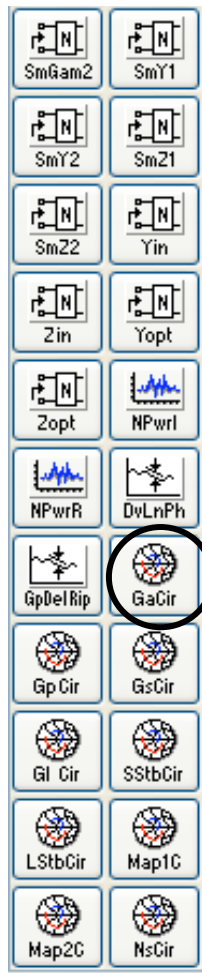
You can also  
 have Sweeps  
 within Sweeps.

SweepPlan  
 SwpPlan1  
 Start=100 MHz Stop=4 GHz Step=100 MHz Lin=  
 Start=1.8 GHz Stop=2.0 GHz Step=2 MHz Lin=

Next, Measurement Equations ...

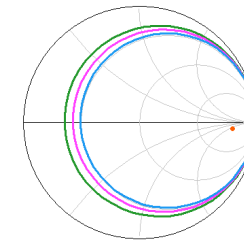


# S-Parameter measurement equations



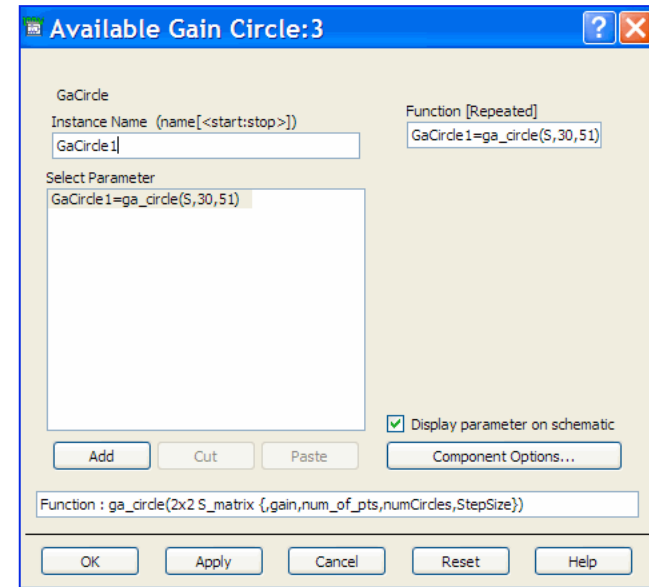
Simulation palettes have specific measurement equations - you set the arguments. Here, S is the matrix, 30 is the value in dB, and 51 points used to draw the circle.

Example: 3 circles for 3 different values of gain.



Arguments explained briefly here.

GaCircle  
GaCircle1  
GaCircle1=ga\_circle(S,30,51)

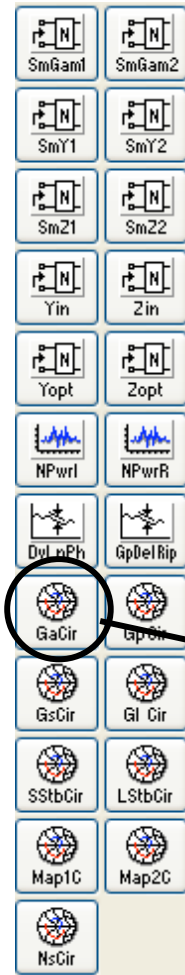
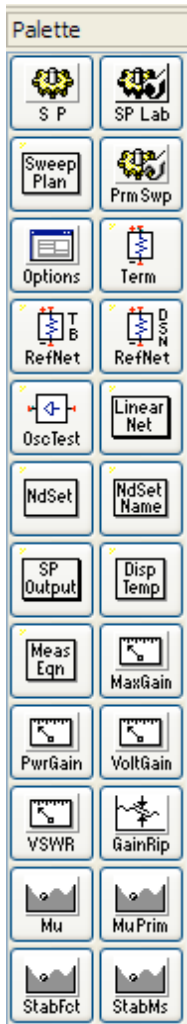


You will use some of these in the labs...

Next, matching...

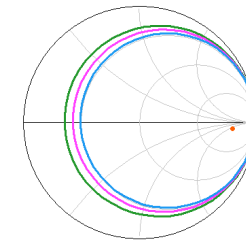


# S-Parameter measurement equations



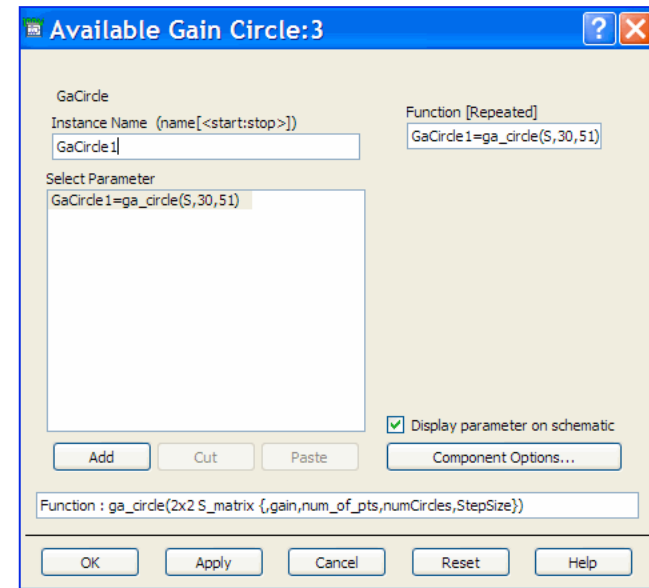
Simulation palettes have specific measurement equations - you set the arguments. Here, S is the matrix, 30 is the value in dB, and 51 points used to draw the circle.

Example: 3 circles for 3 different values of gain.



Arguments explained briefly here.

GaCircle  
GaCircle1  
GaCircle1=ga\_circle(S,30,51)



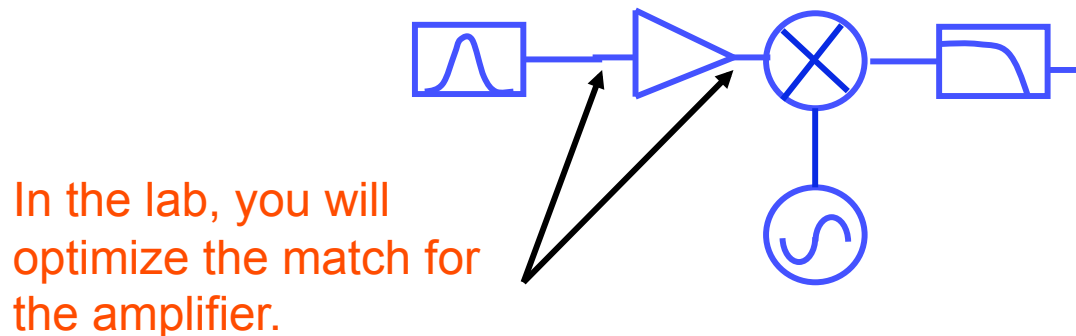
You will use some of these in the labs...

Next, matching...



# Creating Matching Networks

- Various topologies can be used: L, C, R
- Avoid unwanted oscillations (L-C series/parallel)
- Yield can be a factor in topology (sensitivity)
- Use the fewest components (cost + efficient)
- Sweep or tune component values to see S-parameters
- Optimization: use to meet S-parameter specs (goals)



**NOTE:** For a mixer, match  $S_{11}$  @ RF and  $S_{22}$  @ IF.

Use the Smith chart for matching



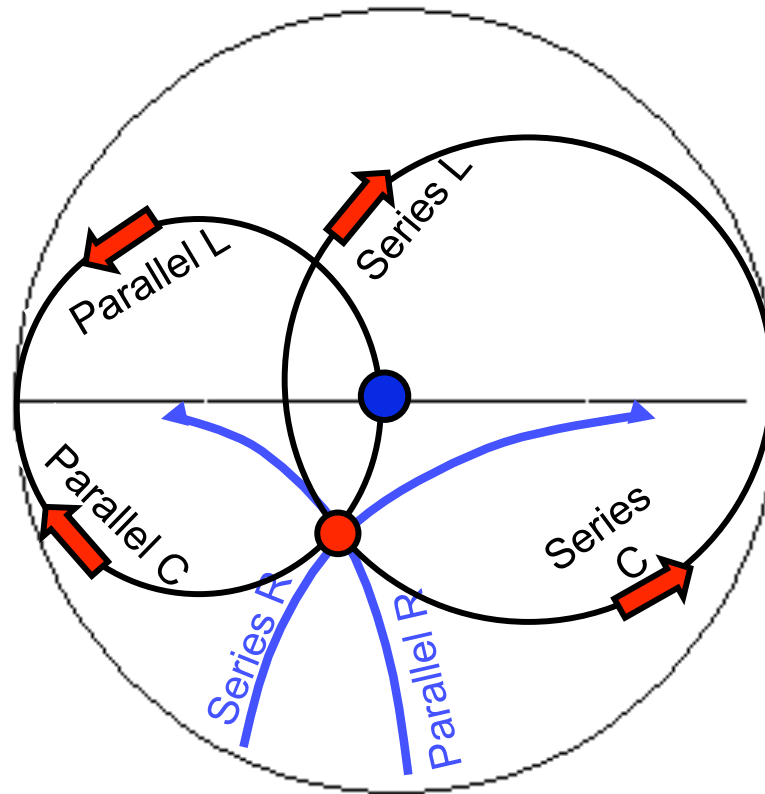
# Matching means:

## Moving toward the center of the Smith Chart!

Add Series or Parallel (shunt) components.

