GENESYS 2004 Enterprise

Simulation

Eagleware Corporation owns both the GENESYS software program suite and its documentation. No part of this publication may be produced, transmitted, transcribed, stored in a retrieval system, or translated into any language in any form without the written permission of Eagleware Corporation.

Copyright © 1985-2004 Eagleware Corporation. All rights reserved

Eagleware Corporation
635 Pinnacle Court
Norcross, GA 30071
USA
Main Phone: 678-291-0995
Sales Phone: 678-291-0259
Support Phone: 678-291-0719
Fax: 678-291-0971

Printed in the United States of America.

Version 2004 first printing March 2004

Table Of Contents

Chapter 1 Overview 1

Simulations / Data 1
Which Simulator Should I Use? 1

Chapter 2 Walkthrough: DC, Linear, HARBEC 5

Harmonic Balance Walkthrough 5
BJT Amplifier Design and Simulation 5
CREATING AN OUTPUT VS. INPUT POWER GRAPH 14

Chapter 3 Walkthrough: SPECTRASYS 17

Overview 17
Creating a Schematic 17
Adding a SPECTRASYS simulation 18
Level Diagrams 22
System Simulation Parameters - Tuning Parameters 23
Add an Amplifier 24
Add a Mixer 26
Multiple Signals 27

Chapter 4 Linear Simulation 31

Overview 31
Linear Simulation Properties 31
S Parameters 32
Overview 32
S-Parameter Basics 32
Stability 34
Matching 35
GMAX and MSG 36
The Unilateral Case 37
Gain Circles 37
Noise Circles 38
Smith Chart 39

Chapter 5 HARBEC (DC & Harmonic Balance) 41

DC Analysis Overview 41
DC Analysis Properties 41
Harmonic Balance Overview 43

HARBEC Options 44
HARBEC Popup Menu 51
Entering Nonlinear Models 52
Typical Harmonic Balance Measurements 53
Compression 53
Solving Convergence Issues 53
Optimizing Simulation Performance 54
Jacobian Calculation 54
Order vs. Accuracy and Time 54
Amplitude Stepping 55
Krylov Subspace Iterations 55
Nonlinear Noise Analysis 55
Oscillator Design Overview 60
Motorola LDMOS 65

Chapter 6 Advanced Modeling Kit 67

Advanced Modeling Kit Overview 67
Using the Additional AMK Models 67
Creating New Verilog-A Models 68
Customizing Built-In Nonlinear Models 69
Verilog-A Tutorial 69
Verilog-A Reference 71
Verilog-A Reference Overview 71
Preprocessor 71
Data Types and Parameters 73
Analog Block 76
Analog Functions 82
System tasks and functions 83
Eagleware Verilog-A Extensions 84

Chapter 7 SPECTRASYS (System) 87

System Models 87
Glossary 87
Dialog Box Reference 88
System Simulation Parameters - General Tab 88
System Simulation Parameters - Paths Tab 89
System Simulation Parameters - Calculate Tab 90
System Simulation Parameters - Composite Spectrum Tab 94
System Simulation Parameters - Options Tab 97
How it Works 100
Amplifiers 100

Channel 104
Coherency 106
Intermods & Harmonics 108
Mixers 113
Broadband Noise 116
Paths 117
Outputs 121
Level Diagrams 121
Composite Spectrum 122
Identifying Spectral Origin 124
Sources 127
Sources 127
Synthesis 132
Synthesis 132
Troubleshooting 133
How come my noise figure decreases through a cascade? 133
Why don't I get the same answer as my spreadsheets? 134
System Simulation Tips 135

Chapter 8 Parameter Sweeps 139

Parameter Sweep 139
Parameter Sweep Properties 139

Chapter 9 Equation Reference 141

Statements 141
Viewing Variable Values 144
Operators 144
Sample Expressions 145
Built-in Functions 145
Constants 148
Strings 148
Arrays (Vectors and Matrices) 148
Post Processing 150
Equation Wizard 153
Graphing an Equation 153
Equations in the Equations Section 153
Logical Operators 154
User Functions 154
Calling Your FORTRAN/C/C++ DLLs 155
Equations Overview 155

Chapter 10 Device Data 157

Linear vs. Nonlinear Device Models 157
Linear Data Overview 157
Using a Data File in GENESYS 157
Link To Data File 158
Link to Data File Setup 158
Provided Device Data 158
Creating New Linear Data Files 158
File Record Keeping 160
Exporting Data Files 160
Noise Data in Data Files 160
Nonlinear Device Library 163

Chapter 11 User Models 165

Overview 165
Creating A Model 165
User Model Example: A Self Resonant Capacitor 167
Model Properties 172
Using A Model In SCHEMAX 173
Single Part Models 173
Text Model Definitions 174

Chapter 12 Link to Spice File 175

Overview 175
SPICE File Compatibility 176
Link to Spice File 177

Chapter 13 Measurements: Overview 179

Overview 179
Linear Measurements 179
Nonlinear Measurements 181
Operators 181
Sample Measurements 183
Using Non-Default Simulation/Data 183
Using Equation Results (post-processing) 184

Chapter 14 Measurements: Linear 185

S-Parameters 185
H-Parameters 186
Y-Parameters 186

Z-Parameters 187
Voltage Standing Wave Ratio (VSWR) 188
Input Impedance / Admittance (ZINi, YINi) 189
Voltage Gain 189
Noise Measure (NMEAS) 190
Noise Figure (NF) / Minimum Noise Figure (NFMIN) 191
Constant Noise Circles (NCI) 191
Noise Correlation Matrix Parameters 192
Simultaneous Match Gamma at Port i (GMi) 193
Simultaneous Match Admittance / Impedance at Port i (ZMi, YMi) 193
Maximum Available Gain (GMAX) 194
Available Gain & Power Gain Circles (GA, GP) 194
Unilateral Gain Circles at Port i (GU1, GU2) 195
Stability Factor (K), Stability Measure (B1) 196
Input / Output Plane Stability Circles (SB1, SB2) 196
Optimal Gamma for Noise (GOPT) 197
Optimal Admittance / Impedance for Noise (YOPT, ZOPT) 197
Effective Noise Input Temperature (NFT) 198
Normalized Noise Resistance (RN) 198
Reference Impedance (ZPORTi) 199

Chapter 15 Measurements: Nonlinear 201

Port Power (Pport) 201
Probe Current (Iprobe) 201
Node Voltage (Vnode) 202
Reference Impedance (ZPORTi) 202
Large Signal S Parameters 203

Chapter 16 Measurements: Load Pull 205

Load Pull Contours 205
To create a new file using load pull contours: 206

Chapter 17 Measurements: SPECTRASYS 207

Adjacent Channel Power (ACP[U or L][n]) 207
Adjacent Channel Frequency (ACF[U or L][n]) 208
Added Noise (AN) 208
Cascaded Gain (CGAIN) 209
Cascaded Gain [Third Order Intermod Analysis] (CGAINIM3) 210
Cascaded Gain - All Signals (CGainAll) 211
Carrier to Noise Ratio (CNR) 211
Cascaded Noise Figure (CNF) 212

Channel (or Path) Frequency (CF) 213
Offset Channel Frequency (OCF) 214
Tone Channel Frequency (TCF) 214
Channel Noise Power (CNP) 215
Channel Power (CP) 215
Desired Channel Power (DCP) 216
Desired Channel Power [Third Order Intermod Analysis] (DCPIM3) 217
Offset Channel Power (OCP) 217
Tone Channel Power (TCP) 218
Gain (GAIN) 219
Gain [Third Order Intermod Analysis] (GAINIM3) 219
Gain - All Signals (GAINALL) 220
Image Frequency (IMGF) 221
Image Channel Noise Power (IMGNP) 222
Image Noise Rejection Ratio (IMGNR) 223
Minimum Detectable Signal (MDS) 223
Image Channel Power (IMGP) 224
Image Rejection Ratio (IMGR) 225
Percent Noise Figure (PRNF) 226
Percent Third Order Intermod (PRIM3) 227
Spurious Free Dynamic Range (SFDR) 228
Stage Dynamic Range (SDR) 229
Stage Noise Figure (SNF) 229
Stage Output 1 dB Compression Point (SOP1DB) 230
Stage Output Second Order Intercept (SOIP2) 231
Stage Output Third Order Intercept (SOIP3) 231
Stage Output Saturation Power (SOPSAT) 232
Input Third Order Intercept (IIP3) 232
Output Third Order Intercept (OIP3) 233
Conducted Third Order Intermod Power (CIM3P) 234
Generated Third Order Intermod Power (GIM3P) 235
Total Third Order Intermod Power (TIM3P) 236
Total Node Power (TNP) 237

Chapter 18 EMPOWER: Operation 239

Overview 239
Features 239
Examples 240
Creating a Layout 240
Creating a Layout Without a Schematic 241
Box Dimensions 242

General Layers 242
EMPOWER Layers 243
Drawing the Layout 245
Centering the Layout 247
Placing EMPOWER Ports 247
Simulating the Layout 248
Viewing Results 250
Using the Viewer 251
Creating a Layout From an Existing Schematic 252
Simulating the Layout 254
Lumped Elements 255
Real-Time Tuning 256

Chapter 19 EMPOWER: Basics 257

Overview 257
2-D SIMULATORS 257
3-D SIMULATORS 257
2 1/2-D SIMULATORS 257
Basic Geometry 257
The Grid 261
Viaholes and Z-Directed Ports 264
EM Ports 265
EMPOWER Options 265
General Tab 265
Viewer / Far Field Tab 267
Advanced Tab 268
Console Window 270
Batch Runs 271

Chapter 20 EMPOWER: Tips 273

Overview 273
Cell Size 273
Maximum Critical Frequency 274
Symmetry 274
Thinning Out 275
Wall & Cover Spacing 275
Cover Type 276
Lossy Analysis 276
Viewer Data 276
Slot-Type Structure 277
Preferred Cell Count 277

Thick Metal ... 279

Chapter 21 EMPOWER: External Ports ... 281

Overview ... 281
Placing External Ports ... 281
EMPort Options ... 281
Deembedding ... 283
MultiMode Ports ... 284
Generalized S-Parameters ... 286

Chapter 22 EMPOWER: Decomposition ... 289

Overview ... 289
Basics ... 289
Spiral Inductor Example ... 290
Losses ... 295
Port Numbering ... 296

Chapter 23 EMPOWER: Lumped Elements and Internal Ports ... 297

Overview ... 297
Placing Internal Ports ... 297
Manually Adding Lumped Elements ... 298
Automatic Port Placement ... 298
Planar (X- and Y-Directed) Ports ... 299
Resonance ... 301

Chapter 24 EMPOWER: Viewer and Antenna Patterns ... 303

Overview ... 303
Interface ... 303
Far-Field Radiation Pattern Viewer ... 308
Examples ... 311
MultiMode Viewer Data ... 315
Via Hole Viewer Example ... 316
Viewer Theory ... 317
Summary ... 319

Chapter 25 EMPOWER: Box Modes ... 321

Overview ... 321
Homogeneous Rectangular Cavity ... 321
Higher Order Box Modes ... 322
Partial Dielectric Loading ... 323
Signal Metal Effects ... 323

Top Cover 323
Cavity Absorber 323

Chapter 26 EMPOWER: Theory 325

Historical Background 326
Problem Formulation 326
Method of Lines 328
Mapping on the Grid 329
Grid Green's Function 331
Informational Multiport 332
Numerical Acceleration Procedures 333
De-Embedding Algorithm 335

Chapter 27 EMPOWER: File Descriptions 337

Overview 337
Text Files vs. Binary Files 338
File Extensions 338
.EMV (EMPOWER Viewer) Files 339
.L1, .L2, ...Ln (Line Data) Files 339
.LST (Listing) Files 340
.PLX (Current/Viewer Data) Files 341
.R1, .R2, ... Rn (Port Impedance) Files 341
.RGF (Line Data) Files 342
.RX (Frequency vs. Impedance) Files 342
.SS (S Parameter) Files 342
.TPL (Topology) Files 343
.WSP (Workspace) Files 343
.Y (Y-Parameter) Files 343
~SS, ~RG, etc. (Backup) Files 343

Chapter 28 EMPOWER: Advanced M/FILTER Example 345

Chapter 29 EMPOWER: References 359

General Background 359
The Method of Lines 360
Richardson's Extrapolation 360
Symmetry Processing 361
EMPOWER Engine Theory and Algorithms 361
Test Examples and Comparisons 362

Index 363

Chapter 1 Overview

Simulations / Data

GENESYS supports several different types of simulations, allowing the exploration of a complete range of circuit performance:

- DC Simulation (nonlinear) (HARBEC)
- Linear S-Parameter Simulation
- Planar 3D Electromagnetic (EM) Simulation (EMPOWER)
- Harmonic Balance Simulation (nonlinear) (HARBEC)
- Spectral Domain System Simulation (SPECTRASYS)

Additionally, the following items are available in Simulations/Data:

- Parameter Sweep
- Link To Data File
- TESTLINK (Covered in the User's Guide).

Several of these capabilities work together. EM co-simulates with either the nonlinear or linear circuit simulator, combining the accuracy of EM analysis with the generality and speed of circuit simulation. Parameters sweeps can be used with DC, linear, nonlinear, and system simulation as well as with other sweeps. Frequency, resistance, substrate height, and DC supply level are just a few of the parameters that are typically swept.

All of these simulations can be added to a workspace by right-clicking the Simulations/Data node on the Workspace Window.

Which Simulator Should I Use?

Often, we at Eagleware are asked which simulation method should be used in a particular circuit: Linear (SUPERSTAR)? Nonlinear (HARBEC)? SPICE (by exporting)? Electromagnetic (EMPOWER)? SPECTRASYS?

For most circuits, you will use a combination of the different simulations. We have developed several guidelines that should simplify the decision for most applications. First, each method has benefits and drawbacks:

	Linear	SPICE	Electromagnetic	HARBEC
Benefits	Extremely fast	Time domain	Extremely accurate	Steady-State Nonlinear
	Schematic or netlist entry	Schematic or netlist entry	Does not require an intimate knowledge of the circuit – simulator figures out coupling, etc.	Study mixing, compression and intermodulation
	Real-time tuning of circuits	Starting waveforms (e.g. oscillator startup)	Can predict radiation, current distribution.	DC biasing information
	Uses manufacturer-provided measured data	DC biasing information	Automatic deembedding	Lots of vendor-supplied models
	Requires very little memory	Lots of vendor-supplied models	Predicts box mode effects (e.g. What happens if the circuit is placed in a box?)	Use frequency dependent equations and post-processing
	Easily use equations and user functions	Non-linear modeling of crossover distortion, etc.	Can use arbitrary shapes – does not require an existing model for them	Use measured data in simulation
Drawbacks	No time domain	Very slow	Extremely slow	Much slower than linear
	No biasing information	Very hard to model frequency domain behavior (e.g. unloaded Q)	Requires lots of memory	Takes a lot of memory and time
	Everything is linear	No distributed models (e.g. microstrip, waveguide, etc.)	Discretizes metal patterns to fit grid	Requires nonlinear models
	Requires knowledge of circuit – coupling factors, parasitics, etc.	Requires knowledge of circuit – coupling factors, parasitics, etc.	Can be difficult to set up a circuit for simulation	Cannot study transient behavior (for example, oscillator startup)

In determining which simulation type to use, several points should be considered:

Linear or Electromagnetic?

1. **Should I use both circuit theory and EM simulation?** Circuit theory simulation in GENESYS is amazingly fast and interactive. No other program at any price approaches the speed of GENESYS. EMPOWER simulations are more accurate and do not require the use of specific geometric objects for which circuit models have been developed. EM simulation complements rather than replaces circuit theory simulation.
2. **What is the highest frequency used in the circuit?** If below about 1 GHz, lumped elements are often used in place of distributed elements. In this case, the final board layout usually won't add any significant parasitics or coupling concerns. Often, however, customers use EMPOWER to simulate the final board layout to make sure that it doesn't differ from the linear simulation.
3. **How big is the circuit?** If the circuit itself is very small compared to a wavelength at the highest frequency of concern, electromagnetic simulation may not be needed. This is because resonances occur at quarter wavelengths, and circuits much smaller than this usually behave as predicted by a complete linear simulation.
4. **Does the circuit have non-standard metal shapes, patterns, or geometries?** If so, electromagnetic simulation may be the only option. EMPOWER can simulate any arbitrary shape, such as ground plane pours. A linear simulator requires a netlist or schematic to describe the circuit, so models would have to exist for the pattern that you plan to simulate.
5. **Do any of the models in the circuit exceed or come close to exceeding the published parameter ranges for SUPERSTAR?** If so, you may want to verify the SUPERSTAR simulation with EMPOWER, or use EMPOWER exclusively. Most of the models in SUPERSTAR were derived from measured data, which was only taken for particular parameter variations. The allowed parameter ranges are published for each model in SUPERSTAR.

Linear or Harmonic Balance?

This question is the easiest to answer: for active circuits you will usually use both. For passive circuits (filters, couplers, power dividers, etc), you will only use linear. Passive circuits are linear-harmonic balance will not give you extra information that you could not get from linear simulation. Active circuits are inherently nonlinear. Harmonic balance will help you analyze DC operating points and nonlinear performance.

For both active and passive circuits, linear simulation is the workhorse of RF design. Matching, noise, and stability studies are all completed quickly using linear simulation. Harmonic balance is used to complete the analysis of most circuits. Examine mixer conversion gain, amplifier compression, and detector efficiency using harmonic balance.

Linear or SPICE?

Often, this question does not have a quick answer. For example, many engineers associate SPICE with time domain simulation, and a linear simulator with frequency domain simulation. Actually, many circuits have data of interest in both the time and frequency domains, which could warrant the use of both simulators. For example, an oscillator has phase noise, transmission, and phase characteristics, which are all frequency domain measurements. Oscillators also have waveform magnitude, starting time, and startup transients, which are all time domain measurements. In this case, both simulators can be used in the circuit design.

There are some guidelines for deciding between SPICE and linear simulation:

1. **Does the circuit depend on time domain characteristics?** If so, SPICE must be used for this portion of the design. If the circuit depends entirely on the time domain, SPICE can be used exclusively. However, if a frequency domain response is also of interest, linear simulation may be used in addition to SPICE.
2. **What is the highest frequency of concern in the circuit?** If it's over about 100 MHz, you may want to use linear simulation. This is because component unloaded Q becomes a concern above this frequency, and SPICE does not have the built-in ability to include this effect in simulations. If the frequency is much higher than this, linear simulation is almost a must since SPICE uses lumped element models for RF parts, which do not usually model high frequency effects accurately.
3. **Is the circuit all lumped elements?** If so, SPICE may be used. However, unloaded Q is not built into SPICE, so guideline #2 must be considered.
4. **Does the circuit contain distributed parts?** If so, linear simulation is a must since SPICE does not include distributed models. The electrical transmission line models in SPICE can be used, but for final verification of the physical implementation of the lines, linear or electromagnetic simulation should be used.

Often, both SPICE and linear simulation are useful in a design. For example, in amplifier design, the linear portion (gain, matching) can be done in SUPERSTAR, and the device biasing can be done in SPICE.

Chapter 2 Walkthrough: DC, Linear, HARBEC

Harmonic Balance Walkthrough

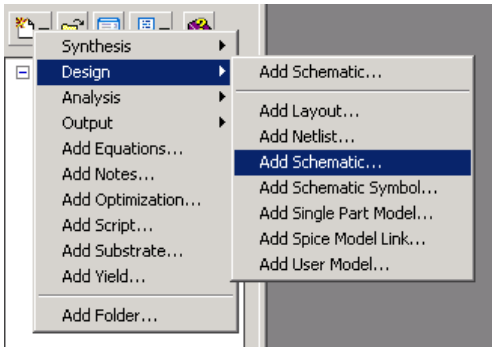
Note: This walkthrough is for customers who have purchased HARBEC (for Harmonic balance and DC simulation). If you have not purchased HARBEC, you can follow the walkthrough given in the SCHEMAX section of the User's Guide which only includes linear simulation.

BJT Amplifier Design and Simulation

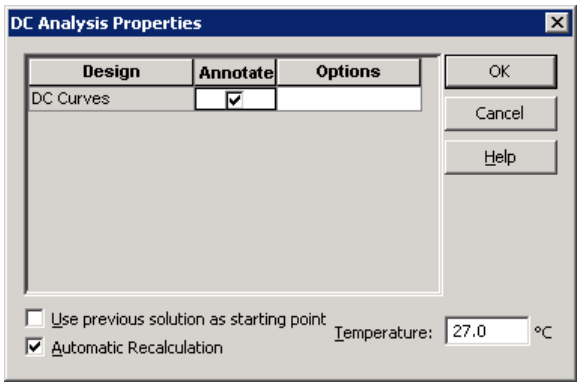
Here is a complete DC, linear, and HARBEC simulation. This example shows a common emitter BIPNPN transistor and its biasing circuit.

CREATING THE DC CURVES

1. The first step is to create the schematic. Click the 'New' button on the workspace tree and find 'Add Schematic' as shown below. Name the schematic 'NPN Sch'.