You will do this  
in the lab.

Adjust the value to move toward open, short, L, C, or center of chart.



Next, Smith Chart Utility for matching networks...



# Smith Chart Utility for matching...

• Insert the component in schematic.  
• Select: Tools > Smith Chart  
• Set freq, source Z, load Z.  
• Select components: L, C, R, etc.  
• View response...build the circuit.

Smith Chart Matching

DA\_SmithChartMatch1\_s\_params  
DA\_SmithChartMatch1  
Fp=1.9 GHz  
Zg=(50+j\*50) Ohm

Gamma: 1.30125 < 44.0659  
VSWR: -7.63902  
Z: -0.84209 +j 2.19860  
Y: -0.15192 +j -0.39665

Build ADS Circuit Auto 2-Element Match Reset Close

Next, optimization...



# ADS Optimization Basics

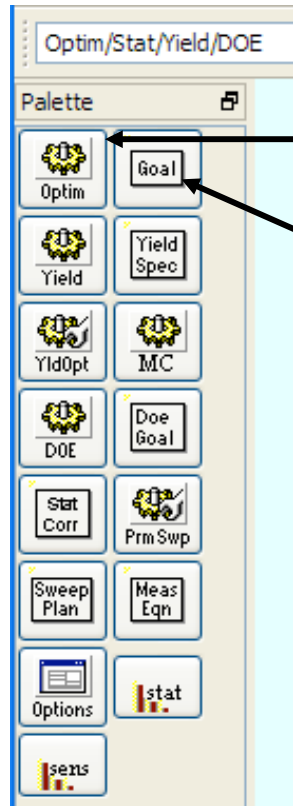
**DEFINITION: Optimization is a simulation that tries to achieve a performance goal.**

- ➔ Start with a simulation that gives you results.
- ➔ Set up the optimization which includes:
  - ◆ An optimizer type and search method.
  - ◆ A specific goal or specification to be met.
  - ◆ Enabled components or parameters to be adjusted.

**NOTE: ADS** has both continuous and discrete optimization. Yield analysis or a yield optimization is also available.

**ADS Optimization in schematic...** 

# Four elements for Optimization setup



**1 - Optim controller:**  
set the type, etc.

**2 - Goal statement:** use valid measurement equation or dataset expression.

**3 - Enable component {o}**  
for optimization.

**R=100 kOhm {o}**

R  
R1  
R=100 kOhm opt{ discrete 80 kOhm to 100 kOhm by 10 kOhm }

**4 - Simulation Controller**

**OPTIM**

Optim  
Optim1  
OptimType=Random

**GOAL**

Goal  
OptimGoal1  
Expr="dB(S(1,1))"  
SimInstanceName="SP1"  
Min=  
Max=-10.5

**S-PARAMETERS**







S\_Param  
SP1


Types of ADS optimization...



# ADS Optimization Types Available

 = Most commonly used types.

<u>Optimizer</u>	<u>Search Method</u>	<u>Error Function Formulation</u>	
 Random	random	least-squares	L2
 Gradient	gradient	least-squares	L2
Random Minimax	random	minimaxL1	MML1
Gradient Minimax	gradient	minimaxL1	MML1
Quasi-Newton	quasi-Newton	least-squares	L2
Least Pth	quasi-Newton	least P-th	seqLP
Minimax	mini-max	mini-max	MM
Random Max	random	worst case	negL2
 Hybrid	random/qNetwon	least_squares	L2
 Discrete	discrete	least-squares	L2
 <span style="border: 1px dashed orange; padding: 2px;">Genetic</span>	genetic	least-squares	L2
 Simulated Annealing	SA	least-squares	L2

 Hybrid is a combination of Random and Gradient.

### Recommendations:

- Use Random before using Gradient.
- Minimax is good for filters.
- Tune before using the optimizer.


**NOTE:** Sensitivity analysis is available in the optimization controller.

Error Function... 



# Error Function Formulation ...

**Optimizer Type determines the Error Form:**



Optim  
Optim1  
OptimType=Random  
ErrorForm=L2

Optimizers	Error Function Formulation
Random Optimizer, Gradient Optimizer, Quasi-Newton Optimizer, Hybrid Optimizer, Discrete Optimizer, Genetic Optimizer, Simulated Annealing Optimizer	Least-Squares EF
Minimax Optimizer	Minimax EF
Random Minimax Optimizer, Gradient Minimax Optimizer	Minimax L1 EF
Least Pth Optimizer	Least Pth EF
Random Max Optimizer	Negated Least-Squares EF

**Least Squares:** Each residual is squared and all terms are then summed. The sum of the squares is averaged over frequency. **Negated Least-Squares:** drives values to their extreme – effectively maximizes the error function. The goal is to find a worst typical response for a given set of parameters.

**Minimax:** attempts to minimize the largest of the residuals. This tends to result in equal ripple responses . **Minimax L1:** is similar but cannot be less than zero, so it accounts for the most severely violated cases.

**Least Pth:** The Least Pth EF formulation is similar to L2, except that instead of squaring the residuals, it raises them to the Pth power with P=2 , 4 ,6 etc.

**Next, how the EF works with your goals...**



# Goals and Error Function

- ➔ The goals are minimum or maximum target values.
- ➔ The error function is based on the goal(s).
- ➔ The weighting factor prioritizes multiple goals.

Error function is defined as a summation of residuals.

A residual  $r_i$  may be defined as:

$$\longrightarrow r_i = W_i | m_i - s_i |$$

Next, search method  
examples... 

$s_i$  is the simulated  $i$ th response (example:  $S_{21} = 9.5\text{dB}$ )

$m_i$  is the desired response for the  $i$ th measurement (example:  $S_{21} = 10\text{dB}$ )

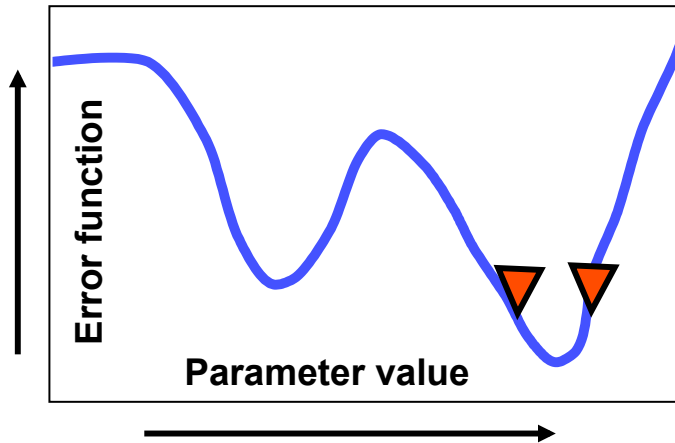
$W_i$  is the weighting factor for multiple goals: higher number is greater.

**NOTE: You can set all goals to be equally weighted.**

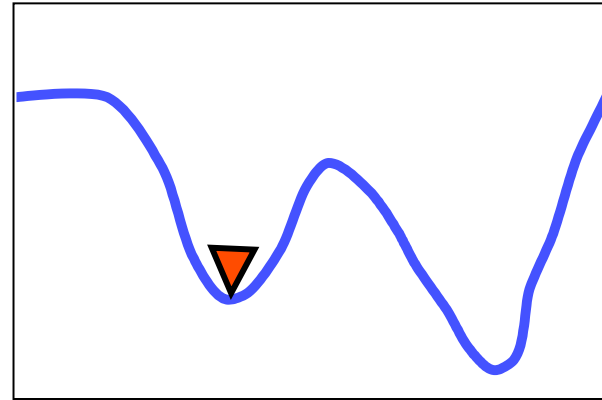
Simulations continue until the maximum iterations is reached or the error function (summation of the residuals) reaches zero (same as 10 dB).

# Search Method examples

**Random** analysis often gets you close to the goal (minimum error function).

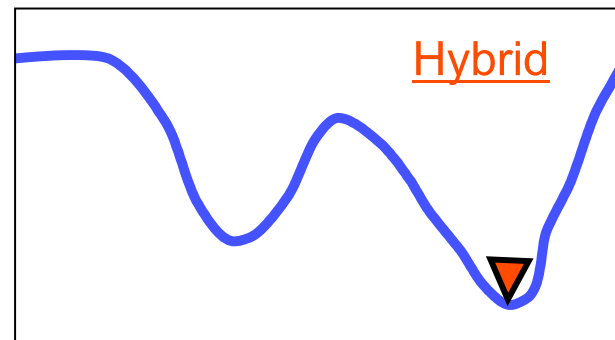


**Gradient** analysis may get stuck in a local minimum (not optimal error function).



**NOTE: Random is not totally random. It uses an adaptation that helps it move closer to the goal. For more details, refer to Help (on-line manuals).**

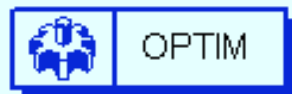
Using both **RANDOM** and **GRADIENT** can reach the desired goal or, in some cases, a hybrid type such as **Genetic**.



Next, the setup...



# Optimization Controller setup



```

Optim
Optim1
OptimType=Random
ErrorForm=L2
MaxIters=125
P=2
DesiredError=0.0
StatusLevel=4
FinalAnalysis="SP1"
NormalizeGoals=no
SetBestValues=yes
Seed=
SaveSolns=no
SaveGoals=yes
SaveOptimVars=no
UpdateDataset=yes
SaveNominal=yes
SaveAllIterations=no
UseAllOptVars=yes
UseAllGoals=yes
SaveCurrentEF=no
    
```

**Setup tab:** Select type and set iterations. Default setting use all Goals and VARs or select specific ones in OptVar tab.

**Parameters tab:** type, iterations, etc. All are displayed by default.

**Final Analysis: SP1SimInstanceName**

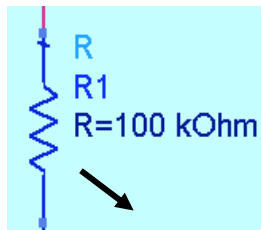
**Normalizes multiple goals!**

Avoid saving unwanted data.



# Enabling components for Opt or Stats (yield)

**OPT:** Enable and specify continuous or discrete (stepped) variation.

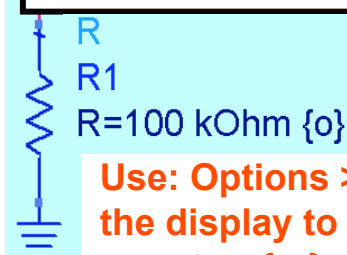


R  
100 kOhm

Equation Editor...

Tune/Opt/Stat/DOE Setup...

**NOTE:** You can also use file based values with a DAC.



**Use:** Options > Preferences > Component Text tab to set the display to show full opt range instead of {o}. Also, noopt or {-o} = disabled after the optimization.

R=96.527 kOhm noopt{ discrete 80 kOhm to 100 kOhm by 10 kOhm }

Tuning Optimization Statistics

Optimization Status Enabled

Type Discrete

Format min/max/step

Minimum Value 80 kOhm

Maximum Value 120 kOhm

Step Value 10 kOhm

Post Production Tuning

OK Cancel Help

Discrete or Continuous

Tuning Optimization Statistics

Statistics Status Enabled

Type Gaussian

Format +/- Std.Dev %

Std. Dev % 10

OK Cancel Help

Gaussian: varies up to +/- 4 sigma. Enter a % or a number for 1 sigma.

PPT is an optimization within a Yield Analysis only. Allows value to be shifted to achieve goal.

Discrete lib parts... →

NOTE: DOE is an advanced topic (statistical yield analysis).

# ADS Discrete Optimization for Library Parts

Inserted library part with listed range of values (like a DAC)

Parameter Entry Mode

Discrete optimize

CR05-181J 180 Ohm  
 CR05-201J 200 Ohm  
 CR05-221J 220 Ohm  
 CR05-241J 240 Ohm  
 CR05-271J 270 Ohm  
 CR05-301J 300 Ohm  
 CR05-331J 330 Ohm  
 CR05-361J 360 Ohm  
 CR05-391J 390 Ohm  
 CR05-431J 430 Ohm  
 CR05-471J 470 Ohm  
 CR05-511J 510 Ohm

sr\_avx\_CR\_05\_J\_19960828  
 Instance Name (name[<start:stop>])  
 R1

Select Parameter  
 PART\_NUM=CR05-391J 390 Ohm, CR05-221J 220 Ohm  
 SMT\_Pad="Pad1"  
 OFFSET=0 mil

Nominal Value  
 CR05-391J 390 Ohm

Minimum Value  
 CR05-221J 220 Ohm

Maximum Value  
 CR05-471J 470 Ohm

Display parameter on schematic

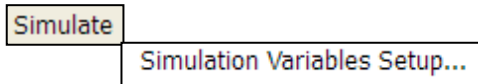
Component Options...

PART\_NUM : PART\_NUM

OK Apply Cancel Reset Help

sr\_avx\_CR\_05\_J\_19960828  
 R1  
 PART\_NUM=CR05-391J 390 Ohm, CR05-221J 220 Ohm to CR05-471J 470 Ohm

NOTE: For a list of all optimization variables:



Simulation Variables Setup

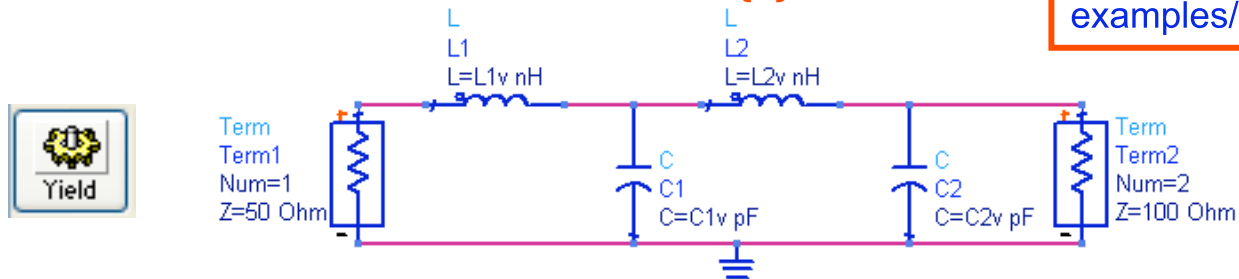
Name	Optimize	Value	Unit	Format	Min/+/-	Unit	Max	Unit
s_opt								
C_match...								
C	<input checked="" type="checkbox"/>	0.4	pF	min/max	0.01	pF	1	pF
L_match...								
L	<input checked="" type="checkbox"/>	14.3	nH	min/max	1	nH	40	nH
Term1								
Z	<input type="checkbox"/>	50	Ohm	Unconstrained				
L_match...								
L	<input checked="" type="checkbox"/>	14.3	nH	min/max	1	nH	40	nH
C_match...								

Next, Yield...

# Yield Analysis: % meeting specs!

Example: 200-400 MHz (50-to-100 ohm) Impedance Transformer. Variables have been optimized {o} and statistical Yield will now test the % of circuits meeting spec. Note that component values (VAR) have defined statistical distributions {s}.

NOTE: Optimize yield results by changing the nominal values until you get maximum (near 100%). See: [examples/Tutorial/yldoptex1\\_prj](#)



<p><b>S-PARAMETERS</b></p> <p>S_Param SP1 Start=100 MHz Stop=500 MHz Step=10 MHz</p>	<p><b>YIELD SPEC</b></p> <p>YieldSpec Spec1 Expr="dB(S11)" SimInstanceName="SP1" Min= Max=-18.0 dB Weight= RangeVar[1]="freq" RangeMin[1]=200 MHz RangeMax[1]=400 MHz</p>	<p><b>YIELD</b></p> <p>Yield Yield2 NumIters=250 PPT_Mode=none ShadowModelType=none Seed= SaveSolns=yes SaveSpecs=yes SaveRandVars=yes UpdateDataset=no SaveAllIterations=yes UseAllSpecs=yes StatusLevel=2</p>
--	---	---

**VAR**  
VAR2  
L1v=2.270670e+01 {o} {s}  
C1v=8.712450e+00 {o} {s}  
L2v=4.356221e+01 {o} {s}  
C2v=4.541352e+00 {o} {s}

Random seed



Optional Exercise:

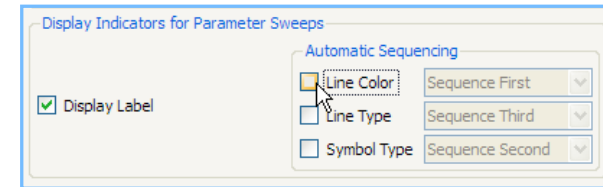
**Step 1- Copy: [examples / Tutorial / yldex1\\_prj](#)**



# ...Yield Analysis Results (data)

**Step 2 - Run the simulation once.**  
**Step 3 - Examine the results.**

NOTE on Trace Options: Uncheck Line Color to get one color.

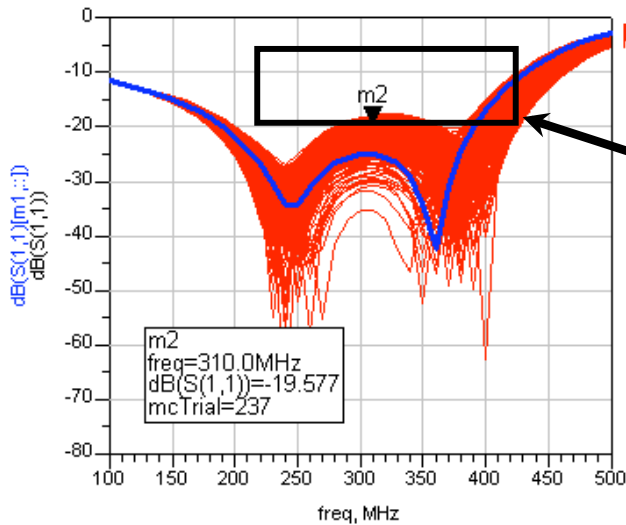


80% will meet spec

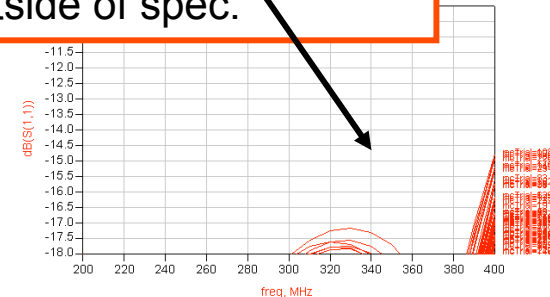
Spec is -18 dB S11  
 from 200 to 400 MHz.

**YIELD SPEC**

```
YieldSpec
Spec1
Expr="dB(S11)"
SimInstanceName="SP1"
Min=
Max=-18.0 dB
Weight=
Save=
RangeVar[1]="freq"
RangeMin[1]=200 Mhz
RangeMax[1]=400 Mhz
```



Zoomed in view: results outside of spec.



Yield Dataset results are stochastic.

More results...