2. Find the 'NPN Bipolar Gummel-Poon' model on the 'Nonlinear' toolbar or in the parts picker and place it in the schematic.
3. Add a 'DC Current Source'. Change the 'Designator' to 'IB' and reference a variable for the 'DC Current' by typing in 'IDC'.

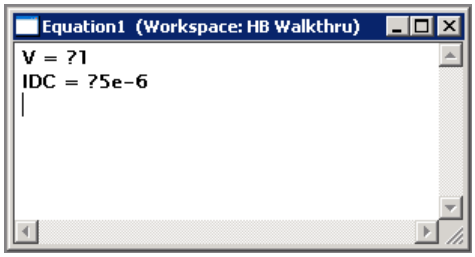


4. Add a 'Current Probe (Ammeter)'. Change the 'Designator' to 'IC'.
5. Add a 'Signal Ground'. Change the 'Designator' to 'VC' and reference a variable for the 'DC Voltage' by typing in 'V'.

Note: You may get some error messages since we have only referenced some variables and haven't created them yet.

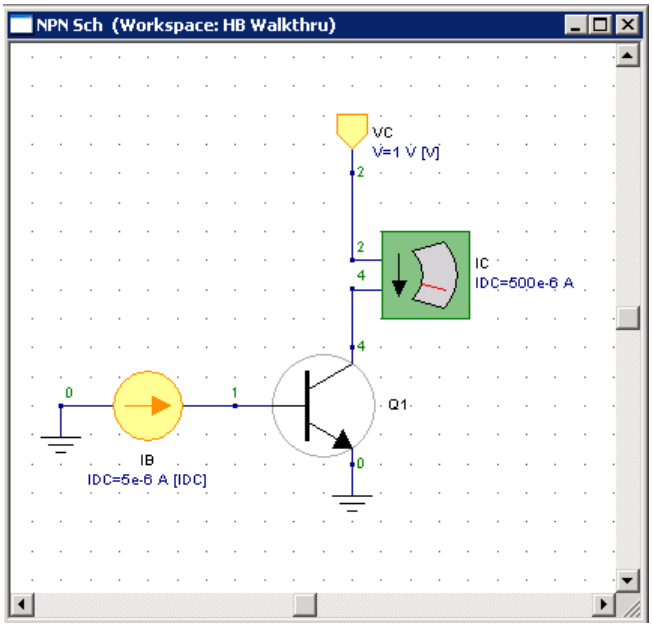
---

6. Create and equation block using the 'New' button on the workspace tree and find 'Add Equations...'. Define tunable variables 'V = ?1' and 'IDC = ?5e-6'. (Any error messages should disappear).



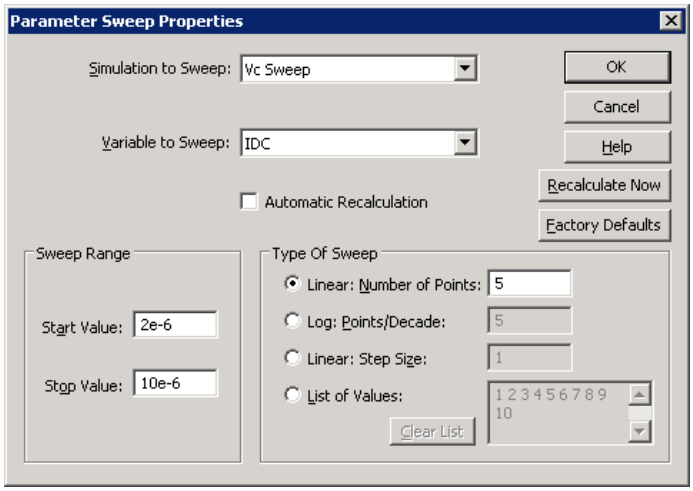
7. The schematic should look as follows:

DC curves schematic:



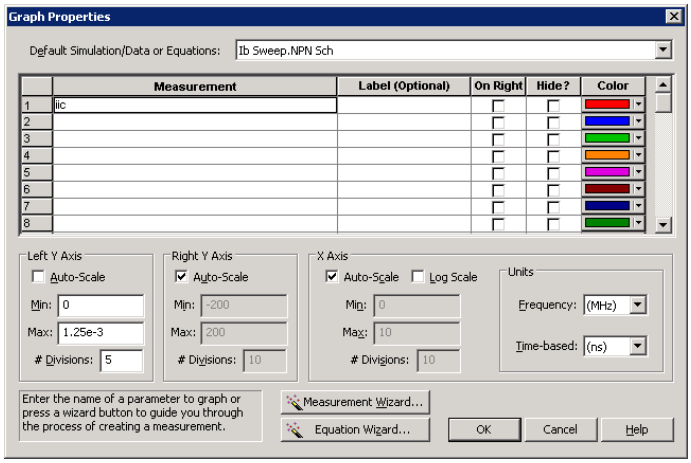
8. Create a DC analysis using the 'New' button and find 'Add DC Analysis..' under the 'Analysis sub-menu. Select 'Annotate' as shown.

9. Create a parameter sweep using the 'New' button and find 'Add Parameter Sweep...' under the 'Analysis' sub-menu. Name the sweep 'Vc Sweep'. Set the parameters up as shown.

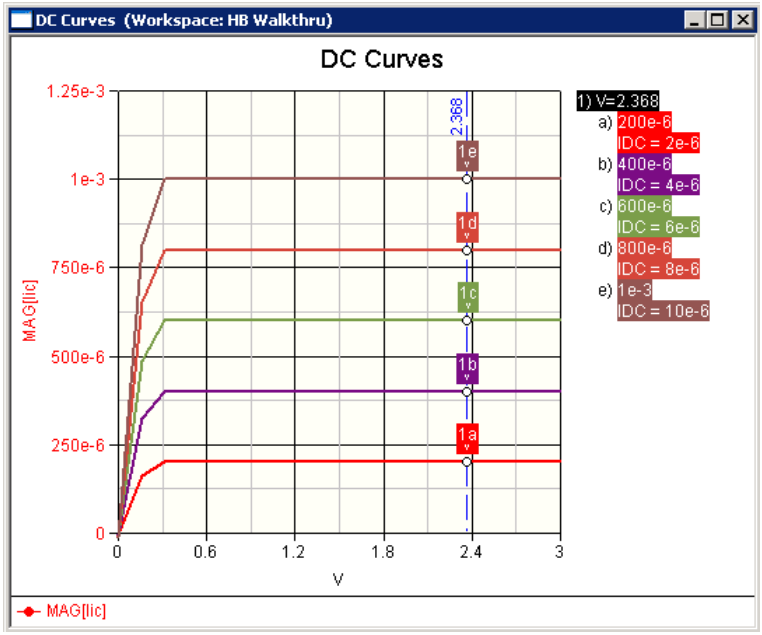


10. Create another parameter sweep and name it 'Ib Sweep' which will sweep our first sweep (Vc Sweep) and fill in the parameters as shown.

11. Create a rectangular graph using the 'New' button and find 'Add Rectrangular Graph...' under the 'Output' sub-menu.
Fill in the parameters as shown.

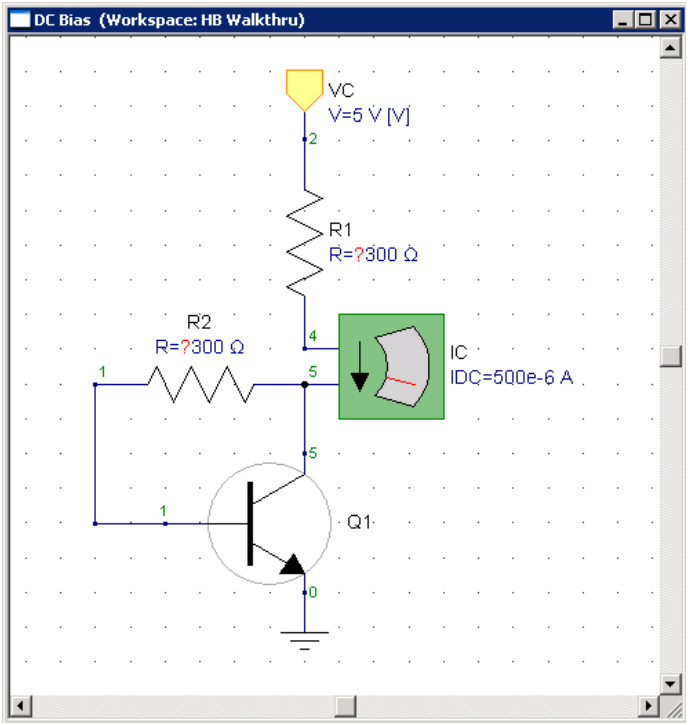


12. The graph should look as follows if we place a marker at V = 2.368 volts, the graphs will be labeled with appropriate values of base current (IDC).

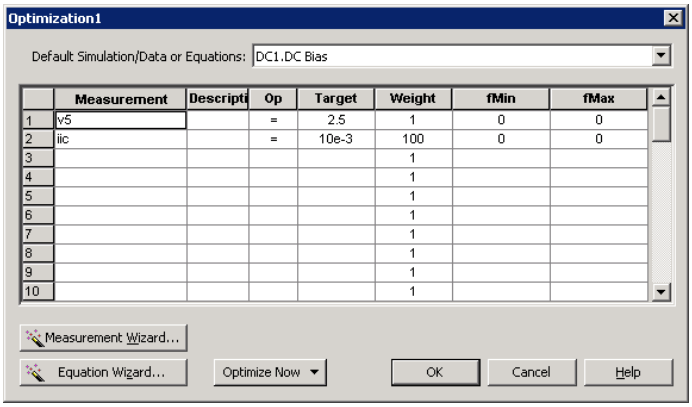


BIASING THE TRANSISTOR

1. For this example we chose to use a collector voltage of 2.5 volts and an Ic of 10mA.
2. Copy the original schematic (DC curves) and paste it into another schematic named DC Bias. From the curves above, the base current is about 0.010 ma for these conditions.
3. Modify the schematic to look as follows. Set the initial resistance to 300 ohms and make them both tunable by checking the 'Tune' box in the part properties for each resistor.



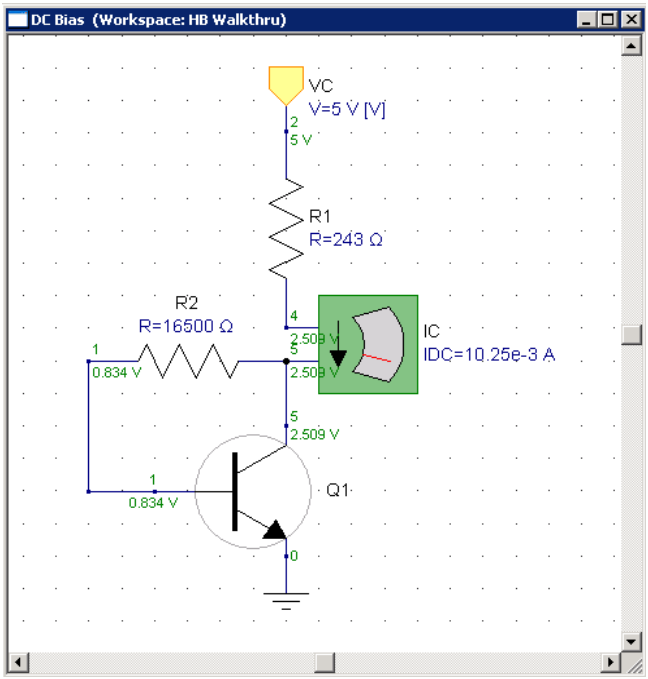
4. The supply voltage (V) should now be fixed to 5V. This can be done by removing the '?' in front of the variable specified in the equation block and changing the voltage from 1 to 5.
5. Create an optimization using the 'New' button and find 'Add Optimization...'. Set the values as follows. Your collector node number may be different than this example. If it is different then change it accordingly (i.e. if your collector node number is 2 then the measurement should be 'v2').



Note: We increased the weight of iic because we care more about optimizing this measurement than v2.

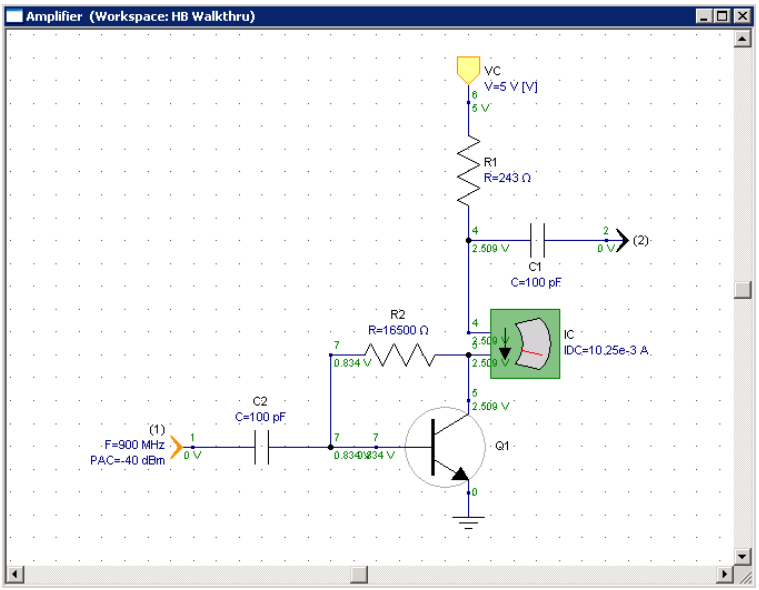
6. Run the optimization. When the desired level of performance is achieved stop the optimizer.

DC bias circuit with resistors set as non-tunable values:

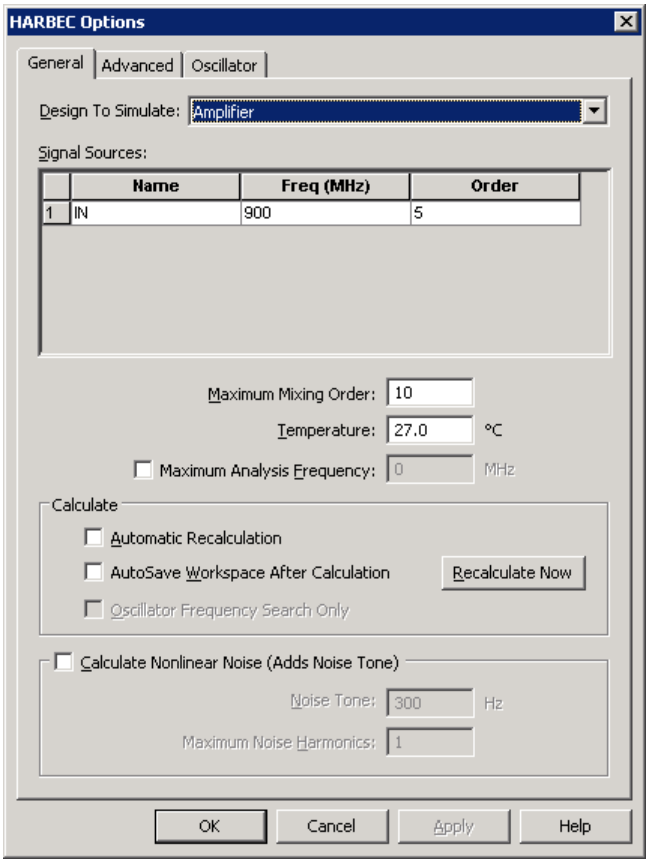


7. Copy this schematic into a new schematic named "Amplifier".
8. Add 100pF capacitors in series with the input and output ports as shown below.
9. Add an input port is an AC Power (PAC) input with a source frequency of 900MHz and a power output of -40dBm. Change its 'Designator' to 'IN'.

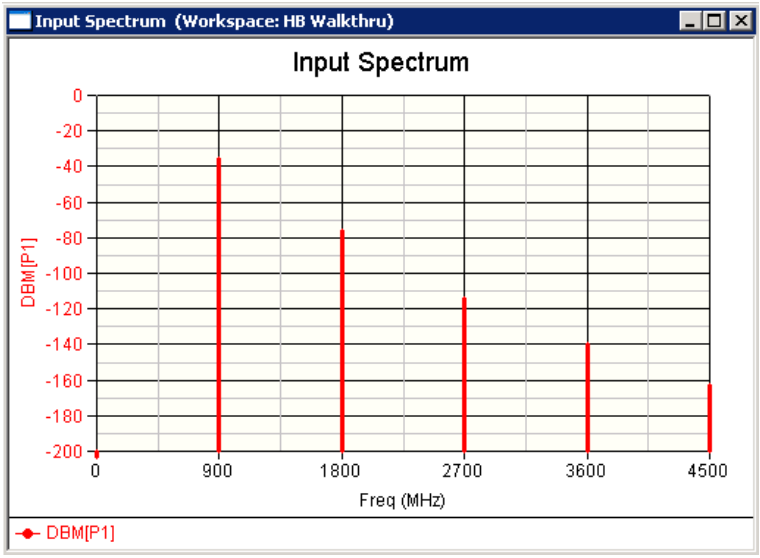
DC bias circuit with DC blocking caps and input/output ports:



10. In the Workspace Window, click the New Item button and select "Add Linear Analysis" in the Analysis submenu. Accept the default name. Click OK.
11. Click OK again to accept the default input values.
12. In the Workspace Window, click the New Item button and select "Add Smith Chart" in the Data Output submenu. Name the Graph **Match**.
13. Choose "Linear1.Amplifier" from the Default Simulation combo box.
14. Add **S11** and **S22** to the measurement list and press OK. You will see a Smith chart with input and output match.
15. Create a HARBEC simulation of the Amplifier schematic. Set simulation options as shown.

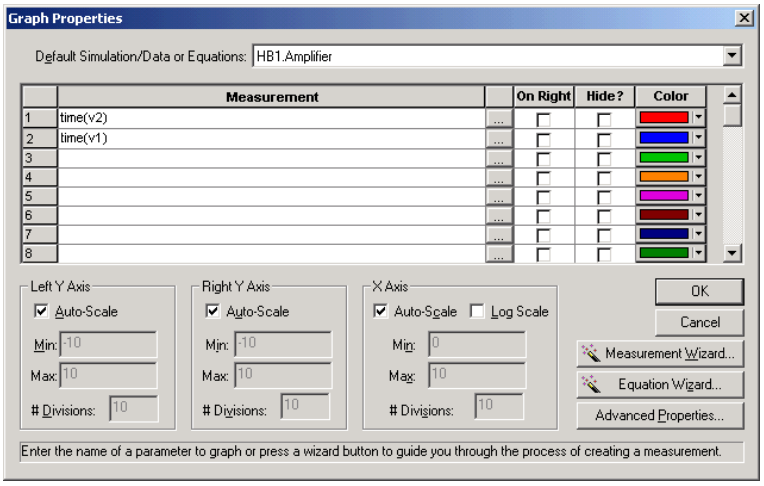


16. Create an output graph named 'Spectrum' to display power at port 1. Under *Measurement*, in the graph properties type 'P1' (power at node 1). The 'Simulations\Data' should be set to 'HB1.Amplifier'.

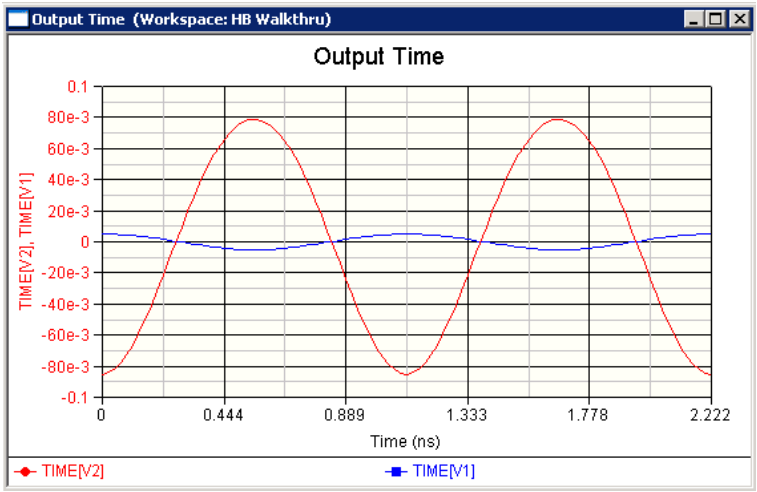


Note: When creating the output graph above, be sure to choose HB1.Amplifier as the default Simulation/Data or Equations.

17. Create an output graph. Under *Measurement,* in the graph properties type 'time(v2)' and 'time(v1)'.

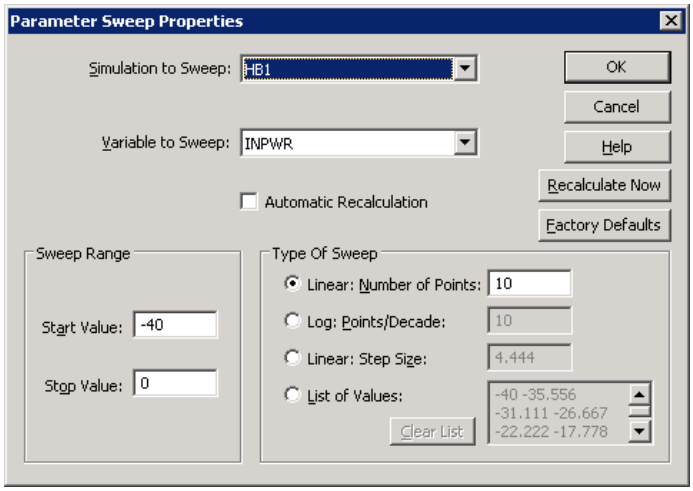


Note: Be sure to select HB1.Ampifier as the default simulation/data or Equations here.

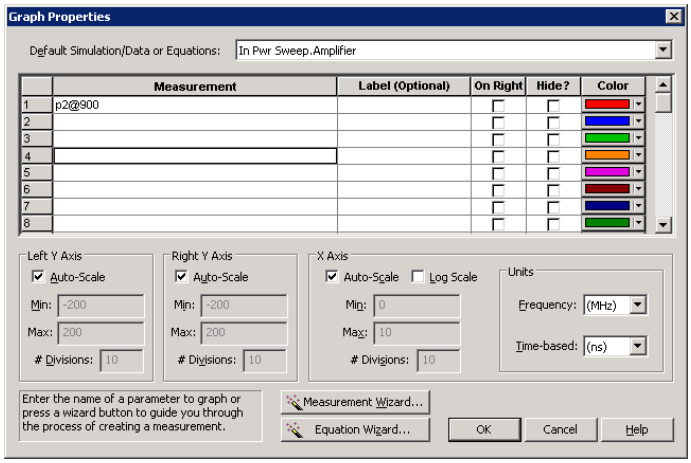


CREATING AN OUTPUT VS. INPUT POWER GRAPH

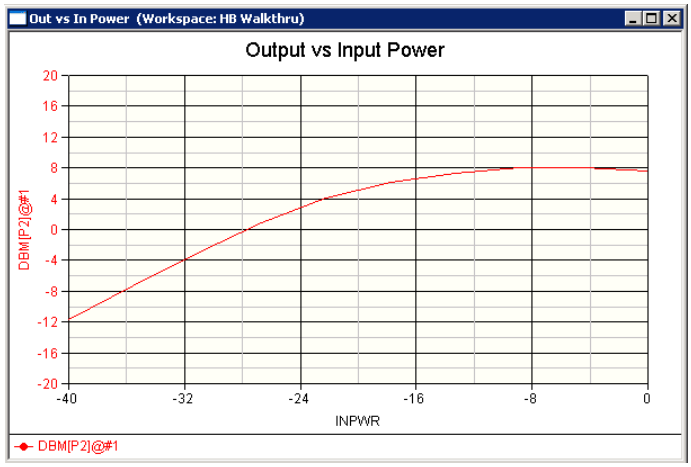
1. Create an new variable to sweep in the equation block as follows 'InPwr = ?-40'. Substitute this variable name for the -40 dBm AC power on the input port.
2. Create a parameter sweep and naming it "Input Power Sweep" and specifying the parameters as shown below.



3. Add a rectangular graph and specify the properties as shown.



4. The output graph should appear as follows.



Note: It is important to choose "Input Power Sweep.Amplifier" as the default simulation/data or equations field.

Chapter 3 Walkthrough: SPECTRASYS

Overview

SPECTRASYS is a spectral domain system simulator. Because of its unique implementation, it has several advantages over traditional simulators. The main focus of SPECTRASYS is to aid the user is analyzing and optimizing the RF performance of a chosen architecture which consists of two or more RF blocks or elements.

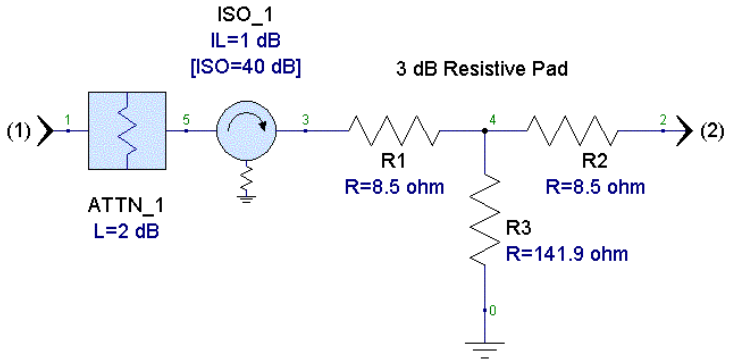
The best way to think about SPECTRASYS is to compare the SPECTRASYS schematic or block diagram to a circuit board and the SPECTRASYS simulation graph to a spectrum analyzer. Just like a circuit board, SPECTRASYS propagates every source and derived spectral component (harmonics, intermods, spurs, etc.) to every node in the system. The graph can then be set to examine the spectrum at any node in the system. Since a channel and a schematic path can be defined, the user can examine any one of over 30 spectrum integrated measurements along this user-defined path on a level diagram.

SPECTRASYS has many advantages over traditional system simulators

- SPECTRASYS is completely integrated into the GENESYS environment and provides the platform that ties all of the synthesis, circuit simulation, layout, electromagnetic simulation, and testing together.
- Any linear component can be placed in the system schematic along with any of over 45 RF behavioral models.
- Arbitrary topologies and multiple paths are automatically accounted for.
- The user can view full spectrums at any node in the system.
- Frequency dependent VSWR interactions between stages are automatically included.
- All measurements are channel based and are a result of spectrum integration.
- Level diagrams can display any of over 30 measurements along any user defined path.
- The origins and paths of all spectral components on every node can be easily identified.
- Broadband noise is readily analyzed and processed.

Creating a Schematic

The first step in creating a SPECTRASYS simulation is to create a schematic. For this walkthrough, we will create the following schematic:



The following circuit elements are used in this schematic:

- Input: Standard (*INP), on main toolbar (or press I)
- Attenuator, on system toolbar
- Isolator, on system toolbar
- Text ("3dB Resistive Pad") on main toolbar
- Resistors, on lumped toolbar (or press R)
- Ground and output, on main toolbar (or press G and O)

Note: Your node numbers may vary from the picture above depending upon how you draw the circuit.

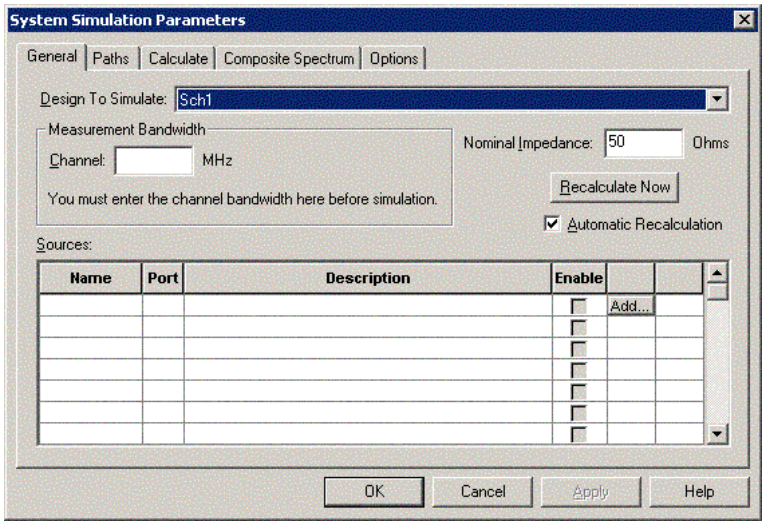
This simple circuit will illustrate the capability of SPECTRASYS to include lumped elements (unlike other types of system simulators).

Note: The walkthrough at this point is saved in Examples\SPECTRASYS\Walkthrough\1 - Create Schematic.WSP

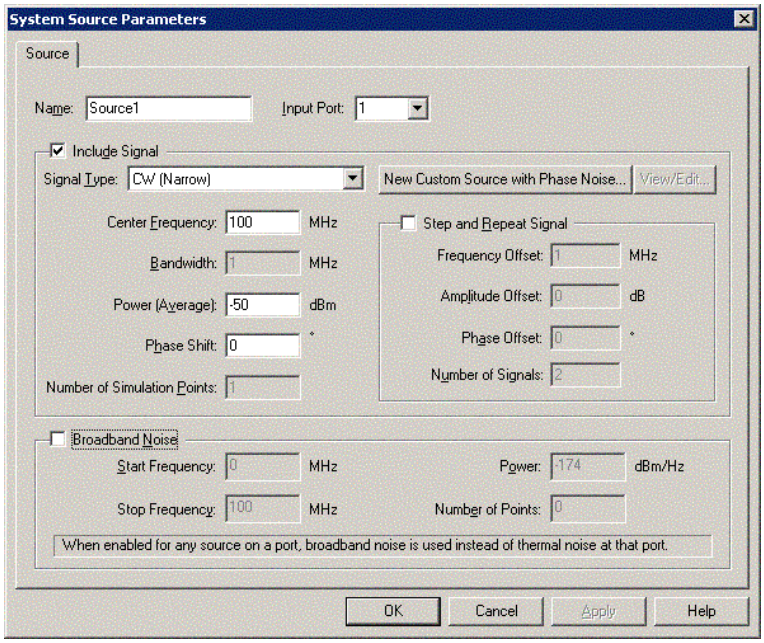
Adding a SPECTRASYS simulation

Next, we will add a SPECTRASYS simulation to the workspace. To add the SPECTRASYS simulation:

1. Right-click on the Simulations/Data tab in the workspace window.
2. Select "Add System Simulation". Accept the name "System1".
3. On the Settings tab, change the "Measurement Bandwidth/Channel" to 1 MHz.
4. Add a source by clicking on the 'Add' button in the source grid.

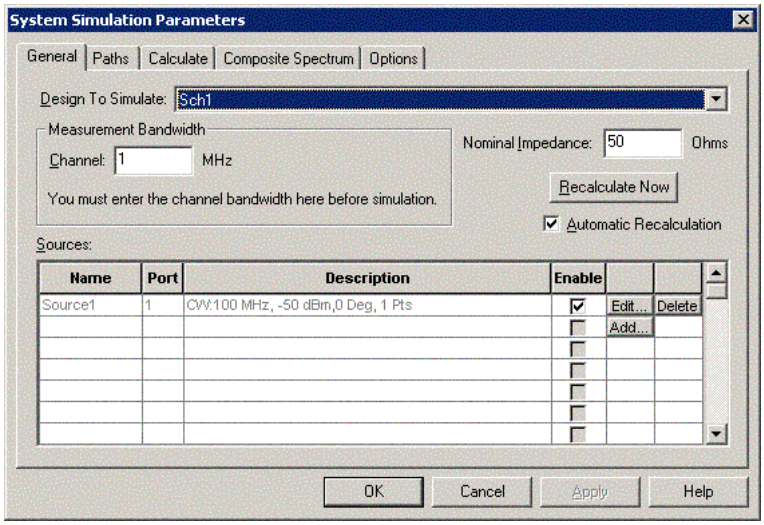


5. The 'Source' dialog box will open as shown.



6. Set the 'Signal Type' to 'CW (Narrow)'. The 'Center Frequency' to 100 MHz and the 'Power (Average)' to -50 dBm.

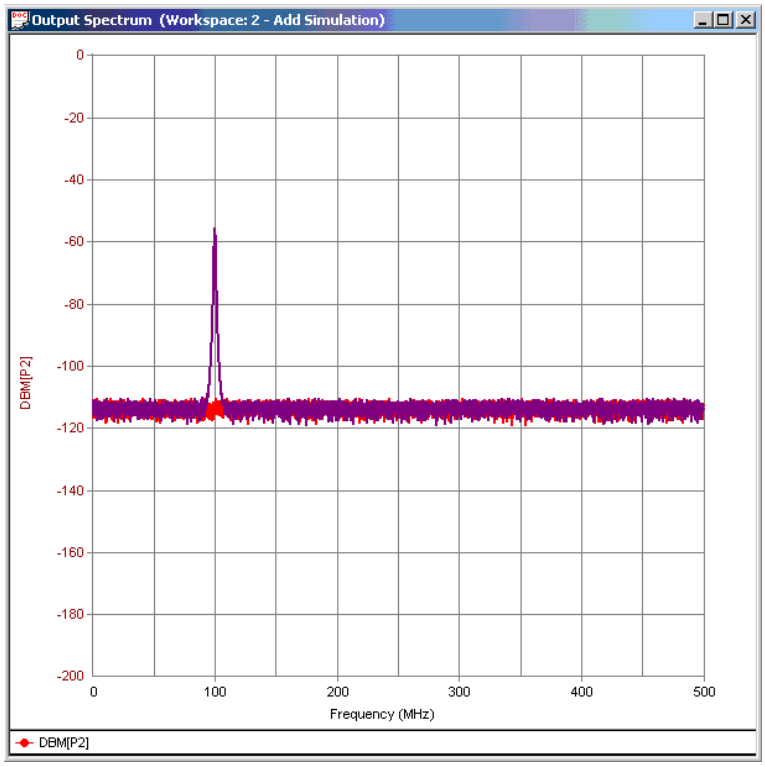
7. Click OK to accept the newly created source.



8. Click OK to accept the system simulation parameters. SPECTRASYS will automatically calculate.
9. Right-click on the Outputs tab in the workspace window.
10. Select "Add Rectangular Graph". Enter the name "Output Spectrum".
11. Click "Measurement wizard" to add a new measurement.
12. Select Simulation "System1 (Sch1) Composite Spectrum" and press Next.
13. Select Pport (power at a port/node), select item P2, and press Finish.
14. Click OK to close the Graph Properties dialog box. You will see a simple graph with the output spectrum.

Note: This graph will be easier to read if you make it larger than the default size.

15. To make this graph easier to see and understand, we can switch to spectrum analyzer mode. Double-click on "System1" in the workspace window.
16. Enter 200 MHz for "Ignore Spectrum/Frequency Above".
17. Click on the Composite Spectrum Tab. Check "Enable Analyzer Mode". Press OK. Your graph should now look more like a spectrum analyzer displaying the data, including random noise.

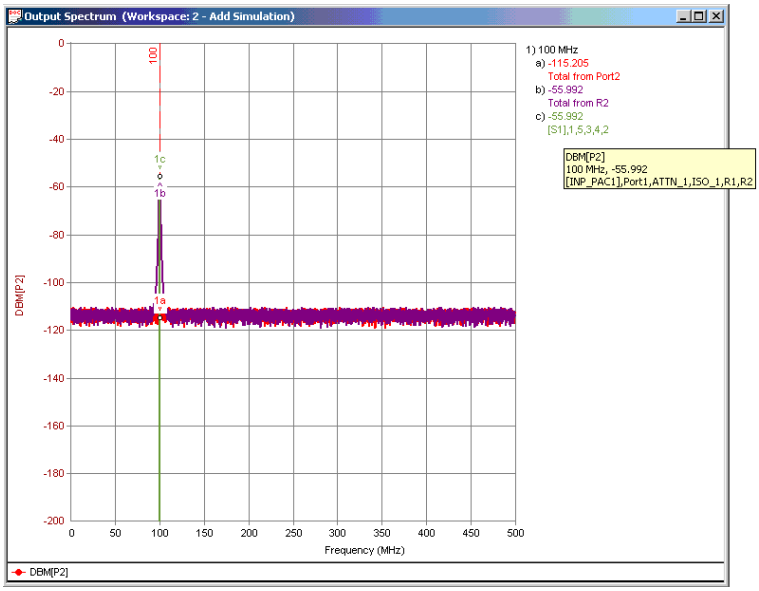


Note: The two total spectrums shown are the spectrums going *both directions* (Signal + noise from input; thermal noise only from output) at node two. To see only the spectrum going one direction, you can plot the spectrum along a path.

Note that you can see all of the pieces that combine to make this composite signal, and can use markers and fly-over help to determine exactly where the signals came from. To do this:

18. Double-click on "System1" in the workspace window.
19. Click on the Composite Spectrum Tab.
20. Check "Signals" and "Intermods and Harmonics".
21. Select the radio button "Individual Components".
22. Press OK. You will see a graph like the one below. Placing a marker on the peak will show the source of the signal, and moving the cursor over the marker will give more details. In this case, the green signal started at the input, INP_PAC1, went out the port, and through the listed nodes and elements.

Note: You can zoom in easily on the graph using your mouse wheel or using the zoom buttons on the toolbar.



Note: The walkthrough at this point is saved in Examples\SPECTRASYS\Walkthrough\2 - Add Simulation.WSP

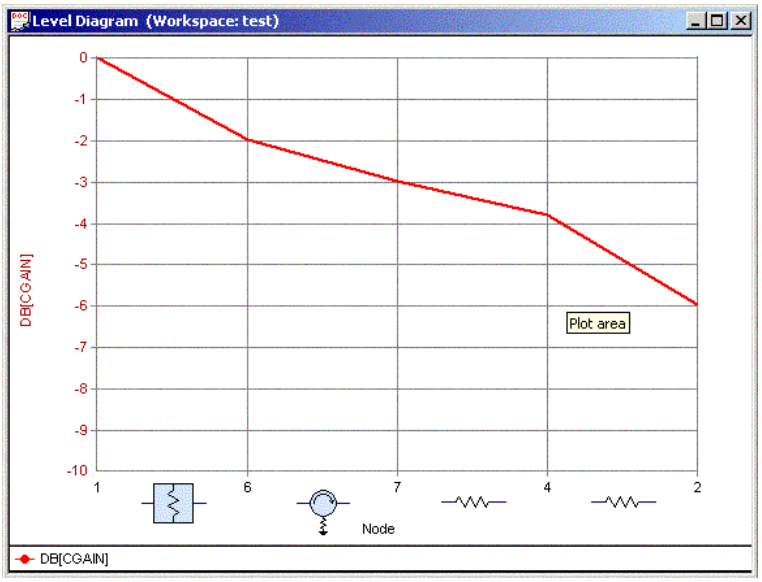
Level Diagrams

Another tool in SPECTRASYS is the level diagram. To create a level diagram:

1. You must first add a path to the system simulation. Double-click on "System1" in the workspace window.
2. On the "Paths" tab, click "Add Path". Enter the beginning path node 1, and ending path node 2. Enter the name "Forward".

Note: You can also click "Add Primary Paths" to automatically add all paths.

3. Click OK.
4. Right-click on the Outputs tab in the workspace window.
5. Select "Add Rectangular Graph". Enter the name "Level diagram".
6. Click "Measurement wizard" to add a new measurement.
7. Select Simulation "System1 (Sch1) Path Forward" and press Next.
8. Choose measurement CGAIN (Cascaded Gain) and press Finish. Press OK. You will see a level diagram similar to the one shown below. This diagram shows the total cascaded gain through the system at each node.

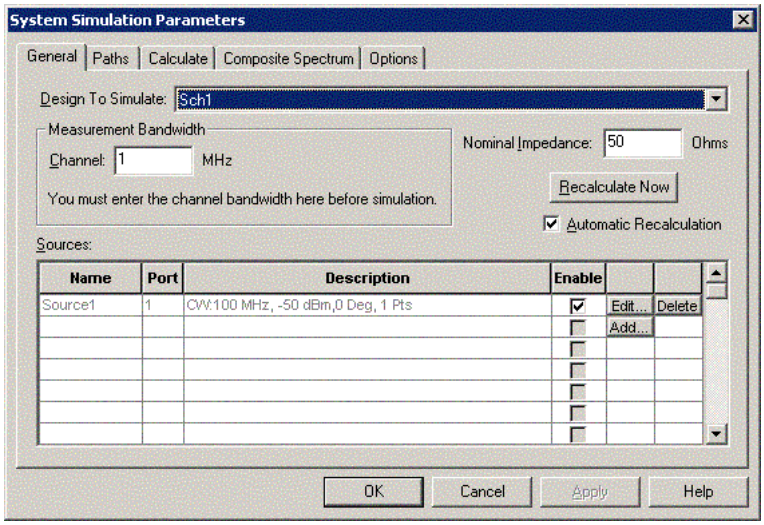


Note: The walkthrough at this point is saved in Examples\SPECTRASYS\Walkthrough\3 - Level Diagram.WSP

System Simulation Parameters - Tuning Parameters

Like the rest of the GENESYS environment, SPECTRASYS features real-time tuning. In addition to the tuning of element values, all parameters in the system simulation dialog box can be tuned.

1. Double-click "System1" in the workspace window to open the system simulation.
2. Click the 'Edit' button at the end of the row where the CW source is located.