# ...more Yield Analysis results

The example has Several Data Display windows: Histograms, Sensitivity and Sigma plots. Various methods are used and equations/functions are described for both histograms and sigma plots.

```

Eqn s1=sort(permute(db(S(1,1)*1e-10),"ascending")
The permute function is used to exchange the order of the dependent variables, mcTrial and freq, so that the sort function sorts with respect to the variable mcTrial and not freq.
The 1e-10 factor is added so that even if S(1,1) is zero, the dB() function does not return an error.

```

mcTrial	freq
110	1100
201	41

```

Eqn one_sigma=0.6826
Eqn two_sigma=0.954
Eqn sigmaVal=one_sigma
Eqn i1 = sweep_size(S(1,1))
Eqn i2=round(i1*(1/2))
Eqn i3=round(i2 * sigmaVal)
x1 is the value of dB(S(1,1)) at each frequency, that corresponds to the mid point of the sorted list of values.
Eqn x1 = s1[:(i2)]
x2 is the value of dB(S(1,1)) at each frequency, that corresponds to the mid point plus the number of standard deviations specified by sigmaVal of the sorted list of values.
Eqn x2= s1[:(i2+i3)]
x3 is the value of dB(S(1,1)) at each frequency, that corresponds to the mid point minus the number of standard deviations specified by sigmaVal of the sorted list of values.
Eqn x3= s1[:(i2-i3)]

```

mean_db_s1	mean_plus_sigma	mean_minus_sigma
12	126	18
13	86	

```

Alternative method of generating the mean and plus and minus n_sigma traces:
Eqn n_sigma=1
Eqn mean_db_s1=mean(permute(db(S(1,1))))
Eqn sdev_db_s1=stddev(permute(db(S(1,1))))
Eqn mean_plus_n_sigma=mean_db_s1+sdev_db_s1*n_sigma
Eqn mean_minus_n_sigma=mean_db_s1-sdev_db_s1*n_sigma

```

```

histogram_stk(data, normalized, innermostIndepLow, innermostIndepHigh, numBins, minBin, maxBin)
will produce the yield histogram. It replaces about 8 functions to produce Spec1_perHist_old.
NOTE:
Theoretically, this function should work with any number of data dimensions, but since it calls the function collapse, it becomes limited to only 4 dimensional data. (collapse function is slow, and take data with no more than 4 dimensions).

```

other related functions

- 1) histogram
- 2) histogram\_multiDim
- 3) collapse
- 4) histogram\_sens
- 5) build\_subrange

```

Eqn Spec1_perHist=histogram_stk(Spec1,200MHz,400MHz,20)
Eqn Spec1_perHist_normalized=histogram_stk(Spec1,"yes",200MHz,400MHz,20)

```

Old Method:

```

Eqn freq2=indeq(Spec1)
Eqn Spec1_freq_low_limit=200MHz
Eqn Spec1_freq_high_limit=400MHz
Eqn Spec1_low_index=find_index(freq[0:1],Spec1_freq_low_limit)
Eqn Spec1_high_index=find_index(freq[0:1],Spec1_freq_high_limit)
Eqn Spec1_subrange=Spec1[:(Spec1_low_index:Spec1_high_index)]
Eqn Spec1_subrange_freq_collapsed=collapse(max(Spec1_subrange))
Eqn Spec1_perHist_old=histogram(Spec1_subrange_freq_collapsed,20)
Eqn num_samples=sweep_size(Spec1_subrange_freq_collapsed)
Eqn Spec1_perHist_normalized_old=100*Spec1_perHist_old/num_samples

```

These two histograms are of the worst case value of dB(S(1,1)) in the frequency range, and would be useful for setting a specification. If you remove the max() function from the Spec1\_subrange\_freq\_collapsed equation, then these two histograms will be identical to the ones above.

Note that Spec1\_perHist is the histogram of all values of dB(S(1,1)) in the 200-400 MHz range, for all Monte Carlo iterations. It is not the distribution of the worst-case value in the frequency range.

MC...



# Monte Carlo Analysis

Similar to Yield but no spec is required. Enabled parameters are varied and you see the response. Also, Process or Mismatch can be selected to see the effects.

## MONTE CARLO

```
MonteCarlo
MonteCarlo1
SimInstanceName[1]="SP1"
NumIters=250
Seed=
SaveSolns=yes
SaveSpecs=yes
SaveRandVars=yes
UpdateDataset=no
SaveAllIterations=yes
UseAllSpecs=yes
StatusLevel=2
```

Format

Full Short None

Tune

Opt

Stat

Process and Mismatch used for Dynamic Link to Cadence.

**NOTE:** Click on the parameter to see highlighted range or set to Full in Options > Preferences.

Monte Carlo Simulation:7

MonteCarlo Instance Name: p1

Setup Parameters Display

Statistical Variations

stat{...}  process  mismatch

Yield Specs

Use All Specs in Design

Select Edit

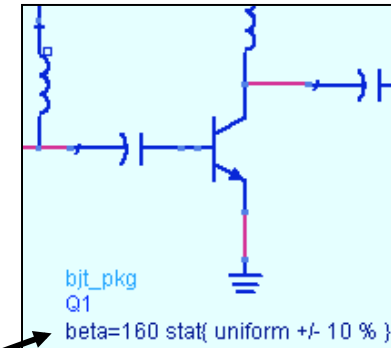
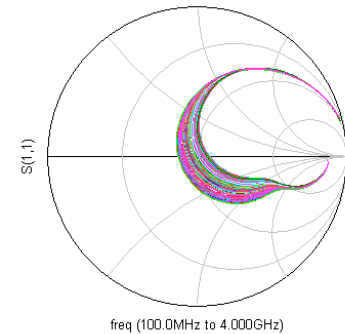
"Spec1"

Add Cut Paste

Stopping criterion

Number of trials: 250

Cancel Help



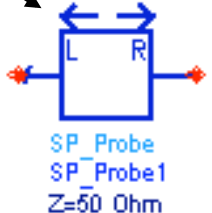
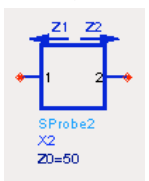
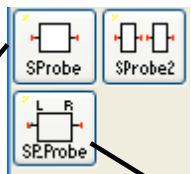
S-probe...



# S- Probes for Impedance...

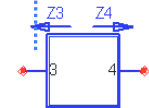
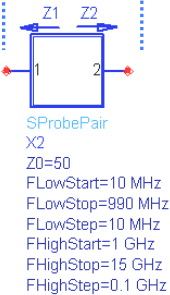
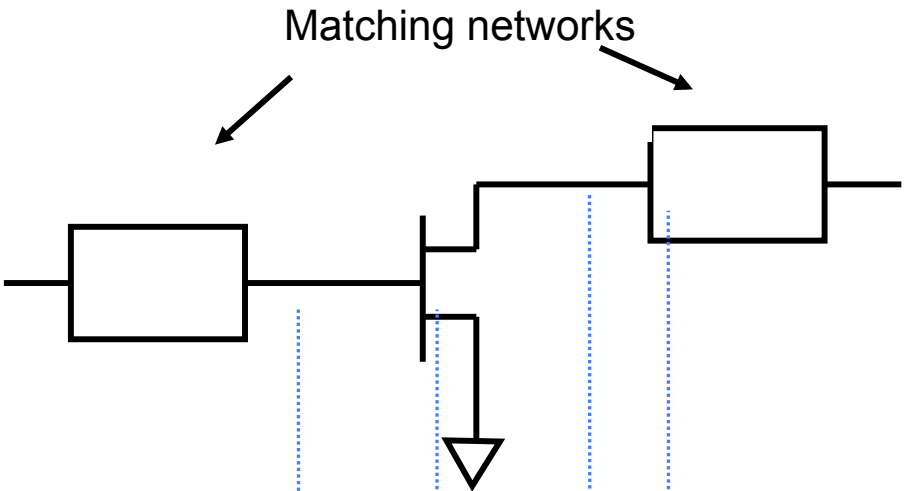
Get impedance (reflection coefficient) for biased circuits by simply connecting the S probe in series.

Probe components palette:



Insert the SP\_Probe between components or stages. Set up the simulation at the higher level.