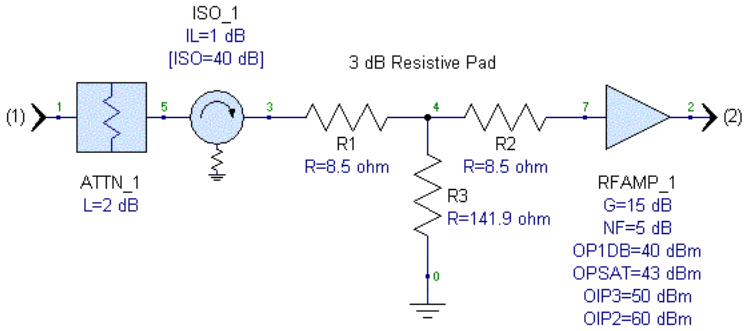
3. Type a '?' in front of the -50 for the source power. The power of the source should look like '?-50'.
4. Click OK to close the source dialog box.
5. Type ?1 (question mark followed by 1) in the channel measurement bandwidth.
6. Click OK to accept the changes to the system simulation.
7. Double-click "Output Spectrum" in the workspace window to open the output spectrum.
8. Tune the input power and bandwidth.

Try tuning the resolution bandwidth to 3 MHz and the input power to 0 dBm. This will allow you to see a good picture of the resolution bandwidth. It also clearly distinguishes the power coming from the input (containing a signal) and the power coming from the output (containing only noise).

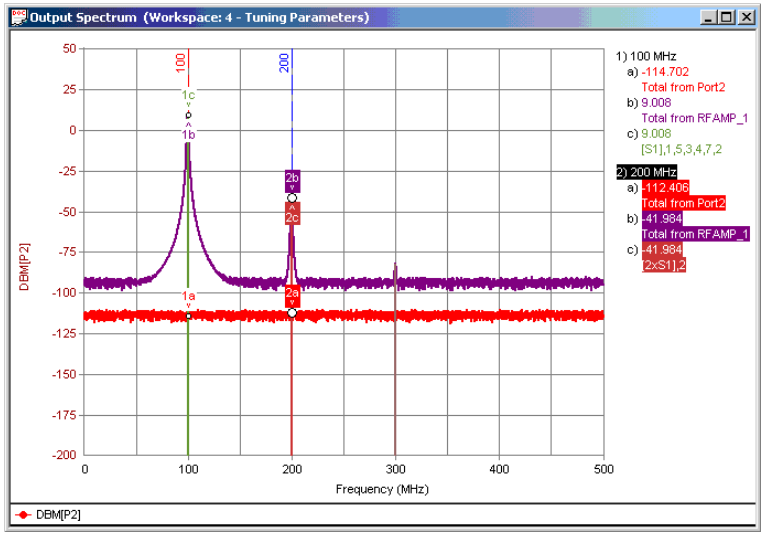
Note: The walkthrough at this point is saved in Examples\SPECTRASYS\Walkthrough\4 - Tuning Parameters.WSP

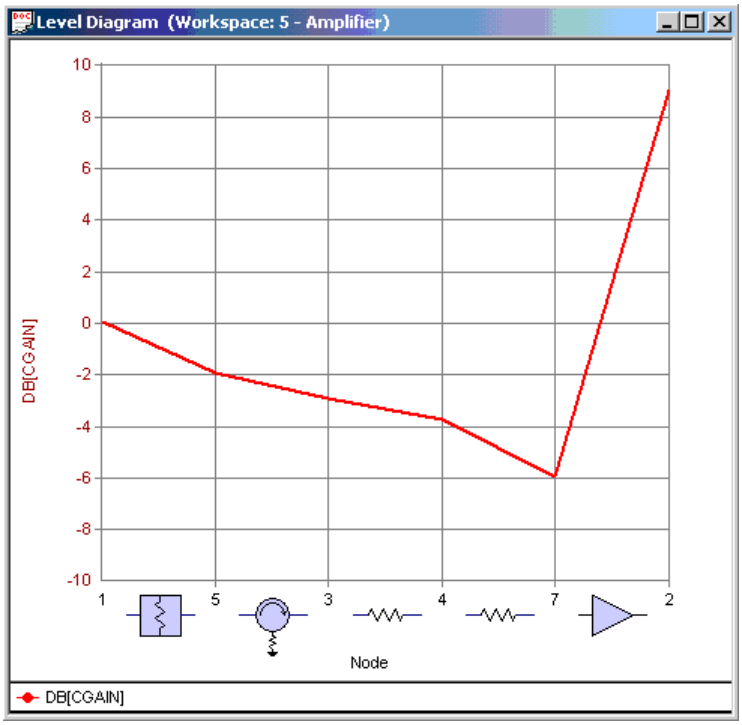
Add an Amplifier

Let's add an amplifier to this circuit. Modify the schematic to look like the following. Don't forget that you can hold Alt down while moving the output to break the connection to the resistor. The RF Amplifier is found on the System toolbar.



1. If you haven't been saving your work, you should save your file now.
2. Open the "Output Spectrum" graph. With the input power tuned to 0 dBm and the measurement bandwidth set to 1 MHz, you should see the following graphs. Note that the noise has risen 20 dB: 15 dB from the amplifier gain, and 5 dB from the amplifier noise figure.





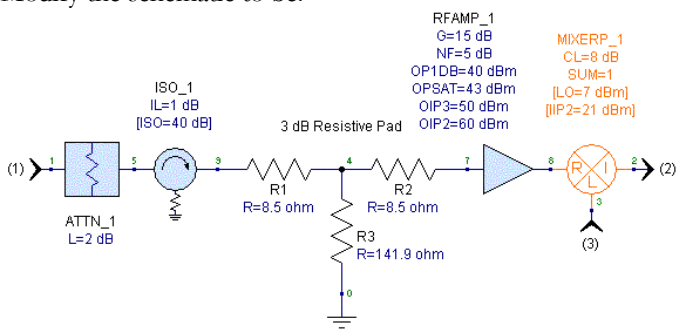
Note that on the output spectrum, you can see the harmonics. Try passing your mouse over the harmonic to see the level and the source.

> **Note:** The walkthrough at this point is saved in Examples\SPECTRASYS\Walkthrough\5 - Amplifier.WSP

Add a Mixer

Next, we will mix our 100 MHz signal up to 2 GHz using a 1.9 GHz LO and a mixer.

1. Modify the schematic to be:



Note that we changed the units on the LO to GHz. This is very easy to do inside

the schematic element dialog box. The mixer is the "Passive Mixer" found on the System toolbar.

2. Looking at the output spectrum, you will see the RF and LO sneaking through. You will also see intermods which were generated.
3. Add another graph called "Input Spectrum". Add measurement P1 from "System1.Composite". You will see that the LO has come back towards the input and is being sent backwards along the RF chain. You will also see that the second harmonic generated at the amplifier comes back towards the input. Seeing these "Sneak paths" is one of the more powerful features of SPECTRASYS.

Note: The mixer is orange because the mixer has a local error after a simulation is performed. The local error can be viewed by right-clicking on the mixer and selecting 'Show Local Errors'. This error shows that the mixer is being overdriven.

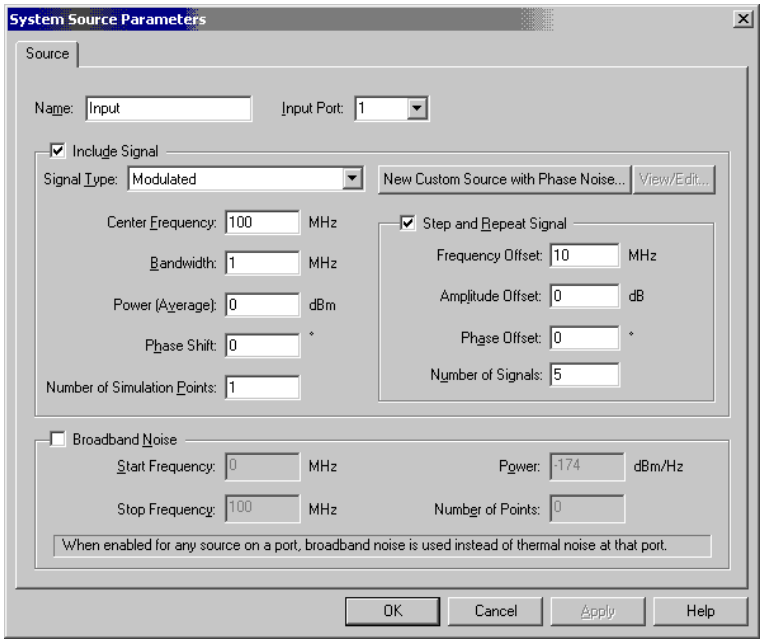
Note: You must have "Show Contributors/Signals" checked to see the sneak paths below noise.

Note: The walkthrough at this point is saved in Examples\SPECTRASYS\Walkthrough\6 - Mixer.WSP

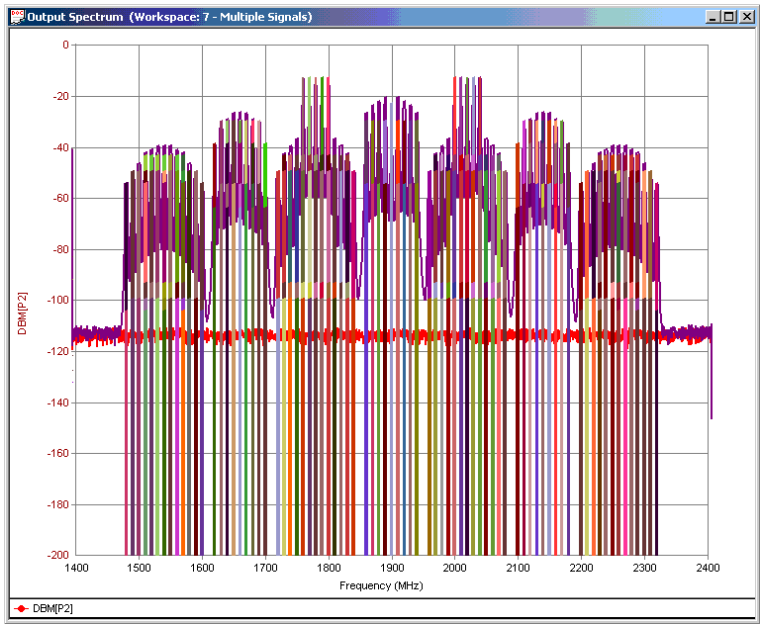
Multiple Signals

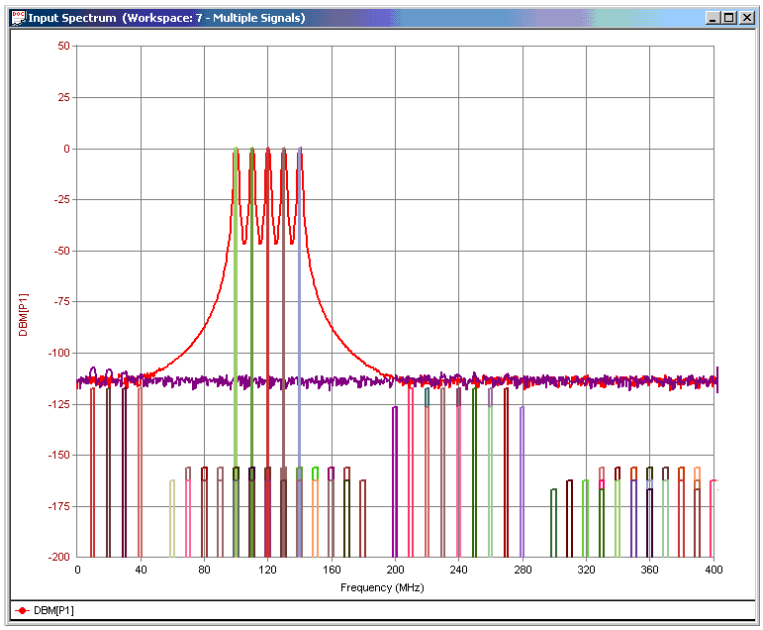
SPECTRASYS can easily handle many signals simultaneously. We will add more signals to the input port to see the impact on the system.

1. Double-click on the system simulation.
2. Uncheck the "Enable" checkbox on the first source line to disable the 100 MHz CW signal coming into Port 1.
3. Click on the Add button on the third (empty) source line.
4. Make the source box look like:



5. Press OK to close the Source box.
6. On the first Forward Path line in the Paths tab, enter "100" for the channel frequency. This is necessary because we now have many signals coming into the input, and we need to specify which one to track for the level diagram.
7. Click OK to close the system dialog and start simulation.
8. Zooming in on the input and output will show the following spectra. (You can either use a mouse wheel or the zoom icons on the toolbar.) Notice all of the junk coming in and out of the circuit!





Note: The completed walkthrough is saved in Examples\SPECTRASYS\Walkthrough\7 - Multiple Signals.WSP

Chapter 4 Linear Simulation

Overview

Linear simulation calculates S-parameters and noise parameters of a circuit. It is a small signal analysis that assumes that the circuit is operating in the linear region. Active devices such as transistors and diodes can be modeled either with S-parameters (measured or provided by a manufacturer) or a nonlinear model (requires HARBEC). If a nonlinear model is used, GENESYS automatically runs a DC analysis to determine the circuit operating point, linearizes the nonlinear circuit around the operating point, and uses that linear model in the analysis.

To add a linear simulation:

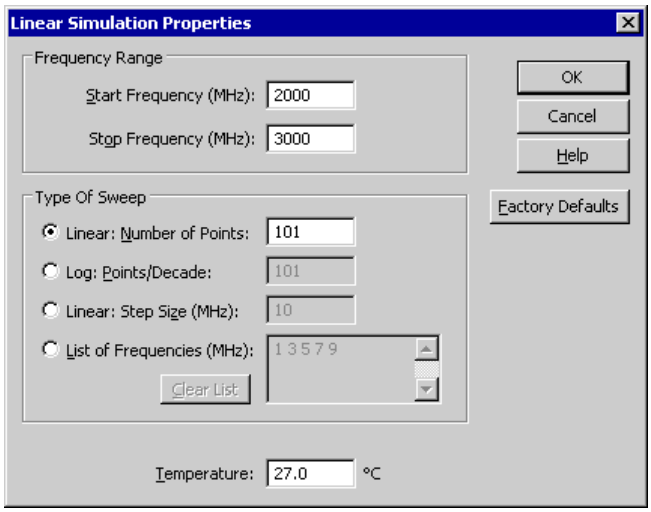
1. Right-click the Simulation/Data node on the Workspace Window.
2. Select "Add Linear Simulation".
3. Complete the Linear Simulation Properties dialog.
4. Add a graph or other output and a measurement to see the results.

See also:

Measurements (later in this manual)
Outputs Overview, User's Guide

Linear Simulation Properties

To open: double-click or create a Linear Simulation.



Frequency Range

- **Start Frequency** - The lower bound (minimum frequency) of the linear simulation.
- **Stop Frequency** - The upper bound (maximum frequency) of the linear simulation.

Type of Sweep

- **Linear: Number of Points** - Allows specification of start and stop frequencies, and number of points
- **Log: Points/Decade** - Allows specification of start and stop frequencies, and number of points
- **Linear: Step Size** - Allows specification of start and stop frequencies, and space between points.
- **List of Frequencies** - Allows the explicit specification of analysis frequencies. These points are entered into the List of Frequencies box separated by spaces.

Factory Defaults - Resets the sweep properties to the original, known-good settings.

Temperature - The ambient temperature of the linear simulation.

S Parameters

Overview

The purpose of this section is to summarize network analysis concepts and to define some of the parameters plotted by GENESYS. For further details on measurements, see the Measurements section of this manual.

Networks are considered as "black boxes". Because the networks are assumed to be linear and time invariant, the characteristics of the networks are uniquely defined by a set of linear equations relating port voltages and currents. A number of network parameter types have been developed for this purpose, including H, Y, Z, S, ABCD, and others. These parameters may be used to compute and display network responses and to compute quantities useful for circuit design such as Gmax (maximum gain) and gain circles. Each parameter type has advantages and disadvantages. Carson [1] and Altman [2] provide additional information.

S-Parameter Basics

S-parameters have earned a prominent position in RF circuit design, analysis, and measurement. Parameters used earlier in RF design, such as Y-parameters, require opens or shorts on ports during measurement. This is a nearly impossible constraint for high-frequency broadband measurements. Scattering parameters [3, 4] (S-parameters) are defined and measured with the ports terminated in a characteristic reference impedance. Modern network analyzers are well suited for measuring S-parameters. Because the networks being analyzed are often employed by insertion in a transmission medium with a common characteristic reference impedance, S-parameters have the additional advantage

that they relate directly to commonly specified performance parameters such as insertion gain and return loss.

Two-port S-parameters are defined by considering a set of voltage traveling waves. When a voltage wave from a source is incident on a network, a portion of the voltage wave is transmitted through the network, and a portion is reflected back toward the source. Incident and reflected voltage waves may also be present at the output of the network. New variables are defined by dividing the voltage waves by the square root of the reference impedance. The square of the magnitude of these new variables may be viewed as traveling power waves.

$|a_1|^2$ = incident power wave at the network input
$|b_1|^2$ = reflected power wave at the network input
$|a_2|^2$ = incident power wave at the network output
$|b_2|^2$ = reflected power wave at the network output

These new variables and the network S-parameters are related by the expressions:

$$b_1 = a_1S_{11} + a_2S_{12}$$

$$b_2 = a_1S_{21} + a_2S_{22}$$

$$S_{11} = b_1/a_1,\ a_2 = 0$$

$$S_{12} = b_1/a_2,\ a_1 = 0$$

$$S_{21} = b_2/a_1,\ a_2 = 0$$

$$S_{22} = b_2/a_2,\ a_1 = 0$$

Terminating the network with a load equal to the reference impedance forces $a_2 = 0$. Under these conditions

$$S_{11} = b_1/a_1$$

$$S_{21} = b_2/a_1$$

S_{11} is then the network input reflection coefficient and S_{21} is the gain or loss of the network.

Terminating the network at the input with a load equal to the reference impedance and driving the network from the output port forces $a_1 = 0$. Under these conditions

$$S_{22} = b_2/a_2$$

$$S_{12} = b_1/a_2$$

S_{22} is then the network output reflection coefficient and S_{12} is the reverse gain or loss of the network.

Linear S-parameters are unitless. Since they are based on voltage waves, they are converted to decibel format by multiplying the log of the linear ratio by 20. It is not always obvious whether an author is refering to linear or decibel parameters. To avoid this confusion, the book Oscillator Design and Computer Simulation and Versions 5.4 and earlier of SUPERSTAR use C for linear S-parameters and S for the decibel form. This is somewhat unconventional. Version 6.0 and later of GENESYS also supports the convention

MAG[S21] which is linear and DB[S21] which is the decibel form. With reflection parameters, the linear form is often refered to as a relection coefficient and the decibel form as return loss.

S_{11}(dB)=input reflection gain=20 log S_{11}

S_{22} (dB)=output reflection gain=20 log S_{22}

S_{21}(dB)=forward gain=20log S21

S_{12}(dB)=reverse gain=20log S12

S_{21} and S_{12} are the forward and return gain (or loss) when the network is terminated with the reference impedance. The gain when matching networks are inserted at the input, output, or both is described later.

S_{11} and S_{22} coefficients are less than 1 for passive networks with positive resistance. Therefore, the input and output reflection gains, S_{11} and S_{22}, are negative decibel numbers. Throughout Eagleware material, the decibel forms S_{11} and S_{22} are referred to as return losses, in agreement with standard industry convention. To be mathematically correct, they have been left as negative numbers. As such, the rigorous convention would be to call them return gain.

Input VSWR ($VSWR_1$) and S_{11} are related by

$$VSWR_1 = (1 + |S_{11}|) / (1 - |S_{11}|)$$

The output VSWR is related to S_{22} by an analogous equation. A circle of constant radius centered on the Smith chart is a circle of constant VSWR. The complex input impedance is related to the input reflection coefficients by the expression:

$$I_1 = Z_o (1 + S_{11}) / (1 - S_{11})$$

The output impedance is similarly related to S_{22}.

Stability

Because S12 of devices is not zero, a signal path exists from the output to the input. This feedback path creates an opportunity for oscillation. The stability factor, K, is

$$K = (1 - |S_{11}|^2 - |S_{22}|^2 + |D|^2) / (2 |S_{12}| |S_{21}|)$$

where

$$D = S_{11}S_{22} - S_{12}S_{21}$$

From a practical standpoint when K>1, S_{11}<1, and S_{22}<1, the two-port is unconditionally stable. These are often stated as sufficient to insure stability. Theoretically, K>1 is insufficient to insure stability, and an additional condition should be satisfied. One such parameter is B1 which should be greater than zero.

$$B1 = 1 + |S_{11}|^2 - |S_{22}|^2 - |D|^2 > 0$$

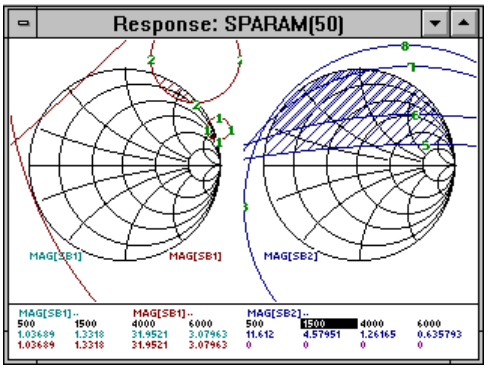
Stability circles may be used for a more detailed analysis. The load impedances of a network which ensure that S_{11}<1 are identified by a circle of radius R centered at C on a Smith chart. The output plane stability circle is

$C_{out} = (S_{22} - DS_{11}^{*})^{*} / (|S_{22}|^2 - |D|^2)$, $R_{out} = | S_{12}S_{21} / (|S_{22}|^2 - |D|^2) |$

This circle is the locus of loads for which $S_{11} = 1$. The region inside or outside the circle may be the stable region.

The input plane stability circle equations are the same as the output plane equations, with 1 and 2 in the subscripts interchanged.

Shown in the figure below are the input plane stability circles on the left and the output plane stability circles on the right for the Avantek AT10135 GaAsFET. The shaded regions are potentially unstable. At the input, the stability circle with marker 1 indicates sources with a small resistive component and inductive reactance of about 200 ohms are unstable. Circles 2 and 3 are also unstable with low resistance and certain inductive source impedances. At the output plane on the right, at 500 MHz, a wide range of inductive loads is potentially unstable.



When designing an amplifier the first step is to examine the stability circles of the device without the matching circuit present. The grounding which will be present at the emitter or source should be included in the analysis. This stability data is used to 1) add stabilizing components such as shunt input and output resistors for bipolars or inductance in the source path for GaAsFETs and to 2) select an input and output matching network topology which properly terminates the device (at low and high frequencies) for stability.

In the example above, matching networks with a small series capacitor adjacent to the device would insure capacitive loads at low frequencies, thus enhancing stability. This is probably sufficient for the input. However, considering that device S-parameter data is approximate and since the output plane of this device is more threatening, it would be prudent to stabilize this device in addition to using series capacitors.

Note: Stability should be checked not only at the amplifier operating frequencies, but also over the entire frequency range for which S-Parameter data is available.

Matching

One definition of network gain is the transducer power gain, Gt: Transducer power gain is the power delivered to the load divided by the power available from the source.

Gt = P(delivered-to-load) / P(available-from-source)

Other gain definitions include the power gain, Gp, and the available power gain, Ga.

Gp = P(delivered-to-load) / P(input-to-network)

Ga = P(available-from-network) / P(available-from-source)

The S-parameter data for the network is measured with a source and load equal to the reference impedance. If the network is not terminated in the reference impedance, Gt can be computed from the reflection coefficients of the terminations on the network and the S-parameters of the network. At this point we have multiple sets of reflection coefficients: those of the terminations and S11 and S22 of the network. To avoid confusion the termination reflection coefficients are given a different symbol, G.

The transducer power gain with the network inserted in a system with arbitrary source and load reflection coefficients is [4]:

$$Gt = (|S_{21}|^2 (1 - |R_s|^2)(1 - |R_L|^2)) / |(1 - S_{11}R_S)(1 - S_{22}R_L) - S_{21}S_{12}R_LR_S|^2$$

where

R_S = reflection coefficient of the source
R_L = reflection coefficient of the load

If and are both zero, then

$$Gt=S_{21}$$

or

$$Gt(dB)=20\log S_{21}=S_{21}(dB)$$

Therefore, when a network is installed in a system with source and loads equal to the reference impedance, S_{21} is the network transducer power gain in decibels.

Because S_{11} and S_{22} of a network are not in general zero, a portion of the available source power is reflected from the network input and is dissipated in the source. The insertion of a lossless matching network at the input (and/or output) of the network could increase the gain of the overall system if reflections toward the source were reduced. Shown below is a two-port network with lossless matching networks inserted between the network and the source and load.

GMAX and MSG

When the input and output networks are simultaneously designed for maximum gain, there is no reflection at the source or load. The maximum transducer power gain, G_{max}, is given by

$$G_{max} = (|S_{21}| / |S_{12}|) * (K - \text{sqrt}(K^2 - 1))$$

The maximum stable gain, MSG, is defined as G_{max} with K=1. Therefore

$$MSG = |S_{21}| / |S_{12}|$$

A GENESYS plot of GMAX shows G_{max} when K>1 and MSG when K<1.

Again, acheiving this maximum gain requires that the input network is designed such that R_S is the complex conjugate of S_{11} and R_L is the complex conjugate of S_{22}. GENESYS returns the required reflection coefficients, impedance and admittance for the input and output networks as GM1, GM2, ZM1, ZM2, YM1 and YM2, respectively.

The Unilateral Case

Historically, to simplify the complex equation for Gt in the previous section on matching, S_{12} was set to zero. At higher frequencies, where the device S_{12} is typically larger, this assumption is less valid. The assumption simplifies manual and graphical design but is unnecessary in modern computer-assisted design. The assumption also allows factoring the above equation into terms that provide insight into the design process. If S12 =0, then

$$Gtu = (|S_{21}|^2 (1 - |R_s|^2)(1 - |R_L|^2)) / |(1 - S_{11}R_S)(1 - S_{22}R_L)|^2$$

where

Gtu=unilateral transducer power gain

When both ports of the network are conjugately matched, and $S_{12} = 0$,

$$Gtu = |S_{21}|^2 / ((1 - |S_{11}|^2)(1 - |S_{22}|^2))$$

The first and third terms indicate the gain increase achievable by matching the input and output, respectively. If S_{11} or S_{22} approach 1, substantial gain improvement is achieved by matching. Matching not only increases the network gain, but reduces reflections from the network.

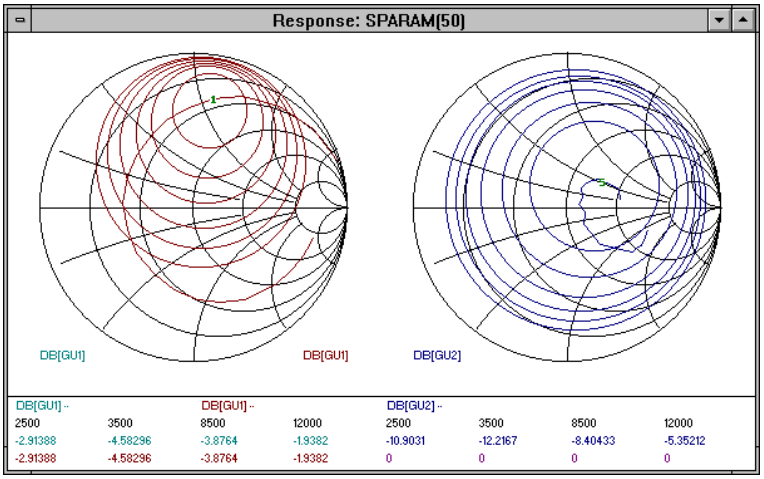
When network gain flatness across a frequency band is more desirable than minimum reflections, the lossless matching networks are designed to provide a better match at frequencies where the two-port gain is lower. By careful design of amplifier matching networks, it is possible to achieve a gain response flat within fractions of a decibel over a bandwidth of an octave or more.

Gain Circles

When the device is complex conjugately matched, the transducer gain is Gmax and if the device is terminated with the same resistance used to measure the device S-parameters the transducer gain is S_{21}. The gain with arbitrary terminations can be visualized on the Smith chart using gain circles.

SUPERSTAR plots three forms of gain circles: transducer gain unilateral circles, GU1 for the input network and GU2 for the output network, power gain output network circles, GP, and available gain input network circles, GA.

Shown below are the input and output unilateral transducer gain circles, GU1 and GU2, of the Avantek AT10135 GaAsFET transistor. SUPERSTAR circles are plotted at the frequency of the first marker, in this case 2500 MHz. Marker 1 is plotted at the center of the smallest circle, the point of maximum gain. The gain at the circumference of each circle of increasing radius is 1 dB lower than the previous inside circle.



The arc which is orthogonal to the gain circles is the locus of smallest circle center points from the lowest to highest sweep frequency. Tuning the first marker frequency moves the center of the circles along this arc.

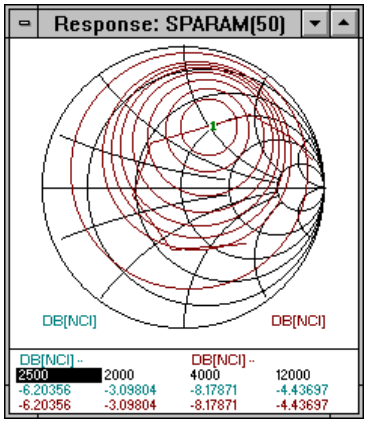
Notice that a complex conjugate match at the input improves the gain by over 3 dB in relation to an unmatched 50 ohm source impedance. However, matching the output provides less than 1 dB gain improvement. An examination of the device S-parameter data at 2500 MHz reveals that the output is originally closer matched to 50 ohms and it is not surprising that a matching network would be less beneficial.

Noise Circles

To achieve the best available noise figure from a device the correct impedance must be presented to the device. The impedance resulting in the best noise performance is in general neither equal to 50 ohms or the impedance which results in minimum reflection at the source.

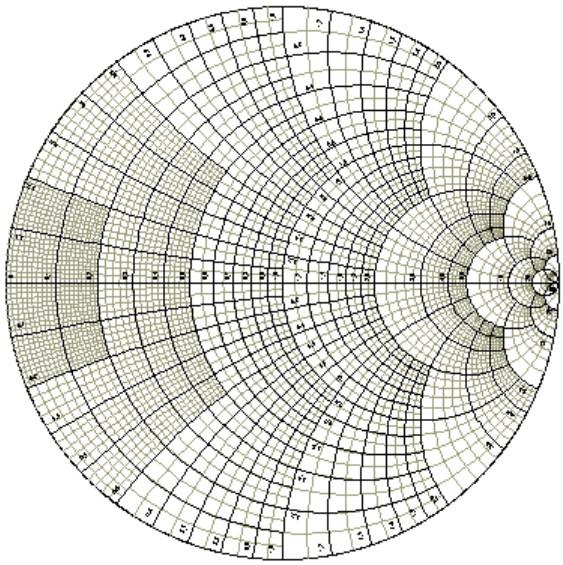
The Avantek AT10135 GaAsFET transistor S-parameter data given earlier includes noise data. This data is comprised of four numbers for each frequency. These numbers are NFopt(dB), the optimum noise figure when correctly terminated, Gopt magnitude and angle, the terminating impedance at the device input which acheives NFopt and Rn/Zo, a sensitivity factor which effects the radius of the noise circles.

Noise circles plotted by SUPERSTAR for the AT10135 at 2500 MHz are given below. Circles of increasing radius plotted by GENESYS represent noise figure degredations of 0.25, 0.5, 1, 1.5, 2, 2.5, 3 and 6 dB. In this case, direct termination of the device with a 50 ohm source results in a degredation of the noise figure of 1 dB. The arc orthogonal to the circles is the locus of Gopt versus frequency.



Smith Chart

In 1939, Philip H. Smith published an article describing a circular chart useful for graphing and solving problems associated with transmission systems [36]. Although the characteristics of transmission systems are defined by simple equations, prior to the advent of scientific calculators and computers, evaluation of these equations was best accomplished using graphical techniques. The Smith chart gained wide acceptance during the development of the microwave industry. It has been applied to the solution of a wide variety of transmission system problems, many of which are described in a book by Philip Smith [37]. The Smith chart as displayed by GENESYS is shown in below. Labels for normalized real and reactive components are added.



The design of broadband transmission systems using the Smith chart involves graphic constructions on the chart repeated for selected frequencies throughout the range of interest. Although the process was a vast improvement over the use of a slide rule, it is

tedious. Modern interactive computer programs with high-speed tuning and optimization procedures are much more efficient. However, the Smith chart remains an important tool for instructional use and as a display overlay for computer-generated data. The Smith chart provides remarkable insight into transmission system behavior.

The standard unity-radius impedance Smith chart maps all positive resistances with any reactance from - to + onto a circular chart. The magnitude of the linear form of S_{11} or S_{22} is the length of a vector from the center of the chart, with 0 length being a perfect match to the reference impedance and 1 being total reflection at the circumference of the chart. The underlying grids of the Smith chart are circles of a given resistance and arcs of impedance.

The reflection coefficient radius of the standard Smith chart is unity. Compressed Smith charts with a radius greater than 1 and expanded charts with a radius less than 1 are available.

High impedances are located on the right portion of the chart, low impedances on the left portion, inductive reactance in the upper half, and capacitive reactance in the lower half. Real impedances are on a line from the left to right, and purely reactive impedances are on the circumference. The angle of the reflection coefficient is measured with respect to the real axis, with zero degrees to the right of the center, 90^o straight up, and -90^o straight down.

The impedance of a load as viewed through an increasing length of lossless transmission line, or through a fixed length with increasing frequency, rotates in a clockwise direction with constant radius when the line impedance equals the reference impedance. If the line and reference impedances are not equal, the center of rotation is not about the center of the chart. One complete rotation occurs when the electrical length of the line increases by 180^o. Transmission line loss causes the reflection coefficient to spiral inward.

The length of a vector from the center to a given point on the Smith chart is the magnitude of the reflection coefficient. The angle of that vector with respect to the real axis to the right is the phase angle of the reflection coefficient. Several common definitions are used to represent the length of this vector. They are referred to as radially scaled parameters because they relate to a radial distance from the center towards the outside circle of the chart.

Chapter 5 HARBEC (DC & Harmonic Balance)

DC Analysis Overview

DC simulation analyzes the static operating points (DC voltages and currents) at each nonlinear node and port in the circuit. When designing circuits using non-linear models, you should always check the DC operating point before doing linear or harmonic balance simulations. DC analysis is very fast and will make sure that you have entered a workable design.

> **Note:** DC Simulation is not generally the same as the DC (zero frequency) level from a harmonic balance simulation. In DC simulation, all AC sources are turned off.

Nonlinear device models have many parameters that can be entered in error. To make sure that the model is correct, it is a good idea to look at the DC characteristic curves of the device before entering a complete circuit. Workspace templates are available (Select New From Template from the File Menu, then BJT Test.wsp) that make it easy to create these curves.

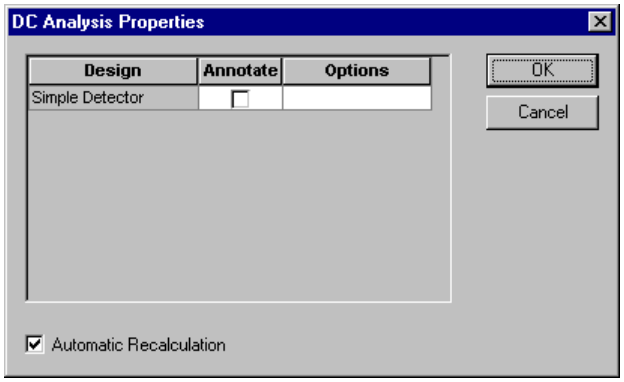
In addition to analysis, DC results can be optimized. For example, you can optimize bias resistor values to achieve a desired collector current and voltage for a bipolar transistor. See the walkthrough DC Analysis - Verifying Transistor Parameters for an example. It is located in one of the following sections.

To add a DC simulation:

1. Click the New Item button in the Workspace Window.
2. Select "Add DC Analysis" from the Analysis submenu.
3. Complete the DC Analysis Properties dialog box. For details, see the Reference manual.

DC Analysis Properties

To open: double-click or create a DC Simulation.



Property	Description	Additional Information
Design	Name of the design to run the DC Analysis on.	
Annotate	Check this to place DC voltages at all nodes connected to a nonlinear device; a source, a voltage test point, a current probe, or a port. DC current will be written to each current probe for display.	Only schematics support back-annotation of DC values. For netlists and EMPOWER simulations, place DC measurements on a tablular output to see the circuit operating point.
Options	Available for advanced applications and are normally not necessary. This field can be left	See chart below for optional parameters for this field
Automatic Recalculation	When selected will recalculate analysis each time a parameter is changed.	

The following table shows optional simulation parameters that can be set in the options field.

***note** - more than one simulation can be added by placing a semicolon between each parameter.

(Ex. gmin=1e-6; reltol=1e-4)

Option	Description	Example
Gmin	Changes the value of conductances added to each nonlinear node in the circuit. The simulator by default attaches a 1 pico-siemens conductance (1 teraohm resistor) between each node in the circuit and ground to assist with convergence.	gmin=1e-6

Option	Description	Example
GminSteps	Specifies the maximum number of Gmin steps used during DC analysis. These steps are used if there are convergence difficulties using the nominal value. Default value is 10.	gminsteps=15
RelTol	The relative accuracy to which the sum of node currents must sum to zero to achieve DC convergence. The simulator is converged if the ratio of (the vector sum of the currents into a given node currents) to (the sum of magnitudes of the current entering that node) is less than the specified relative tolerance. Default value is 1e-3.	reltol=1e-3
AbsTol	The absolute accuracy to which the sum of node currents must sum to zero to achieve DC convergence. The simulator is converged if the magnitude of the vector sum of the currents entering a given node at a given frequency is less than the specified absolute tolerance. Default value is 1e-12.	abstol=1e-10
Itl1	Specifies the maximum number of steps used in DC convergence. Default value is 10000.	itl1=15000
SrcSteps	Specifies the maximum number of amplitude steps used in DC analyses. When having difficulty finding DC convergence, the HARBEC will automatically adjust the amplitude of independent sources in the circuit. Default value is 10.	srcsteps=15

Harmonic Balance Overview

The HARBEC harmonic balance simulator simulates the steady-state performance of nonlinear circuits. Circuits can be stimulated with a variety of periodic signals (voltage, current, and power) such as single CW tones, pulsed waves, or dual tones. Complex waveforms can be constructed by combining various periodic signals; HARBEC makes this through the custom voltage and current sources. The two assumptions that harmonic balance uses are 1) the signals in the circuit can be accurately modeled using a finite number of spectral tones and 2) the circuit has a steady-state solution.

HARBEC works by solving Kirchoff's current law in the frequency domain. It applies the stimulus sources to the designed network. It then searches for a set of spectral voltages that will result in currents that sum to zero at each node and each frequency in the circuit. It adjusts the voltage levels (a spectrum of voltages at each node) through a variety of methods until the sum of the currents is less than a user-specified level (see "Absolute

Error" and "Relative Error" on the Harmonic Balance dialog box in the Reference Manual). This process of searching is known as "convergence." The length of time it takes to take a search step is roughly equal to the cube of the product of the number of frequencies and the number of nonlinear nodes. Thus, if you double the number of frequencies in the circuit, you can expect the solution to take roughly 8 times longer. However, this is only a rough estimate. The convergence process is complex and difficult to predict.

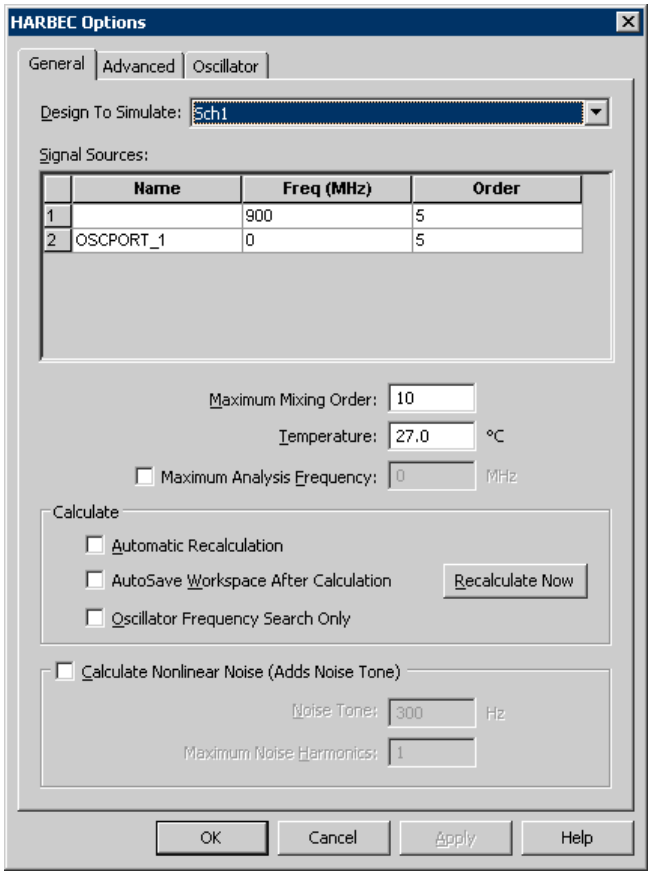
At a fundamental level, harmonic balance solves a simultaneous set of nonlinear differential equations. No mathematical approach is guaranteed to find a solution to the problem. Years of work have gone into HARBEC to develop the most robust strategies available.

To add an harmonic balance simulation:

1. In the Workspace Window, click the New Item button and select "Add Harmonic Balance Analysis" from the Analysis submenu.
2. Complete the HARBEC Options dialog box. For details, see the Reference manual.

HARBEC Options

To edit a Harmonic Balance properties, double-click the Harmonic Balance Analysis or click the analysis and click the Properties button on the Workspace Window.



General Tab

Design To Simulate - Defines the schematic or EMPOWER electromagnetic simulation that will be analyzed. If an EMPOWER simulation is selected, electromagnetic results will be co-simulated with the circuit elements associated with the layout.