Matching networks



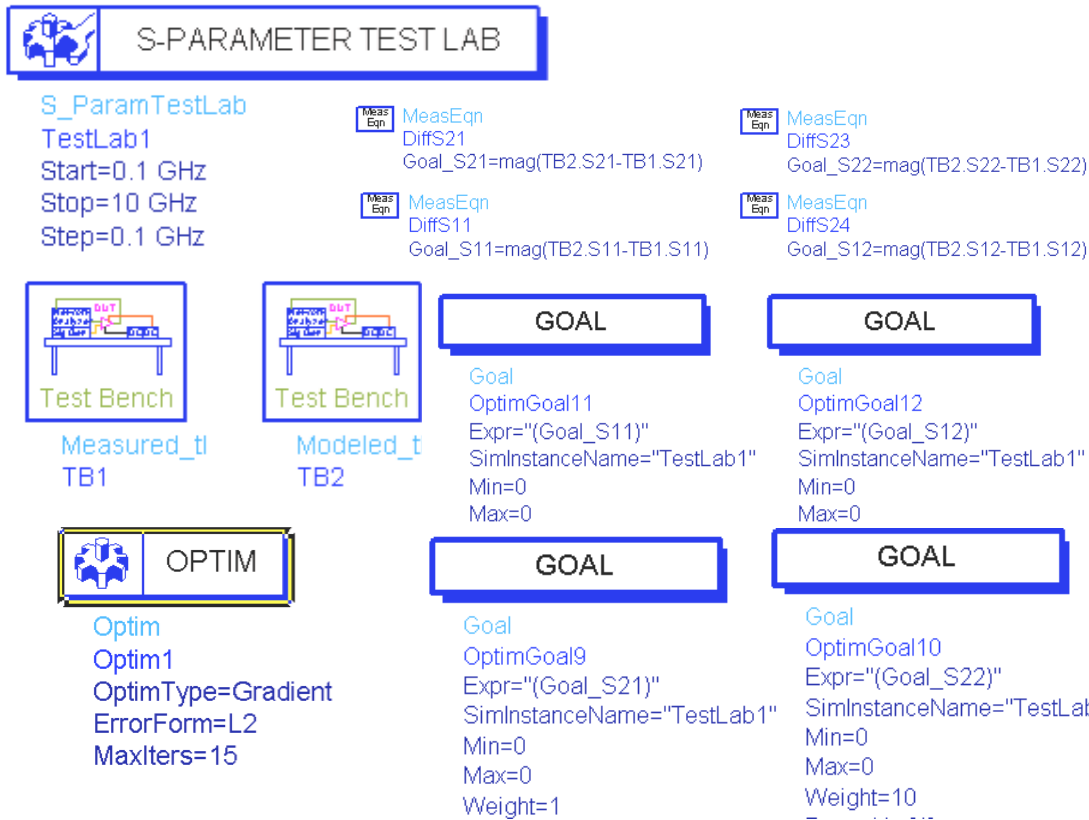
S-probe pair

DisplayTemplate  
disptemp1  
"SProbePairT"

Data Display template gives Impedance and stability.

# S\_Param Test Lab with Test Benches

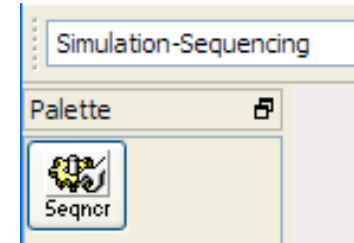
Examples/Tutorial: TestLab\_HOWTO\_prj. Shows how to Optimize a two-port model using measured two-port data.



Goals are MeasEqns = Difference between modeled and measured Test Bench values goes to zero when optimized!



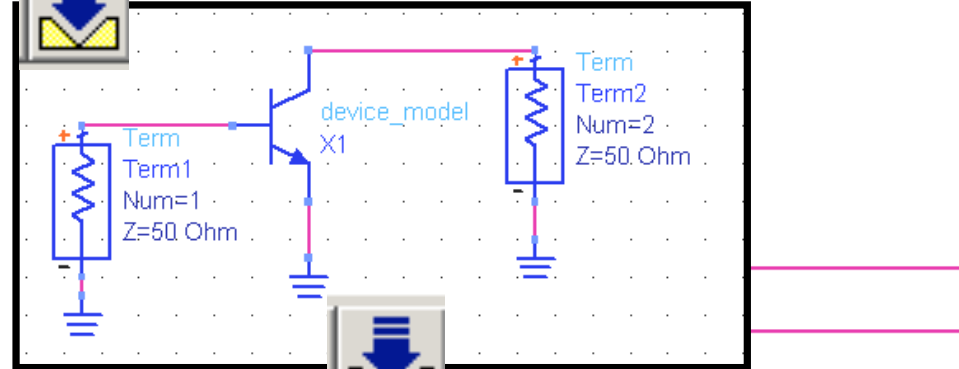
NOTE: The sequencer can also be used for an upper level design to run simulations in order.



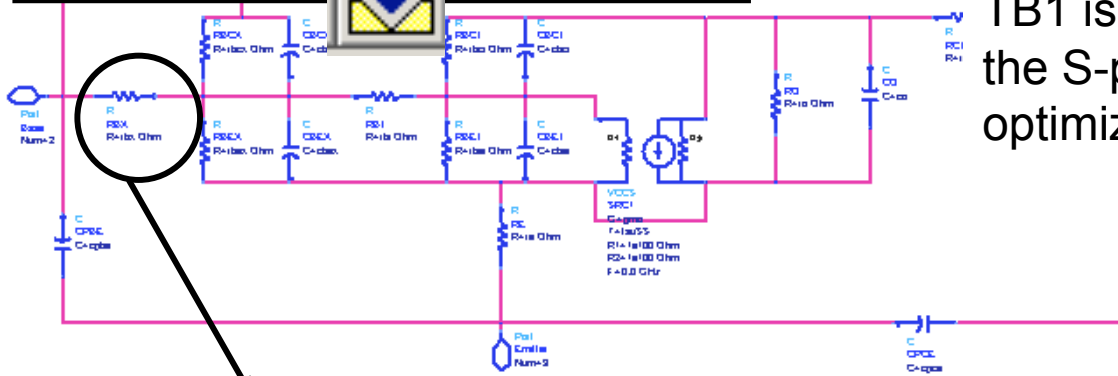
NOTE: Test benches are created using the test bench symbol (Design Parameters).

# Test Lab example (continued)...

## Hierarchy of model test bench: TB2



Enabled (opt) variables are used for all values in the lower level design. When the optimization completes successfully, TB2 will be a model of the measured device.



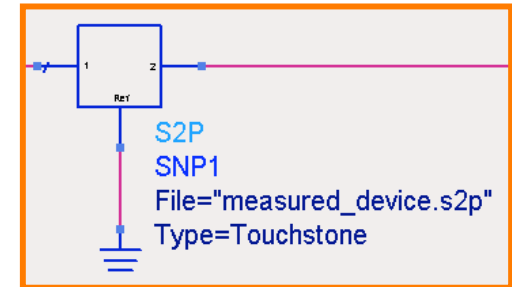
TB1 is file based and reads-in the S-param values during the optimization process.



```

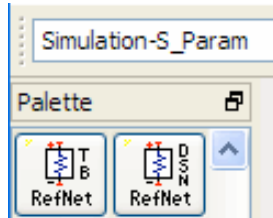
Var Eqn
VAR
VAR1
rbx=1 opt{ 0 to 1000 }
rbcx=1e12 opt{ 0 to 1e15 }
cbcx=0.05e-12 opt{ 0 to 10e-12 }
cbcx=0 opt{ 0 to 1e-9 }
    
```

## Measured test bench: TB1

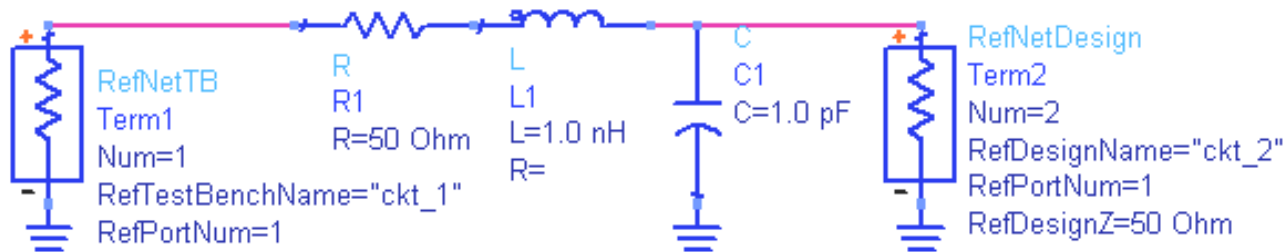


Next, Ref Nets!

# Ref Nets are Terminations



**RefNets (2 types) are terminations that reference the impedance of other schematics in your project: test benches or designs.**



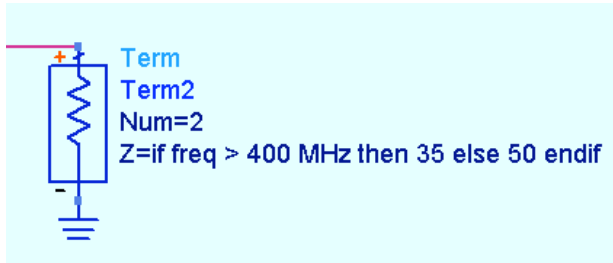
S\_Param  
SP1  
Start=100 MHz  
Stop=2 GHz  
Step=10 MHz

Next, frequency sensitive components



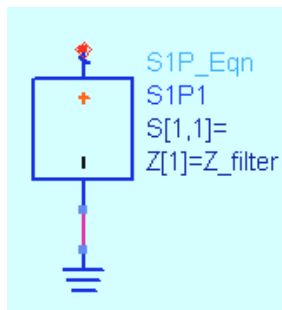
Term1 has the impedance of the test bench “ckt\_1” looking into its port 1 and Term 2 has the impedance the design “ckt\_2” looking into its port number 1.

# ADS allows frequency sensitive components!



Set a Term Z to change with frequency using IF THEN ELSE.