Note: If an EM simulation is selected, it is very important that the "Use Ports from Schematic" option be properly checked on the EMPOWER Properties dialog.

Frequency Table and Order Control

Name - The schematic designator of the source. GENESYS searches the specified design for all sources and places them in the table.

Freq - The frequency specified on the source. GENESYS fills in this value by reading the frequency from the schematic.

Order - The number of harmonics to be analyzed. The larger the number of harmonics, the more accurately waveforms will be represented. However, the length

of time to find a solution increases as roughly the cube of the number of frequencies. Order must always be set large enough to model the majority of the energy in each branch current. Typical numbers for mildly nonlinear circuits are 4-5. For circuits deep in compression (square waves present), the order may need to be 8-16 to achieve the desired accuracy.

Maximum Mixing Order - Specifies the maximum combined order of signals to be simulated. In the example shown, all 4th order products will be calculates. For example, the 1900*2 -1905*1-1800*1 (95MHz)-the mixer third order intermodulation term-is a 4th order term (2+1+1) and will be calculated. This term only affects the mixing terms and will not override the order of individual sources specified in the frequency table.

Temperature - The temperature, in degrees Celsius, at which to perform nonlinear analysis.

Maximum Analysis Frequency - Frequency above which no nonlinear analysis is performed. If not checked, all frequency points in the analysis (input frequencies, their specified number of harmonics, and intermods) will be used.

Calculation

Automatic Recalculation - Checking this box will cause the harmonic balance simulation to be run any time there is a change in the design. If the box is not checked, the simulation must be run manually either by right clicking on the simulation icon and selecting "Recalculate Now" or by clicking the recalculation (calculator) button on the main tool bar.

Auto-save Workspace After Calculation - Checking this box will cause GENESYS to save the current workspace after the simulation is complete. This is particularly useful with long simulations or simulations that run overnight. If this box is checked when optimizing, the file will be saved after each optimization step.

Recalculate Now - Dismisses the dialog box and starts the simulator if required. If the circuit has already been simulated and has not been changed, the simulator will not calculate again.

Oscillator Frequency Search Only - Just perform analysis of oscillation frequency, not full HarBEC simulation.

Noise Parameters

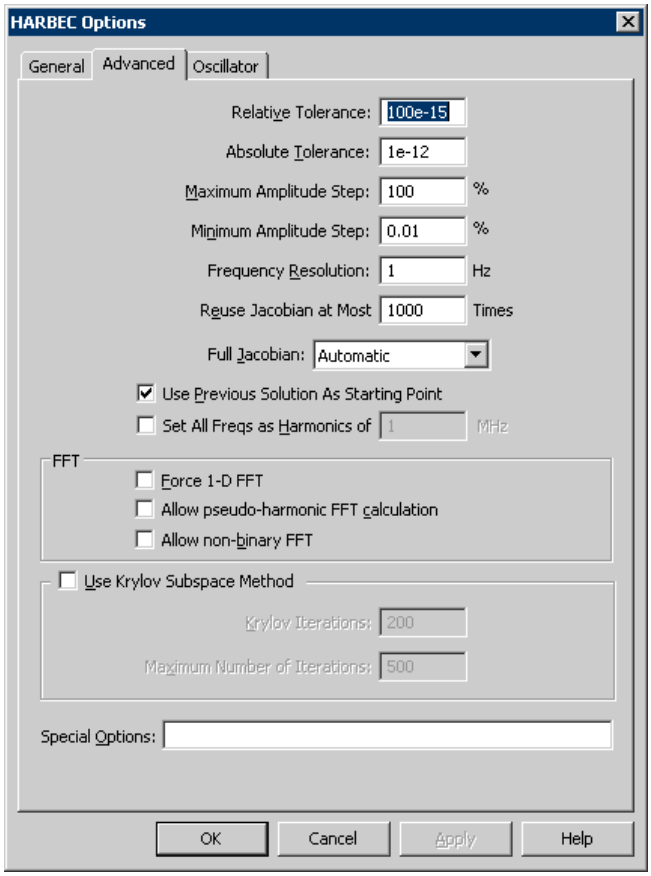
Calculate Nonlinear noise (Adds noise tone) - This options allows the user to add a noise tone with its harmonics to the simulation.

Noise Tone - Frequency in Hz of a noise tone that will be added to the harmonic balance simulation.

Maximum Noise Harmonics - Maximum number of noise tones that will be used in the harmonic balance simulation.

OK - Dismisses the dialog box. If automatic recalculation is on and a simulation is needed, the simulator will run after the box is dismissed

Cancel - Dismiss the dialog box, canceling any changes made.



Advanced Tab

Convergence Parameters

Refer to "Optimizing Simulation Performance" in the User Manual for details on the convergence process and the use of the parameters described below.

Relative Tolerance - The relative accuracy to which the sum of node currents must sum to zero to achieve harmonic balance convergence. The simulator is converged if, for all frequencies and all nodes, the ratio of (the vector sum of the currents into a given node currents) to (the sum of magnitudes of the current entering that node) is less than the specified relative tolerance.

Absolute Tolerance - The absolute accuracy to which the sum of node currents must sum to zero to achieve harmonic balance convergence. The simulator is converged if the magnitude of the vector sum of the currents entering all node at all frequencies is less than the specified absolute tolerance.

Maximum Amplitude Step - The highest amount that the simulator will increase the amplitude of the independent sources during the search for a solution. Normally set to 100%, it can be set smaller to improve the speed of convergence for some circuits.

Minimum Amplitude Step - The smallest amount the simulator will step increase the amplitude of the independent sources before the simulator tries another approach or terminates.

Frequency Resolution - The minimum difference in frequencies before the simulator will merge frequency terms. If the difference between two calculated frequencies (usually mixed frequency terms) is less than the frequency resolution, they will be considered a single frequency term for simulation.

Maximum Number of Jacobian Reuse - The largest number of times that a Jacobian matrix will be used before another Jacobian is calculated. Notice that, since HARBEC uses numeric techniques to calculate the Jacobian, it can be reused many more times than with other harmonic balance implementations.

Full Jacobian - Controls whether a full Jacobian or Fast Newton search step is taken during convergence. Select automatic, never, or always.

Use Previous Solution As Starting Point - Usually checked, this option will start the convergence process using the previous set of node voltages. If the parameters changed (or swept) are relatively small, starting with the previous solution can dramatically speed convergence. If the parameters changed are large is sometimes better to start from scratch. Certain circuits will always converge faster from scratch than previous solutions.

FFT

Force 1-D FFT - The simulator will normally convert frequency spectrums to time waveforms (and back) using multidimensional FFTs. If the frequencies are evenly spaced (have a large common factor), it may be faster to use a one-dimensional FFT. On some occasions, convergence can also be affected.

Allow psedo-harmonic FFT calculation - Artificially changes quasi-periodical signal to periodical. This allows a increase of calculation speed in multitone analysis. An example of this would be a noise analysis with many harmonics.

Allow non-binary FFT - Allows the use of FFTs that have powers other that 2^n where n=1, 2, 3, ... This allows a decrease the number of FFT points for multitone analysis and will results in calculations speed ups and decreases the needed memory for FFT arrays.

Krylov Subspace Method

Use Krylov Subspace Method - Select this box to use the Krylov technique. For large harmonic balance problems, this technique can dramatically reduce the amount of memory and time required to converge.

Krylov Iterations - The largest number of steps the Krylov simulator will attempt before aborting.

Maximum Number of Iterations - The largest number of convergence steps the simulator will take before adjusting the signal levels in the circuit.

Special Options

Enter any of the following parameters for advanced simulator control. Multiple parameters can be added on the line, separated by spaces.

Gmin - Changes the value of conductances added to each nonlinear node in the circuit. The simulator by default attaches a 1 pico-siemens conductance (1 teraohm resistor) to each node in the circuit to assist with convergence. For example, to change the value to 1 micro-siemens, enter gmin=1e-6.

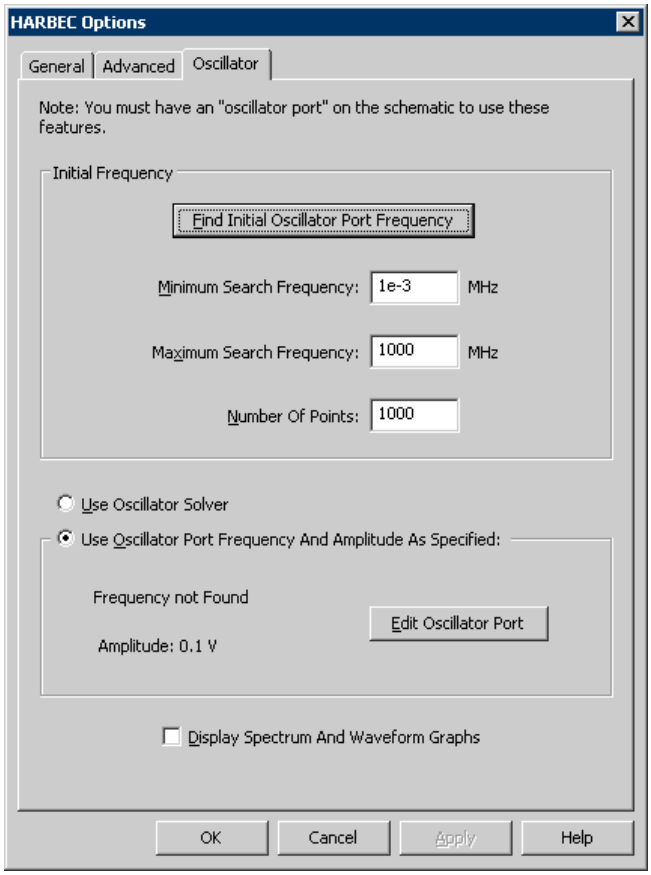
HB_Oversampl - Sets a factor for additional time points to be calculated during nonlinear device simulation, which can improve convergence but will take additional time. The factor should be set greater than 1. Typical values are 2-4.

HB_NonBinaryFFT - Allows the use of an FFT which is not a power of 2. For multitone problems, this can greatly reduce the size of the FFT required. For example, a 5-tone circuit with 4 harmonics per frequency normally requires 1,024,768 points, but only requires 100,000 points if this option is checked. This speedup often makes convergence take longer for smaller circuits, so it is not set by default.

HB_dfRelRec - The amount of improvement in the error function needed before a new Jacobian is calculated. Default value is 0.001

HB_dxRel - The relative step size used in calculating numerical derivatives. Default value is 0.001.

HB_dxAbs - The absolute step sized used in calculating numerical derivatives when the nominal parameter value is zero. Default value is 1e-10.



Oscillator Tab

Initial Frequency

Find Initial Oscillator Port Frequency - Calculates and fills in the frequency of oscillation for an oscillator port in the schematic using a linearization (frequency-dependent Y-matrix) of the nonlinear response of the circuit.

Minimum Search Frequency - The smallest frequency to search for the frequency of oscillation.

Maximum Search Frequency - The largest frequency to search for the frequency of oscillation.

Number of Points - the number of frequencies in the above range, linearly spaced, to search for the frequency of oscillation.

Harmonic Balance Calculation Options

Use Oscillator Solver - Perform nonlinear calculation of oscillation frequency, then use that frequency for HarBEC simulation.

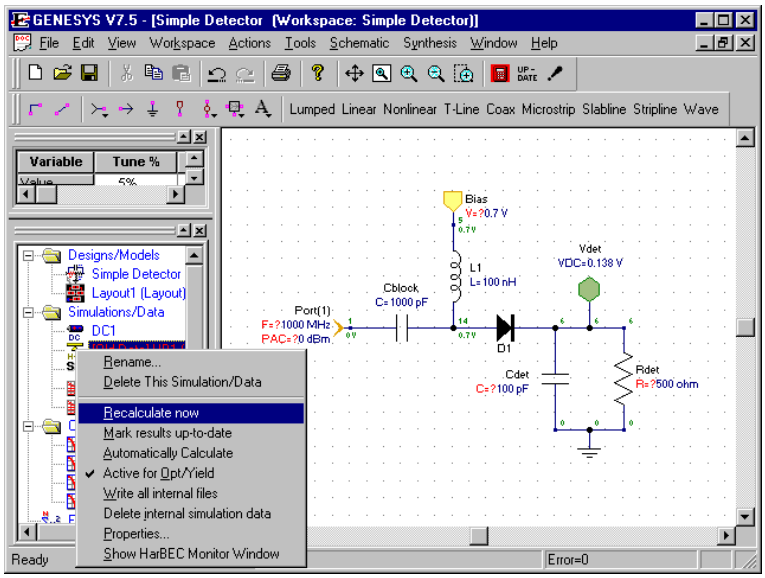
Use Oscillator Port Frequency and Amplitude as Specified - Use frequency and amplitude calculated from above "Initial Frequency" linearized calculation for the HarBEC simulation.

Edit Oscillator Port - Manually edit the frequency and amplitude of the oscillator port.

Display Spectrum and Waveform Graphs - Have the HarBEC simulation produce a frequency domain and time domain representation of the output signal.

HARBEC Popup Menu

By right-clicking on the HARBEC simulation icon on the workspace window, the following menu appears.



Rename - Allows the name of the icon to be changed

Delete This Simulation/Data - Removes the icon and all of its associated data from the system.

Recalculate Now - Starts a simulation if required. If the simulation is up-to-date (no changes have been made to the design since the last simulation), this command will be "gray" and the simulation will not be re-run. To force a new simulation, either make some change in the design or select "Delete internal simulation data".

Mark results up-to-date - Changes the status of a simulation to current. Use this feature when a change has been made to the design that does not affect the simulation results (such as changing a value and then changing it back).

Automatically Calculate - Toggles on or off the state that starts a simulation any time a change is made to the design.

Active for Opt/Yield/Recalc - Toggles on or off the simulation status. If not Active, the simulation will not be run when during optimization, yield analysis, or when the recalculation button is clicked.

Write all internal data - Creates a set of external ASCII files containing the simulation netlist, the simulator log messages, raw simulation results, and simulation errors.

Delete internal simulation data - Discards all existing calculated results. Selecting this menu will cause the simulator to start from a new state on its next run.

Properties - Opens the HARBEC Options dialog box

Show HarBEC monitor window - Opens a window that contains detailed information about the HARBEC simulation run. Only available for harmonic balance simulations.

Entering Nonlinear Models

GENESYS supports four different way to enter nonlinear models:

- Direct Schematic Entry
- Single Part Model
- Nonlinear Model Library
- SPICE Link

The simple way is to enter a nonlinear model is through direct schematic entry. You place a nonlinear device, such as an NPN transistor, from the schematic tool bar. Then double-click the device and type in the device parameters. The advantage of this technique is that it is simple. The disadvantage is that it is not as easy to reuse the device in another design.

Another way to enter a nonlinear model is to create a single part model. This is similar to using a model statement in other simulators. See the Designs: Single Part Model section in this User's Guide for details.

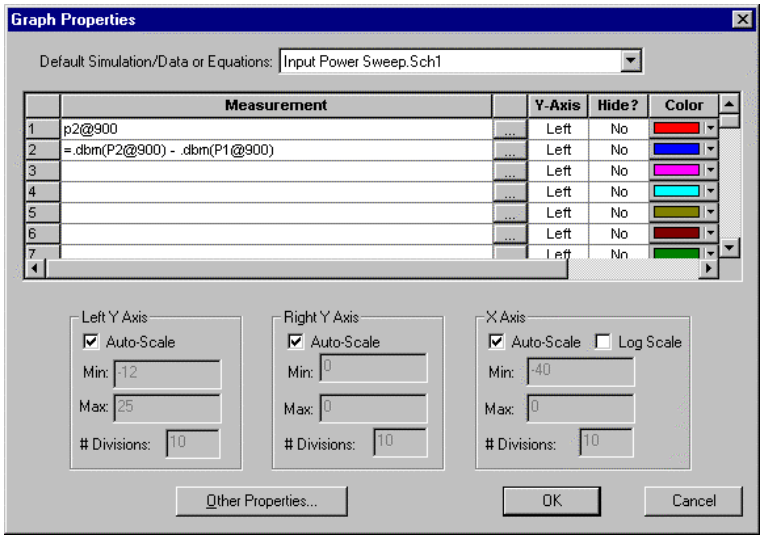
A third way to enter nonlinear models is to choose one from the supplied library of parts. To do this, just enter the base nonlinear model that you would like (for example, a PNP), then change the model to the desired part using the "Model..." button on the element parameter dialog.

The final way to enter a nonlinear part is to link the model to a SPICE netlist. GENESYS can read SPICE 3 compatible netlists, extracting models and subcircuits. Most vendors supply nonlinear models, including package parasitics, in the form of SPICE netlists. One advantage of SPICE links is that complex models can be included very easily in the simulation. The chance of error in entering numbers is reduced. The disadvantage of the link is that parameters are difficult to view and cannot be tuned or optimized. See the Designs: Link to Spice File section in this User's Guide for details.

Typical Harmonic Balance Measurements

Compression

To calculate compression of a circuit (a decrease in circuit transmission gain), use a parameter sweep to increase the power from a low level through compression. Assuming that the power input is on port 1 and the output is on port 2, the figure below shows how to plot the output power and the gain. Note that the default simulation is set to the power sweep. The first trace is P2@900, meaning the power at port 2 at 900 MHz. The second trace is the gain. Note that this is an inline equation. It starts with an equals sign, and the data is referred to by operator (the dBm operator is required in the equation, it is not needed as a direct plot as in trace 1).



Solving Convergence Issues

The simulator searches for a solution until the user-specified accuracy is reached, or until a specified number of searching steps. Sometimes you might run into convergence issues. Below are a few steps that you can use to improve convergence results. Each of the parameters below is changed on the Harmonic Balance (HARBEC) Options dialog box:

1. Increase the number of frequencies (the order) used in analysis. If not enough frequencies are used, the data is being undersampled and cannot accurately represent the solution. For example, modeling a square wave with three harmonics will ignore a lot of energy in the circuit, often leading to convergence issues. Increasing the number of frequencies analyzed will more accurately model the signals (at the expense of more time).

2. Try "Always" and "Never" options for calculating the Jacobian. If a Jacobian is calculated, the simulator will search in a different direction from the Fast Newton method. Sometimes the Jacobian will be a better direction, sometimes it will be worse. Try both approaches.

3. If the convergence issue occurs during a parameter sweep, sweep more points so that that each simulation is closer to the previous one, often requiring less total time. Or, if this is not practical or desired, turn off "Use Previous Solution As Starting Point." This will cause the simulator to start fresh with each new parameter value.
4. Increase the value of Absolute Tolerance and Relative Tolerance. This should speed up the solution but will be less accurate, particularly for low signal levels.

Optimizing Simulation Performance

A variety of methods and parameters are available to control the approach that HARBEC uses to find convergence. The speed of performance can be improved by adapting these parameters to the specific circuit being analyzed. To understand how these parameters work, it is useful to understand a little about how the simulator searches.

To find a solution, the simulator uses a Newton-Raphson search to find the solution. It starts with an initial guess and calculates an error function. The derivative of the error function is used to extrapolate the next point. In harmonic balance, partial derivatives exist for every node and every frequency. The full matrix of partial derivatives is known as a Jacobian.

Jacobian Calculation

The full Jacobian is usually the most accurate way to determine the next point. However, the matrix can be very large, requiring a lot of time to calculate and invert. To make the simulator faster, HARBEC generally tries Fast Newton steps first. A Fast Newton step calculates only a portion of the Jacobian and uses it to calculate the next point. For many circuits, the entire solution can be found quickly using only Fast Newton steps.

The default setting for HARBEC is to automatically switch between using Fast Newton and full Jacobian steps. Artificial intelligence techniques are used to determine which technique to use, and when. Usually, the automatic switching will find the solution quickly. However, for certain circuits, it will be better to always use the Jacobian or never use the Jacobian. On the HARBEC Options dialog box, you can specify either "Automatic," "Always," or "Never" use of the Full Jacobian. Experimenting with different values may improve convergence speed.

Order vs. Accuracy and Time

The easiest way to affect simulation performance is to change the order of the frequencies used in simulation. Harmonic balance models signals in the circuit by using a finite number of harmonics of the fundamental signals and a finite number of mixing terms. The larger the number of harmonics and mixing terms, the better the approximation of the actual signals. However, the larger the number of frequencies the longer the simulator takes to work. The length of time to take a search step is roughly proportional to the cube of the number of frequencies. So, doubling the number of frequencies will take about 8 times longer to simulate.

However, if not enough frequencies are present to adequately model the signals, then the results will not be accurate. Moreover, the simulator may have difficultly converging if not enough of the energy in the circuit is modeled.

The best practice in selecting order is to start with a reasonable number of harmonics of each signal (typically 5 is a good point), then increase the number until the results stop changing. "Order" and "Maximum Mixing Order" on the HARBEC Options dialog box control the number of terms. In this way, you can make tradeoffs of speed versus accuracy.

Amplitude Stepping

To start the search for convergence, HARBEC analyzes the circuit at DC, this is, with all independent AC signal turned off. Using DC as a first guess, it turns on the signals to "Maximum Amplitude Step" percentage of full signal. If convergence is reached at this step, it takes another equal step. If convergence is not reached, it decreases the step size and tries at the lower signal level. Some circuits will converge in a single 100% step. Others will require a smaller step to find the solution. If a smaller step is required, it will be faster to start with that step. If the step size is too small, the simulator may waste time calculating intermediate steps to find the final solution. Convergence speed can be improved by setting "Maximum Amplitude Step" to the ideal step.

Krylov Subspace Iterations

When the Jacobian matrix gets very large, it can become very slow to calculate and use. Krylov subspace iterations can dramatically reduce the size of the matrix and thus speed up calculations of very large circuits. In general, however, Krylov will have more convergence issues than full Jacobian steps. Also, for smaller circuits, Krylov may be slower than full Jacobian steps. For very large problems, try selecting Krylov to reduce memory requirements and speed convergence.

Nonlinear Noise Analysis

Nonlinear noise analysis (NNA) allows a nonlinear analysis to take into account internal circuit noise. It based on an approximation of continuous noise spectra by discrete spectra with spectral components having random equal distributed phases in the [-pi, pi] band.

Due to the stochastic nature of the noise analysis the algorithm does averaging of noise spectral amplitudes on the set of the sequential HARBEC analyses (each analysis actually generates a sequence differing by initial phases of the noise spectral components). Typically each analysis after the first is calculated very quickly, because it uses a very good initial guess as a result of the previous analysis and due to the fact that noise spectral components amplitudes are significantly less then the deterministic spectral components.

The current realization of the NNA suggests that the spectral densities of all sources of circuit noises are calculated at DC operating point and they will not be changed at time of the HB-analysis.

To create a Nonlinear noise analysis just set the flag "**Calculate Nonlinear Noise (Add Noise Tone)**" in the HARBEC options dialog window:

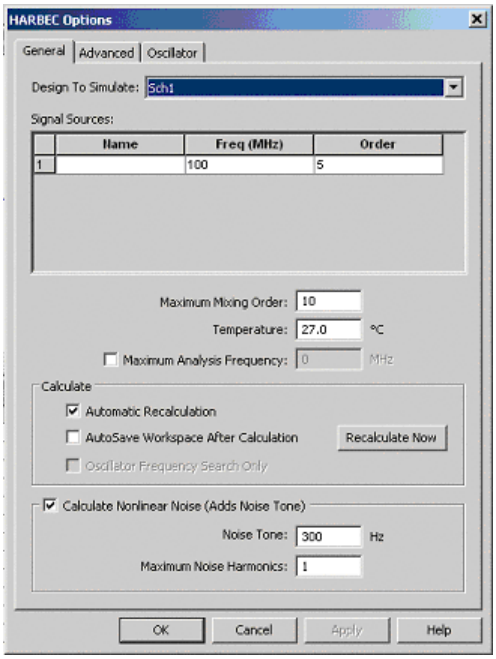


Fig.1 Harbec options dialog window

In the Nonlinear Noise Analysis the set of analysis frequencies *FB* (set of balanced frequencies) are defined as:

$$FB = \{fb = \sum_{i=1}^{n} m_i F_i \mid fb \geq 0;\ |m_i| \leq MaxOrder_i \mid i \in \overline{1,n};\ \sum_{i=1}^{n} |m_i| \leq MaxMixingOrder\}$$

This is extended by noise generated frequencies up to frequency set *FBn*:

$$FBn = \left\{Fbn = fb + i \cdot Fnoise \mid i \in \overline{-Nn, Nn};\quad Fbn \geq 0;\quad fb \in FB\right\}$$

where ***Nn*** – number of noise harmonics, taken into account in the analysis.

Which means that the noise analyses create the noise harmonics spectrums nearby and at the frequency of every spectral component from the ***FB*** set. After that each noise circuit element will add its noise current sources at all frequencies in the set ***FBn***.

For most cases it is enough to use the only 1st harmonic of the noise tone. Using more then 1 noise harmonic allows calculating the noise envelope of the discrete spectral components (noise side bands) with nonlinear interaction between noise components of the sidebands. In most cases the interaction is negligible because of the low level of circuit noise. The sidebands may be calculated using a sweep of 1 noise tone (see the option "**hb_sweepnoise**"). This significantly decreases the number of balanced frequencies and increases the speed and improves convergence.

"hb_noiserecalc" - the number of noise analyses recalculations for each set of base frequencies (used for averaging of the large signal noise analysis);

"**hb_onesidenoise**" - flag to calculate nonlinear noise for only single-sideband (SSB) of each deterministic spectral component;

"**hb_sweepnoise**" if set, will use a sweep of only 1 noise harmonic in the nonlinear noise analysis. This uses the results of a nonlinear noise analysis as starting point for the noise analysis with different value of the noise frequency. It decreases the order of the system of nonlinear equations and speeds up multi-noise-tones harmonic balance analysis.

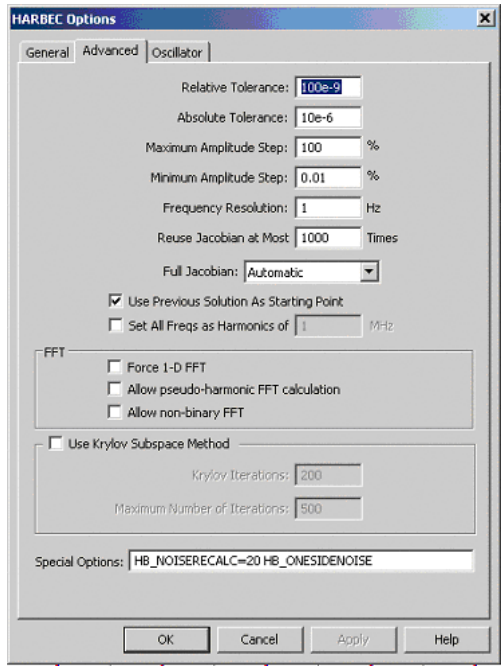


Fig.2 Harbec options dialog window (Special Options)

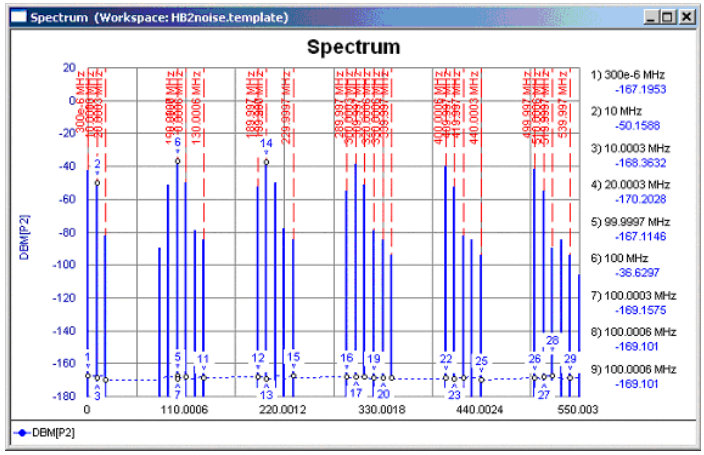


Fig. 3 The solution spectrum with noise

This graph includes noise spectral components (NSC) of each deterministic spectral components (DSC).

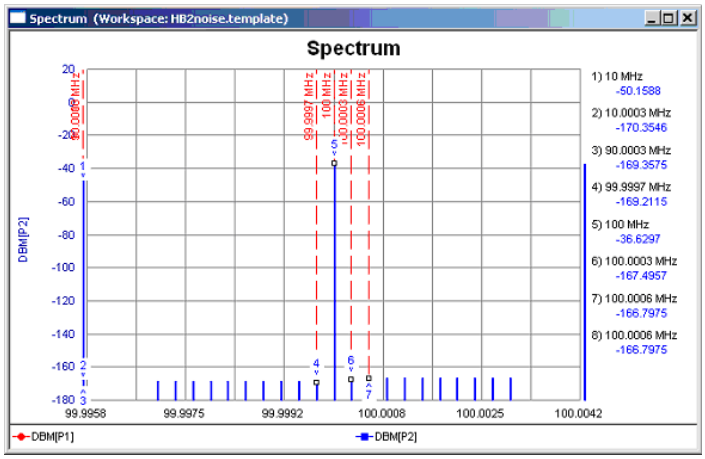


Fig. 4 Spectrum of the double-sideband (DSB) noise with *Nn*=10

The option **"hb_onesidenoise"**, defines that the NSC will be added only at the upper side of each DSC.

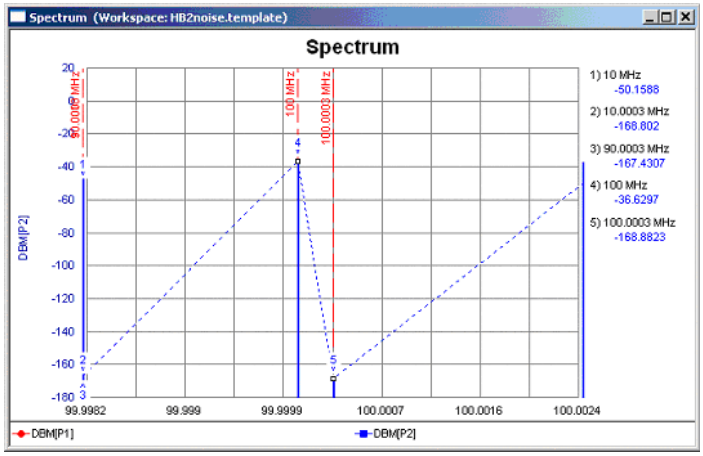


Fig. 5 Spectrum in vicinity of a DSC (*Nn* = 1)

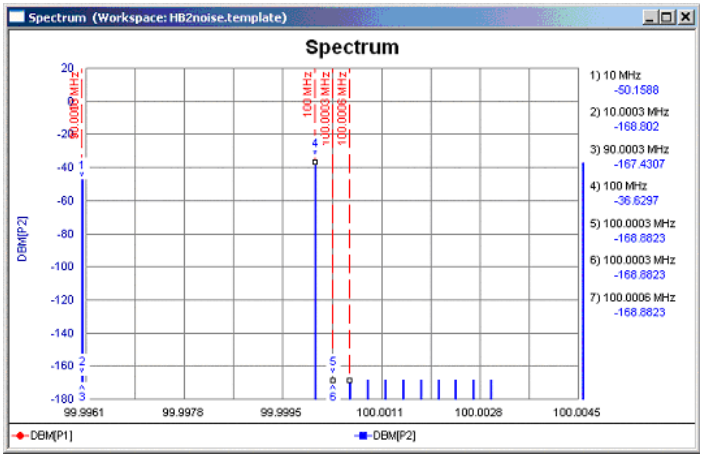


Fig. 5 Spectrum in vicinity of a DSC (SSB noise, Nn = 10)

Using more than 1 noise harmonic makes sense for noise analyses of nonlinear circuits working in very nonlinear mode (as at saturation or cut-off modes), where the nonlinear interconnection between different frequency parts of noise spectra become significant or for noise analysis in the vicinity of DSC, where flicker noise sources are more evident. In other cases it will be enough to use only the first harmonic of noise.

The noise characteristics of a circuit may be gotten from the spectrum by using an equation.

The Noise Figure (NF) is calculated as difference between signal/noise-ratio (SN) at signal source resistor (SNin) and calculated from output spectrum (SNout):

SNin= dbm(Pin)-dbm(k*Tin)

SNout=dbm[Pout(Fout)]-dbm[Pout(Fout+Fnoise)]

NF = SNin-SNout

Where

Pin – peak power of input tone;

Pout(F) – peak power of output spectral component having frequency **F**;

Fnoise – frequency of discretization of the noise spectrum (defined by "**Noise Tone**" option);

k - Boltzman's constant;

Tin – the absolute temperature of the signal source, Kelvin's degrees.

A simple example of the equations block for nonlinear noise characteristics calculation:

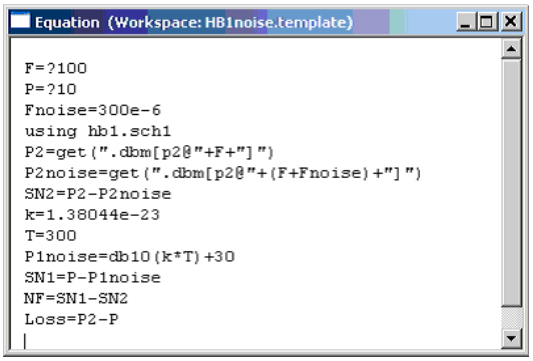
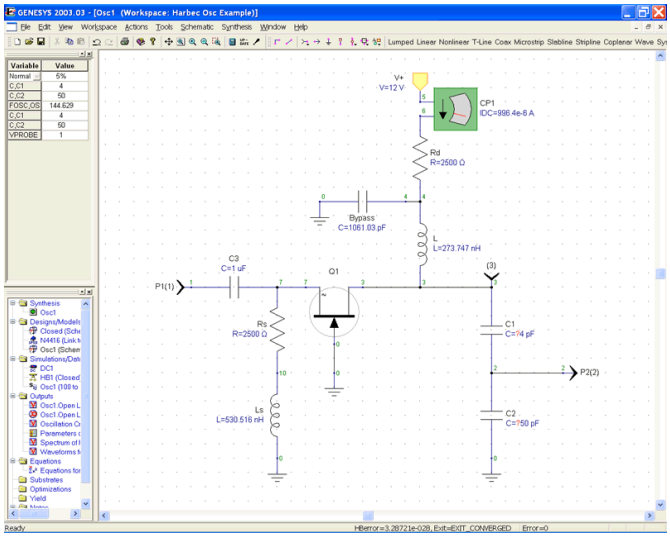


Fig 6. Example of the equations for nonlinear noise characteristics calculation

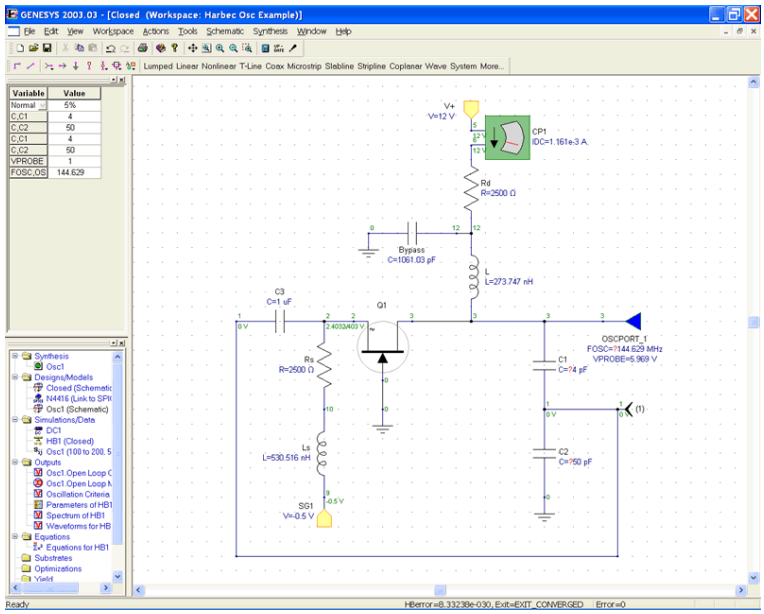
Oscillator Design Overview

Oscillator design begins with three basic elements. Amplification, a frequency determining circuit or device, and feedback to overcome network losses and provide power to the load.

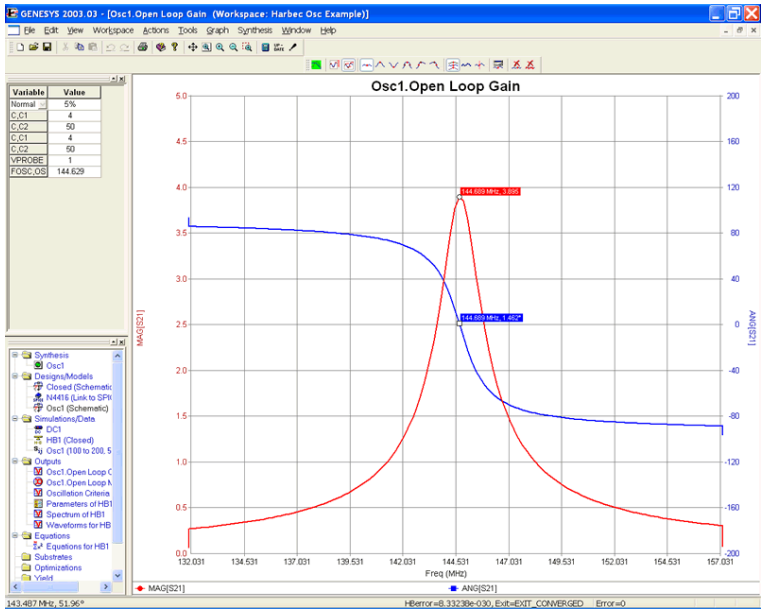
We start by selecting an amplifying device and topology that will provide gain at the desired frequency (band of frequencies for tunable oscillators). Next some form of a frequency selective network is added (e.g. crystal, L-C circuit, cavity, or dielectric resonator). And finally a feedback path that provides power flow from the amplifiers output back to the frequency selective network. There are generally many topologies available to provide positive feedback however, the the path should be chosen such that opening the path would result in termination of oscillation. A path that provides positive power flow from input to output (S21>1) in a broken feedback loop is an excellent starting point.



Having established a topology with a broken feedback loop we perform a two-port linear analysis on the oscillator to determine if oscillation is possible given the circuit and active device gain. In analyzing our circuit at the desired output frequency two conditions must be met that are required but not necessarily sufficient to ensure oscillation. At the desired output frequency the open circuit gain must be greater or equal to one(1) and the phase shift is a multiple of $2\pi n$ (n*360 degrees) where n is an integer including zero.

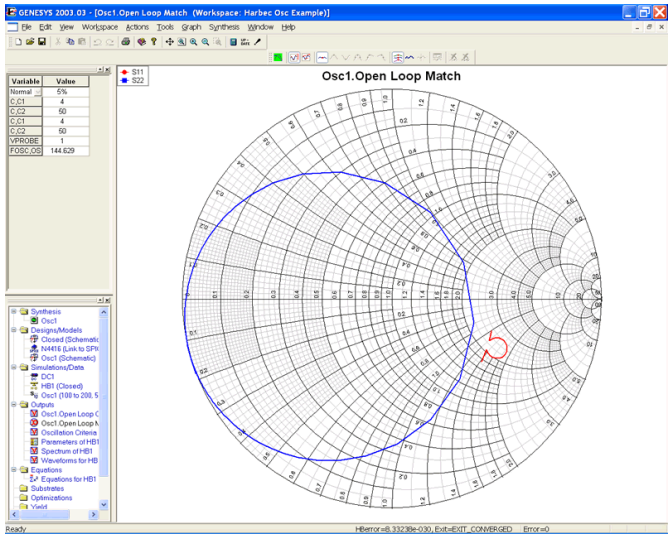


Using the tuning option in GENESYS allows for the exact selection of circuit elements to ensure a gain margin of >1 and phase shift of zero degrees at the desired frequency output. Considering the fact that our closed loop requires the connection of both ports it is helpful to view the return loss at the ports of our network to determine if additional matching structures would be necessary to aide in the maximum transfer of power. This data is already available as a result of our linear two port simulation. Ideally, S11=S22*. In the case that the port impedances are divergent, closing the loop might prevent sustained oscillation.



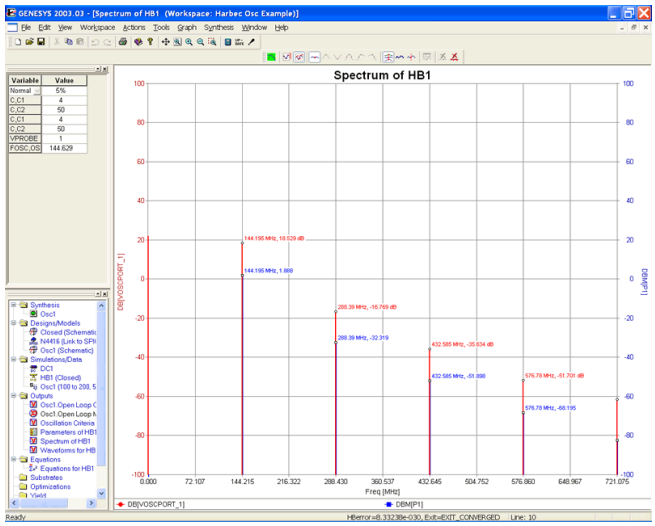
Having met the conditions for gain, phase, and match the next step is to verify oscillator performance and accurately determine the frequency of operation, power delivered to a load and the harmonic content. To accomplish this we use the OSCPORT element along with HARBEC analysis.

We begin by connecting the two ports together to close the loop. To initiate or provide start-up impetus to our circuit we will insert an OSCPORT component into the circuit. The placement is not critical, any node is useable, however best results are obtained if we do not place it on the output port or node. We access the OSCPORT element from the source selection icon on the GENESYS toolbar.