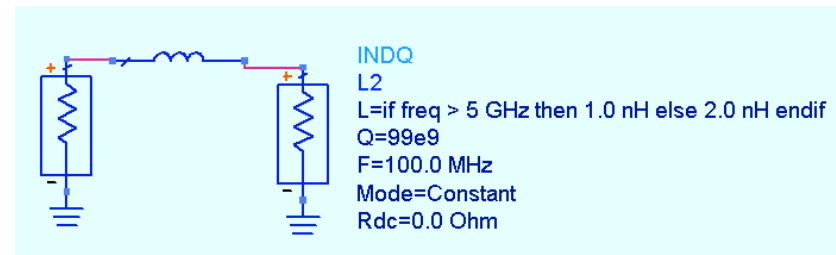
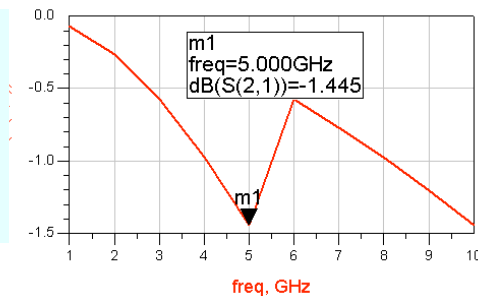
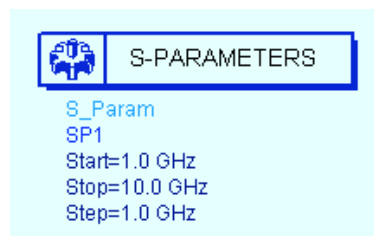
freq	PortZ	
	PortZ(1)	PortZ(2)
100.0MHz	50.000 / 0.000	50.000 / 0.000
200.0MHz	50.000 / 0.000	50.000 / 0.000
300.0MHz	50.000 / 0.000	50.000 / 0.000
400.0MHz	50.000 / 0.000	50.000 / 0.000
500.0MHz	50.000 / 0.000	35.000 / 0.000
600.0MHz	50.000 / 0.000	35.000 / 0.000
700.0MHz	50.000 / 0.000	35.000 / 0.000



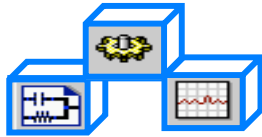
Use an SNP Eqn: Z changes with changing frequency.

```
var Eqn VAR if then elseif then elseif then else endif
VAR1
Z_filter=if freq < 1 GHz then 100 elseif freq < 500 MHz then 1K elseif freq < 1MHz then 10K else 1M endif
```

CAPQ and INDQ: Equation describes changing L or C with frequency.



Try these as an extra exercise after the lab!



What the lab is about ...

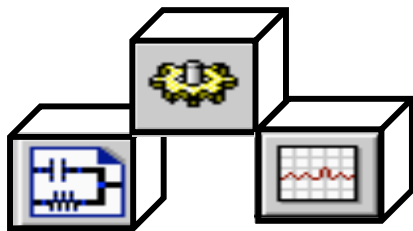
Lab 5:

# S-parameter Simulation and Optimization



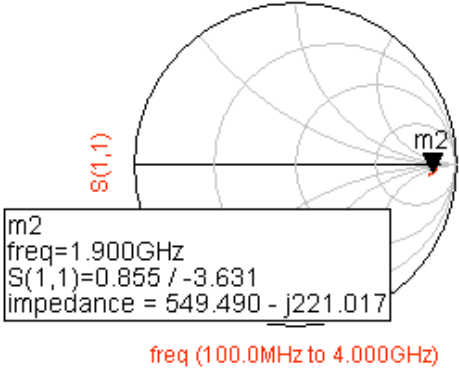
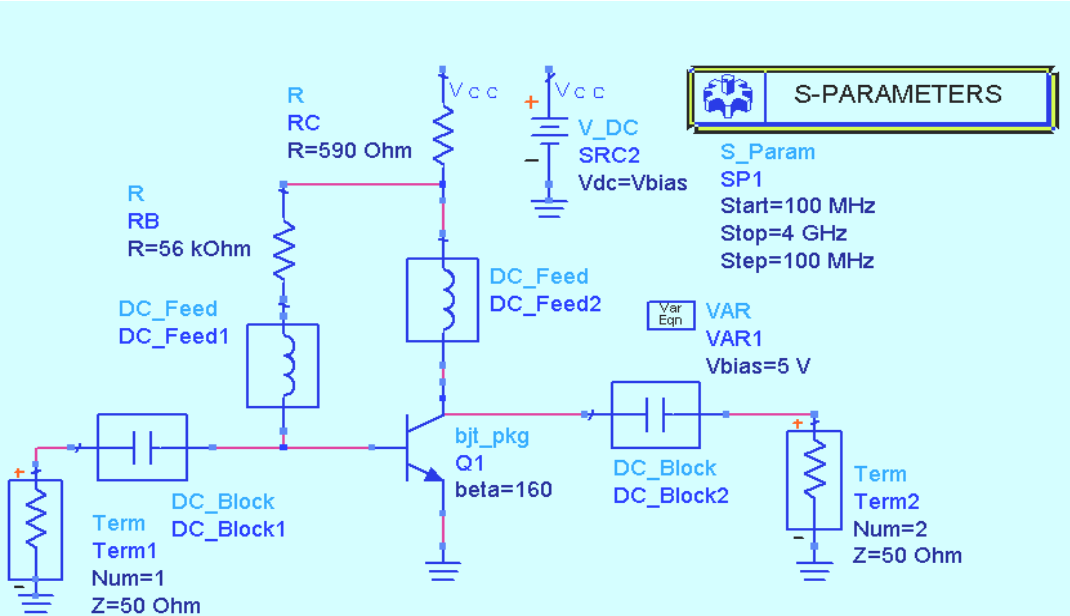
# Steps in the Design Process

You are here:

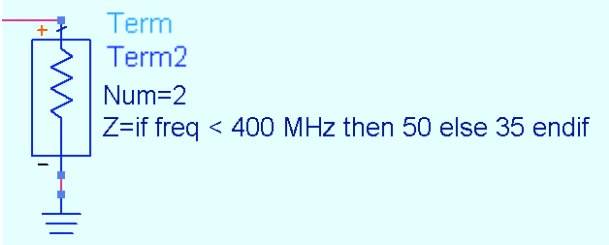


- Design the RF sys behavioral model receiver
- Test conversion gain, spectrum, etc.
- Start amp\_1900 design – subckt parasitics
- Simulate amp DC conditions & bias network
- Simulate amp AC response - verify gain
- Test amp noise contributions – tune parameters
- **Simulate amp S-parameter response**
- **Create a matching topology**
- **Optimize the amp in & out matching networks**
- Filter design – lumped 200MHz LPF
- Filter design – microstrip 1900 MHz BPF
- Transient and Momentum filter analysis
- Amp spectrum, delivered power, Zin - HB
- Test amp comp, distortion, two-tone, TOI
- CE basics for spectrum and baseband
- CE for amp\_1900 with GSM source
- Replace amp and filters in rf\_sys receiver
- Test conversion gain, NF, swept LO power
- Final CDMA system test CE with fancy DDS
- Co-simulation of behavioral system

# First, simulate with ideal components



- Plot the S parameter data
- Write eqn IF-THEN-ELSE for a Term Z.
- Simulate and list the data



freq	PortZ(2)
100.0MHz	50.000 / 0.000
200.0MHz	50.000 / 0.000
300.0MHz	50.000 / 0.000
400.0MHz	35.000 / 0.000
500.0MHz	35.000 / 0.000
600.0MHz	35.000 / 0.000
700.0MHz	35.000 / 0.000
800.0MHz	35.000 / 0.000
900.0MHz	35.000 / 0.000
1.000GHz	35.000 / 0.000



# Calculate C and L values and re-simulate

Reactance of 10 pF at 1.9 GHz and a list of L values:

**Eqn**  $X_c = -1 / (2 * \pi * 1900M * 10e-12)$

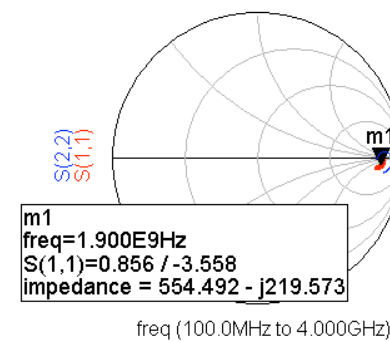
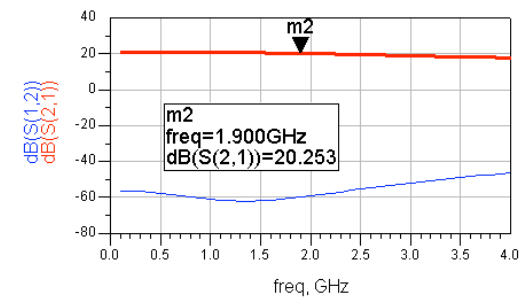
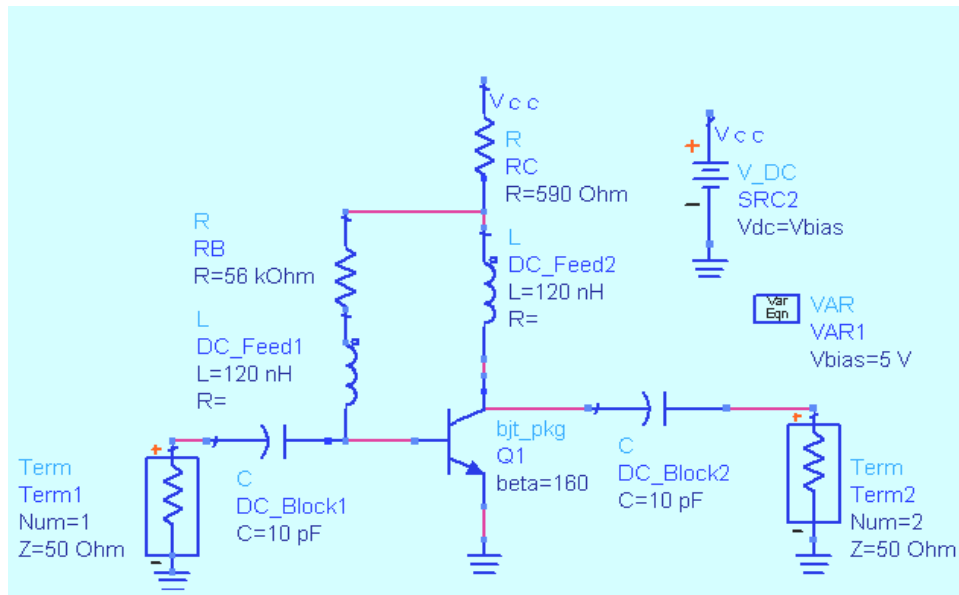
Cap value: reactance at 1900 MHz

Xc	-8.377
----	--------

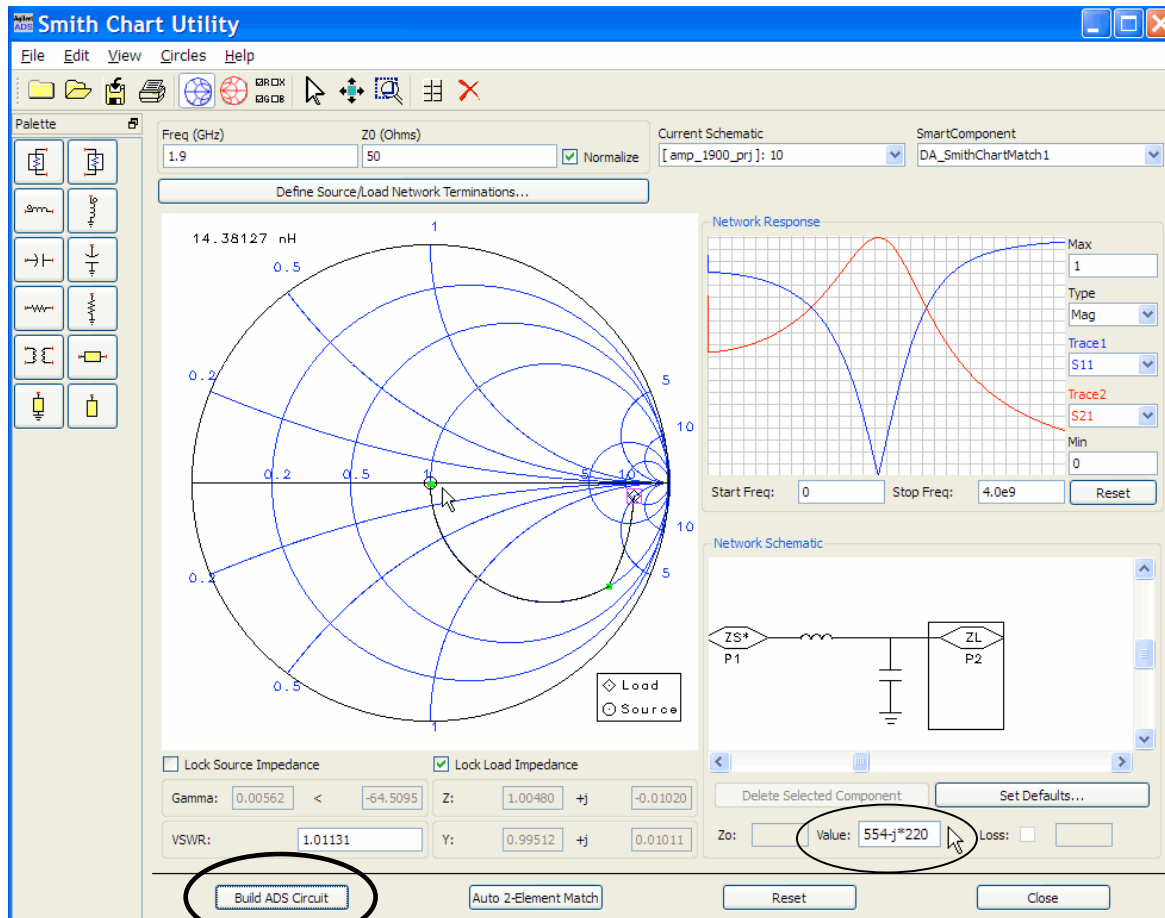
**Eqn**  $X_L = 2 * \pi * 1900M * L\_val$

**Eqn**  $L\_val = [1n :: 10n :: 200n]$

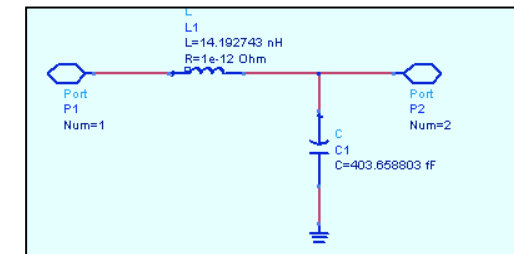
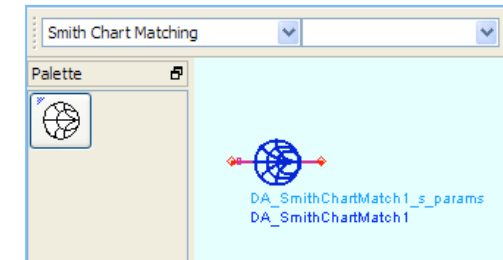
L_val	XL
7.100E-8	847.602
8.100E-8	966.982
9.100E-8	1086.363
1.010E-7	1205.743
1.110E-7	1325.124
1.210E-7	1444.504
1.310E-7	1563.885



# Smith Chart Utility for a matching network...



Smart component in schematic becomes a sub-circuit when built:



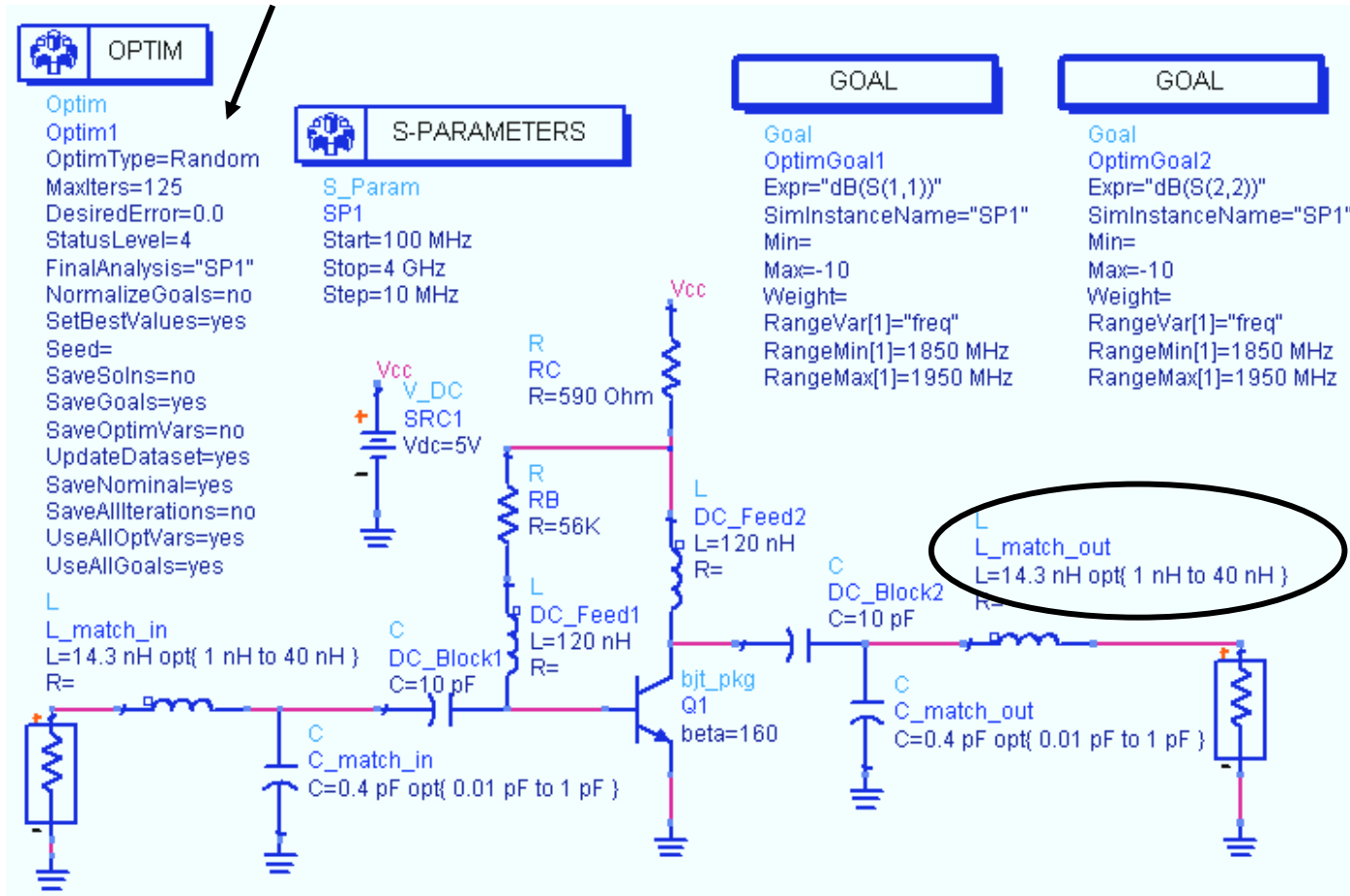
- Start the tool, define value of Z Load at 1900 MHz, insert the component
- Select the desired palette components (C and L) to move to Smith Chart center
- Build the matching circuit with one click...

Next, Optimize!



# Add output matching components and Optimize !

Use **Random** with 2 **Goals**: Max for S11 and S22 = -10 dB



**NOTE:** opt function or {o} indicates components are enabled.



Note: If GOAL RangeVar is not set, the optimizer uses all the simulation frequency points.

# After a successful optimization...

```

Status / Summary
-----
Iteration/Trial #56:
CurrentEF: 0
Optimization variables:
L_match_out.L = 28.0706e-09
C_match_out.C = 153.647e-15
C_match_in.C = 343.882e-15
L_match_in.L = 21.0353e-09

Simulation finished: dataset 's_opt' written in:
'C:\users\default\amp_1900_prj\data'.
    
```

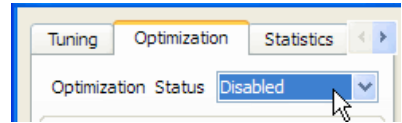
**EF = 0 = goals met!**

Nominal Optimization found initial Error Function (EF) to be zero or less. Optimization specification is currently satisfied... Try to tighten specification

Update the values and disable the Opt components: {-o} = noopt function.

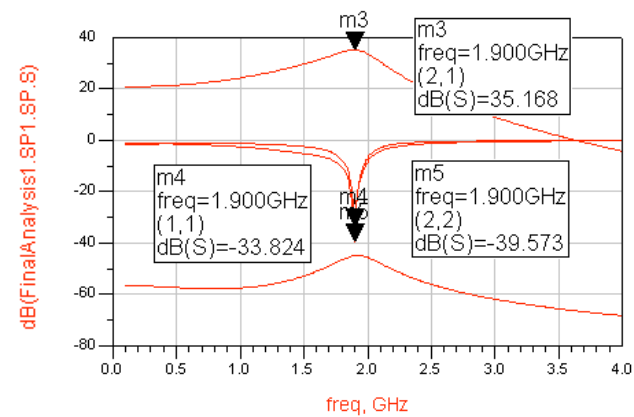
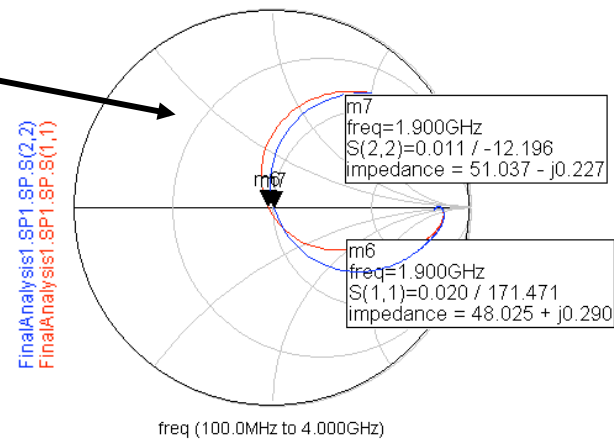
```

Simulate
Update Optimization Values
C
C_match_out
C=0.223529 pF {-o} pF
    
```



Simulator will tell you if it can improve results!

Lab exercise has final component values for these S-parameters which are good!



# Also, gain & noise circles + stability

S-parameter simulation with gain and noise circles, and stability.  
 These pre-defined measurement equations use ADS functions.



**GaCircle**  
GaCircle1  
GaCircle1=ga\_circle(S,30,51)

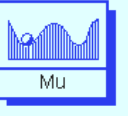


**NsCircle**  
NsCircle1  
NsCircle1=ns\_circle(nf2,NFmin,Sopt,Rn/50,51)


You set the arguments if necessary!

**OPTIONS**

Options  
 Options1  
 Temp=16.85  
 Tnom=25  
 TopologyCheck=yes  
 V\_RelTol=1e-6  
 V\_AbsTol=1e-6 V  
 I\_RelTol=1e-6  
 I\_AbsTol=1e-12 A  
 GiveAllWarnings=yes  
 MaxWarnings=10



**Mu**  
Mu1  
Mu1=mu(S)

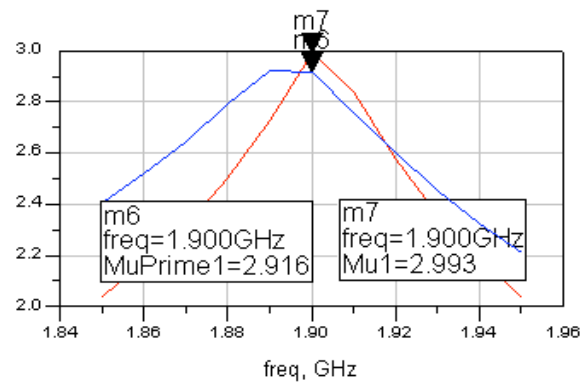
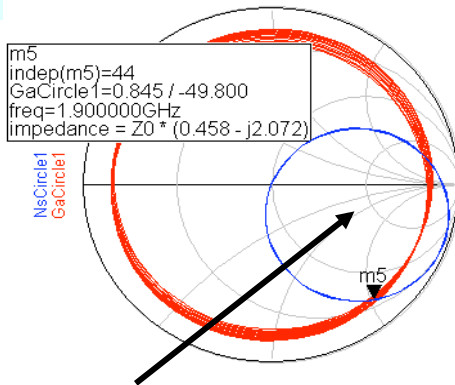


**MuPrime**  
MuPrime1  
MuPrime1=mu\_prime(S)

**NOTE: You must turn on Noise in the simulation controller for NsCircle and list values.**



freq	NFmin	Sopt
1.850GHz	1.055	0.788 / -20.779
1.860GHz	1.055	0.789 / -20.735
1.870GHz	1.056	0.791 / -20.691
1.880GHz	1.056	0.792 / -20.647
1.890GHz	1.056	0.794 / -20.602
1.900GHz	1.057	0.795 / -20.557
1.910GHz	1.057	0.797 / -20.512
1.920GHz	1.057	0.798 / -20.467



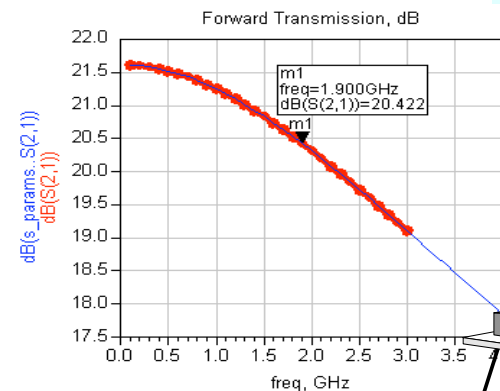
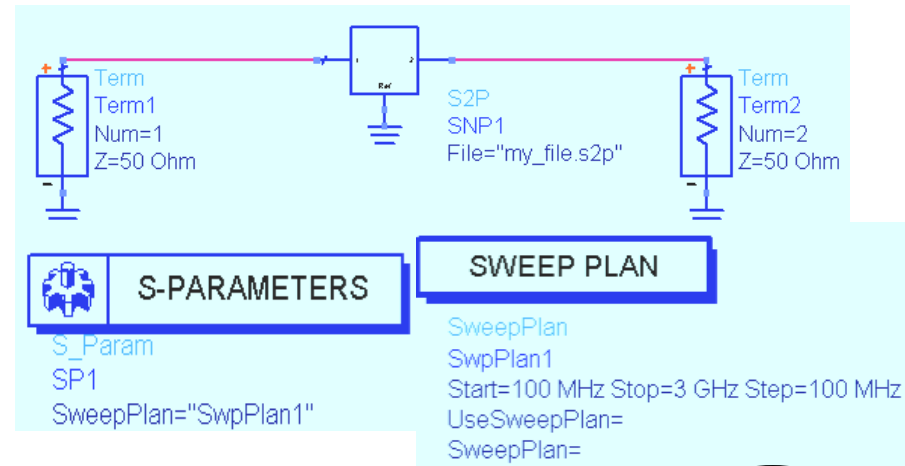
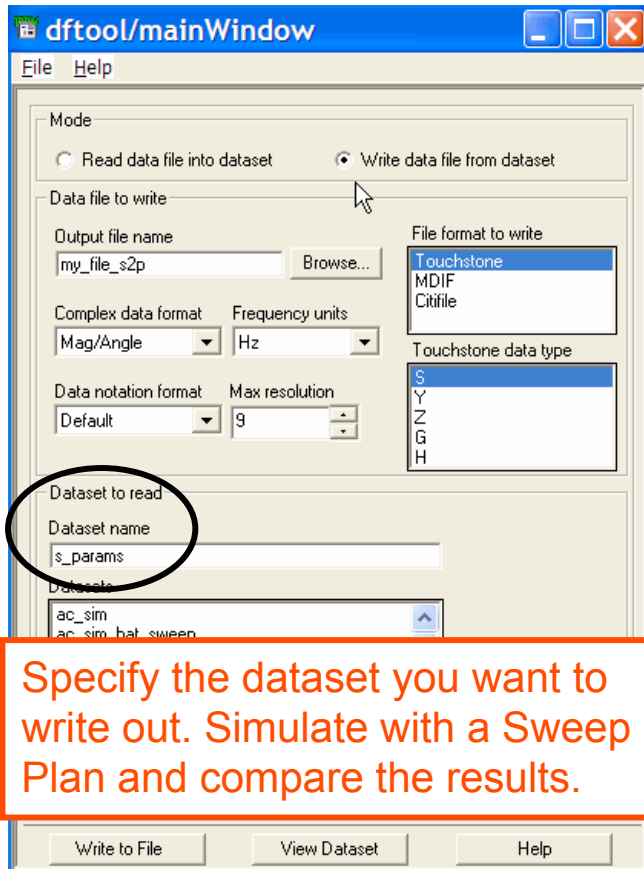
Circle center is source match for optimum NF.

# Optional: Read / Write data files

**★ Or, try the yield analysis example!**



**Data File Tool:** write an ADS S-parameter dataset as a Touchstone file, then Read it back in... as if it came from a Network Analyzer!



Start the lab now!