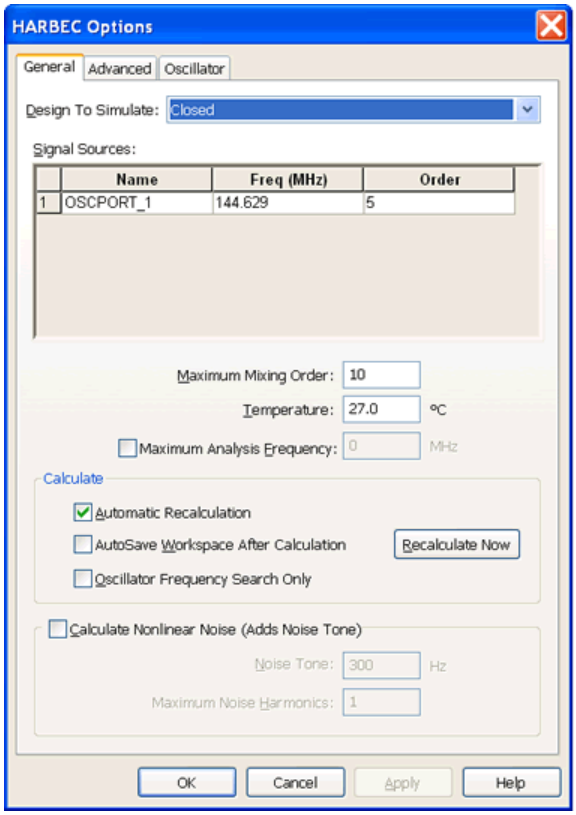
From the Simulations/Data folder in the GENESYS Workspace window add a DC Analysis simulation to determine the operating point of our device. This is necessary prior to any Harmonic Balance simulation.

From the Simulations/Data folder in the GENESYS Workspace Window add a Harmonic Balance analysis. Accept the default name or choose another. Note that the default analysis is for the current schematic. The oscillator frequency and note voltage will be filled in by the simulator after a successful run. The Harmonic balance dialog establishes a default value for the number of harmonics. Using the default values results in faster simulation. Having established a working oscillator this value may be changed to improve accuracy. Generally the default values will yield sufficient accuracy.



Selecting the Oscillator tab allows to enter the search range for analysis.

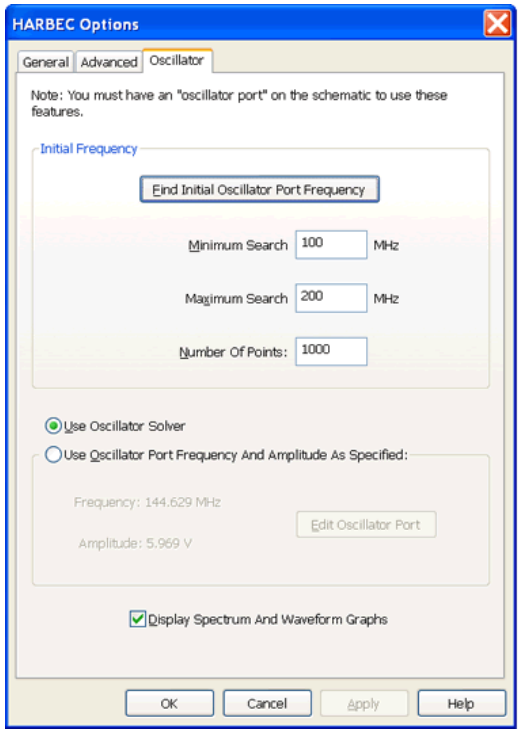
This gives the simulator a range of frequencies to search over to find the exact oscillation frequency. For resonator elements such as crystals, addition points may be required to find the exact frequency considering the higher Q.



Having set the range for search we are ready to perform an analysis.

Select the Update Icon from the GENESYS toolbar to perform an Harmonic Balance simulation. To view the results of our simulation add a rectangular graph from the Outputs file in the GENESYS Workspace Window. By double clicking on the graph or alternately selecting 'properties' from a right mouse click select the 'Measurement Wizard' to help place the spectrum data on the graph.

For Harmonic Balance simulations we are able to select from a range of node voltages, branch currents and ports.



Node or port waveforms are also available via the Measurement Wizard. This may aide in viewing the distortion and voltage levels at various nodes in the circuit.

Motorola LDMOS

The Motorola LDMOS library is now built into GENESYS. To add a Motorola LDMOS model to a schematic:

1. Click "More..." on the Schematic Toolbar.
2. Choose a "Builtin Motorola LDMOS" Category.
3. Select the desired part and Click OK.
4. Click OK to accept the default symbol.
5. Place the part on the schematic.

Note: the LDMOS models are nonlinear and require the HARBEC module. Contact Eagleware for information on purchasing this module if you do not own it.

You can now run a harbec, DC, or linearized simulation. To see an example of an LDMOS model in use, select File/Open Example, then load "Amplifiers/Large Signal S Parameters".

Chapter 6 Advanced Modeling Kit

Advanced Modeling Kit Overview

The GENESYS Advanced Modeling Kit (AMK) consists of three main parts:

- Approximately 12 additional nonlinear models for use in HARBEC. These models are ready to use and do not require knowledge of the Verilog-A language.
- A built-in Verilog-A compiler for creating your own nonlinear models.
- Verilog-A source code for all non-proprietary nonlinear models contained in GENESYS. These files allow you to make custom changes to any existing nonlinear model. For example, you can make a new model identical to a built-in transistor but with a change to the nonlinear capacitance equations.

Hardware description languages were developed as a means to provide varying levels of abstraction to designers. Integrated circuits are too complex for an engineer to create by specifying the individual transistors and wires. HDLs allow the performance to be described at a high level and simulation synthesis programs can the take the language and generate the gate level description. As behavior beyond the digital performance was added, a mixed-signal language was created to manage the interaction between digital and analog signals. A subset of this, Verilog-A was defined. Verilog-A describes analog behavior only; however, it has functionality to interface to some digital behavior.

Most other Verilog-A implementations are interpreted languages and are relatively slow. However, the GENESYS AMK includes a Verilog-A compiler that creates C++ code which is compiled, yielding simulation times similar to hand-coded models. Additionally, since the derivatives symbolically calculated by the AMK are often more accurate, convergence of circuits using Verilog-A models is generally better. This often results in a speed-up (not slow-down) when using Verilog-A models.

Using the Additional AMK Models

The GENESYS AMK includes additional nonlinear models not available with the standard HARBEC simulator. As of the time of this writing, the new models are:

- Angelov NFET/PFET
- BSIM4 NMOS/PMOS
- EKV NMOS/PMOS
- HiSIM NMOS/PMOS
- Philips JUNCAP
- MEXTRAM NPN/PNP
- Philips MOS9 NMOS/PMOS

- Philips MOS11 NMOS/PMOS
- Parker/Skellern NFET
- TFT NMOS/PMOS
- UCSD HBT NPN

To access these models:

1. Click the "More..." button on the right side of the schematic toolbar.
2. Choose the Category "Builtin Advanced Modeling Kit".
3. Choose the desired model and click OK.
4. Generally, you will want the default schematic symbol, so click OK on the "Choose Symbol to Place" dialog box.
5. Place the part on the schematic.

Creating New Verilog-A Models

The basic steps in creating a new Verilog-A Model are:

1. Open the Tools/Options Dialog, go to the Startup Tab, and click the Models button to open the Model Manager. You must place your Verilog-A source files in one of the directories listed here, or you must add a new directory.
2. Create a text file containing the Verilog-A source code in a text editor such as Windows Notepad and save it into the directory chosen in step 1. Be sure to use the extension ".va" on the file. See the Verilog-A Tutorial for information about creating a Verilog-A file.
3. Exit and restart GENESYS. At startup, GENESYS automatically compiles any out-of-date files by comparing the time and date of the Verilog-A (.va) file to the time and date of the compiled model library (.cml) file.

Note: GENESYS only compiles Verilog-A files on startup. You must restart GENESYS if you make a change to the Verilog-A source.

---

4. If there are errors shown in your Verilog-A code, fix them and restart GENESYS.
5. When the model successfully compiles, a "Compiled" subdirectory is created in the same directory as the Verilog-A source file. GENESYS creates a compiled model library (.cml) file and a model (.mod) file for each Verilog-A file.
6. To use your new model:
    a. Click the "More..." button on the right side of the schematic toolbar.

b. Choose the Category with the filename of the .mod file created by GENESYS.

c. Choose the desired model and click OK.

d. Choose a schematic symbol, and click OK on the "Choose Symbol to Place" dialog box. If you have used the Eagleware DEVICE_CLASS extension, this symbol will be chosen automatically.

e. Place the part on the schematic.

Customizing Built-In Nonlinear Models

GENESYS supplies Verilog-A source code for most of the built-in nonlinear models. This allows you to create models identical to the GENESYS built-in nonlinear models and then customize these as you see fit.

Note: You cannot change the built-in models. Instead, you must create a new model and must use this new model in your schematic.

The source code is in the Examples\VerilogA directory (normally installed to **C:\Program Files\GENESYS 2003.10\Examples\VerilogA**). To use these files, you should copy them to a new directory as described in step 1 of "Creating New Verilog-A Models" above.

The example modules have "_va" added to the end of the name to keep them from conflicting with the built-in models. All modules in GENESYS must have unique names, so it is recommended that you change the module name anytime you create a new Verilog-A file. If you only copy the file but don't rename the module, you will get errors due to duplicate GENESYS models or Verilog-A modules.

After you have copied the Verilog-A source file, you should follow the steps in "Creating New Verilog-A Models" above.

Note: As of the first release of GENESYS 2003.10, not all of the Verilog-A source was ready for distribution. If you are interested in customizing a nonlinear model which is not yet available, contact Eagleware to get the latest Verilog-A source libraries.

Verilog-A Tutorial

Verilog-A is a procedural language, with constructs similar to C and other languages. While the language does allow some knowledge of the simulator, most model descriptions should not need to know anything about the type of analysis being run.

Perhaps the simplest possible Verilog-A file is a resistor (the line numbers are not part of the verilog file):

```
1:  `include "disciplines.vams"
```

```
 2:
 3:  module resistor(p,n);
 4:    inout p,n;
 5:    electrical p,n;
 6:    parameter real r=50 from (0:inf] exclude 7;
 7:    analog
 8:       begin
 9:        V(p,n) <+ r*I(p,n);
10:      end
11: endmodule
```

You can use this resistor as a starting point for your own Verilog-A files, or you may start with a more complex file such as the built-in nonlinear models.

Line 1: `include "disciplines.vams"

This line includes the definitions for electrical nodes, among other things, and should be the first line of most Verilog-A files. Note the use of the ` symbol. *It is not a normal apostrophe (').* On most keyboards it is located on the upper left key, the same key as the tilde (~).

Line 3: module resistor(p,n);

Declares the start of a module named resistor with two external terminals, p and n. These terminals are used in order by GENESYS, so p becomes pin 1 and n becomes pin 2 in the symbol.

Line 4: inout p,n;

Declares that these ports are input/output ports.

Line 5: electrical p,n;

Declares that these nodes are electrical. If internal nodes are needed, they should be added to this line.

Line 6: parameter real r=50 from (0:inf] exclude 7;

Declares model parameter "r" with a default value of 50. This value can range from greater than zero (using opening parenthesis to indicate zero is not allowed) to infinity. Infinity is a legal value since square bracket was used. The value 7 is specifically excluded.

Line 7: analog

Header for the analog equations. Required in all files.

Line 8: begin

Starts the actual analog equations. Often, this is combined with "analog" on one line: "analog begin".

Line 9: V(p,n) <+ r*I(p,n);

Adds a voltage due to the resistor (V=IR). V(p,n) is the voltage from node p to node n. I(p,n) is a branch current flowing from node p to node n. Note: This branch current is automatically added by the compiler as another variable to be solved, and the matrix entries to support this Modified Nodal Analysis relationship are also added automatically.

Line 10: end

Ends the analog equations started at line 8.

Line 11: endmodule

Ends the resistor module started at line 3.

Other commonly used features not shown in this simple example include local variables/equations and if/then statements. See the Verilog-A examples or the Verilog-A reference section of this manual

Verilog-A Reference

Verilog-A Reference Overview

This manual does not give a complete technical reference to Verilog-A. Rather, the objective is to give a model developer enough details to implement complex models without being weighted down with syntax charts and excessive details. To purchase a complete reference to Verilog-A, contact Accellera at www.accellera.org.

Preprocessor

The preprocessor supports certain directives in order to simplify code development. These directives are very similar to their C counterparts.

Include

The `include directive is used to insert the entire contents of a source file during compilation. The `include can be used to simplify code by including global definitions or without repeating code within module boundaries. The compiler directive `include can be specified anywhere within the Verilog-A file. The filename is the name of the file (with either the full or relative path) to be included in the source file.

Only white space or comments can appear on the same line as the `include directive.

A file included in the source using `include can contain other `include compiler directives; however, infinite nesting is not permitted.

```
`include "filename"
```

Examples:

```
`include "/user/include/global_decl.vams"
`include "../myIncludes.txt"
`include "myFunctions.va"
```

Macros

String substitution can be performed with the `define directive both inside and outside module definitions. The macro is used in the source file by insert the ` character, followed by the macro name. The preprocessor then substitutes the text of the macro for the string `text_macro_name. All compiler directives are considered predefined macro names and so redefining a compiler directive as a macro name is not allowed.

A text macro can be also be defined with arguments to provide much more flexibility. However, the use of macros can complicate symbolic debugging, so the user should be careful in their use.

Examples:

```
`define EPSSI    (1.03594e-10)
`define KboQ    (`P_K / `P_Q)
`define strobe(flag, xName, X)    if (_debug >= flag)  $strobe("\n %s = %g", xName, 1.0 *(X))
```

The macros are then accessed in the code as:

```
factor1=sqrt(`EPSSI / `EPSOX * tox);
`strobe(1, "Vth", Vth);
```

ifdef, else, endif

These are conditional compiler directives for optionally including lines of Verilog-A source file. `ifdef checks for if variable name is defined. If it is defined, the lines following `ifdef are included up to the `endif directive. If the variable name is not defined, but an `else directive exists, this source is compiled. The ifdef, else, and endif directives can appear anywhere in the Verilog-A source file.

Examples:

```
`ifdef Thermal
    module bjt(c,b,e,dt);
`else
    module bjt(c,b,e);
`endif
```

Note: GENESYS does not support predefining a macro (as is often done from a command line build). You must define any necessary switches within the Verilog-A source. A useful method is to create a Verilog-A file that does nothing but `define macros and then `includes the real Verilog-A source. In the example shown above, that file could `define Thermal before including the bjt module.

Data Types and Parameters

Integer

An integer declaration declares one or more variables of type integer holding values ranging from -231 to 231-1. Arrays of integers can be declared using a range which defines the upper and lower indices of the array where the indices are constant expressions and shall evaluate to a positive or negative integer, or zero.

Example:

```
integer flag, MyCount, I[0:63];
```

Real

A real declaration declares one or more variables of type real using IEEE STD-754-1985, (the IEEE standard for double precision floating point numbers). Arrays of reals can be declared using a range which defines the upper and lower indices of the array where the indices are constant expressions and shall evaluate to a positive or negative integer, or zero.

Example:

```
real X[1:10], Tox, Xj, Cgs;
```

net_discipline

The net_discipline is used to declare analog nets and for declaring the domains of digital nets and regs. A net is characterized by the discipline that it follows. A net is declared as a type of discipline and so a discipline can be considered as a user-defined type for declaring a net.

A discipline is a set of one or more nature definitions forming the definition of an analog signal whereas a Nature defines the characteristics of the quantities for the simulator. A discipline is characterized by the domain and the attributes defined in the natures for potential and flow.

The discipline can bind:

- One nature with potential
- One nature with potential and a different nature with flow
- Nothing with either potential or flow (an empty discipline)

The disciplines are typically predefined in the disciplines.vams file, a portion of which is shown below.

```
// Electrical
// Current in amperes
nature Current
units = "A";
```

```
access = I;
idt_nature = Charge;
`ifdef CURRENT_ABSTOL
   abstol = `CURRENT_ABSTOL;
`else
   abstol = 1e-12;
`endif
endnature
// Charge in coulombs
nature Charge
units = "coul";
access = Q;
ddt_nature = Current;
`ifdef CHARGE_ABSTOL
   abstol = `CHARGE_ABSTOL;
`else
   abstol = 1e-14;
`endif
endnature
// Potential in volts
nature Voltage
nature Voltage
units = "V";
access = V;
idt_nature = Flux;
`ifdef VOLTAGE_ABSTOL
   abstol = `VOLTAGE_ABSTOL;
`else
   abstol = 1e-6;
`endif
endnature
```

genvar

Genvars are integer-valued variables which compose static expressions. They are used for instantiating structure behaviorally e.g., accessing analog signals within behavioral looping constructs.

genvar *list_of_genvar_identifiers*;

where *list_of_genvar_identifiers* is a comma separated list of genvar identifiers.

Example:

```
genvar I, j;
```

Parameters

Parameters provide the method to bring information from the circuit to the model.

Parameter assignments are a comma-separated list of assignments. The right hand side of the assignment is a constant expression (including previously defined parameters).

For parameter arrays, the initializer is a list of constant expressions containing only constant numbers and previously defined parameters within { and } bracket delimiters.

Parameters represent constants; their values can not be modified at runtime.

Parameters can be modified from the declaration assignment at compilation time. The purpose is to allow customization of module instances. A parameter, however, can be modified with the defparam statement or the module_instance statement. It is not legal to use hierarchical name referencing from within the analog block to access external analog variable or parameter values. An example is:

```
parameter real TestFlag = 0 from [0:inf) exclude (10:100) exclude (200:400];
```

The general format is:

parameter {real | integer} *list of assignments* ;

where the *list of assignments* is a comma separated list of
parameter_identifier = constant {*value-range*}

where *value-range* is of the form
from *value_range_specifier*
| exclude *value_range_specifier*
| exclude *constant_expression*

where the *value_range_specifier* is of the form
start_paren expression1 : *expression2 end_paren*

where *start_paren* is
[| (
and *end_paren* is
] |)
and *expression1* is
constant_expression | -inf
where expression2 is
constant_expression | inf
and where a *constant_param_arrayinit* is
{ *param_arrayinit_element_list* }
where *param_arrayinit_element_list* is made of
param_arrayinit_element {, *param_arrayinit_element* }

where *param_arrayinit_element* is a *constant_expression*.

The type (real | integer) is optional. If it is not given, it will be derived from the constant assignment value. A parenthesis indicates the range can go up to, but not include the value, whereas a square bracket indicates the range includes the endpoint.

The value range specification is quite useful for range checking. Some examples of this are:

```
parameter real Temp = 27 from [-273.15:inf);
parameter R = 50 from (0:inf];
```

and value ranges can have simple exclusions:

```
parameter R = 50 from (0:inf] exclude (10:20) exclude 100;
```

Analog Block

Expressions and statements

Conditional statement (if-else statement)

The conditional statement is used to determine whether a statement is executed or not. The syntax is

```
if ( expression ) true_statement_or_null ;
[ else false_statement_or_null ; ]
```

If the expression evaluates to True (non-zero), then the *true_statement* will be executed (or not, if false). If there is an else *false_statement* and the expression evaluates to False, the *false_statement* is executed instead.

Case statement

A case statement is useful for multiple actions to be selected based on an expression.

The format is:

```
{ case | casex | casez } ( expression )
   case_item
   { case_item }
endcase


where case_item is
expression { , expression } : statement_or_null
| default [ : ] statement_or_null
```

The *default-statement* is optional but if used, can only be used once. The *case-expression* and the *case_item* expression can be computed at runtime (neither expression is required to be a constant expression). The *case_item* expressions are evaluated and compared in the exact order in which they are given. If one of the *case_item* expressions matches the case-expression given in parentheses, then the statement associated with that *case_item* is executed. If all comparisons fail then the default item statement is executed (if given). Otherwise none of the *case_item* statements are executed.

Repeat and while looping statements

The repeat() statement executes a statement a fixed number of times. Evaluation of the expression determines how many times the statement is executed.

The while() looping executes a statement until an expression becomes False. If the expression is False when the loop is entered, the statement is not executed at all.

The syntax for the repeat() and while() statements is shown

repeat (expression) statement
while (expression) statement

For() statement

The for() statement controls execution of its associated statement(s) using an index variable. If the associated statement is an analog_statement, then the control mechanism must consist of genvar_assignments and genvar_expressions.operators (no use of procedural assignments and expressions).

for (*procedural_assignment* ; *expression*; *procedural_assignment*) *statement*

where for analog_for_statement the format is

for (*genvar_assignment*; *genvar_expression*; *genvar_assignment*) *analog_statement*

Signals

Accessing net and branch signals

Signals on nets and branches are be accessed only by the access functions of the associated discipline. The name of the net or the branch is specified as the argument to the access function.

Examples:

```
Vin = V(in);
CurrentThruBranch = I(myBranch);
```

Events

The analog behavior of a component can be controlled using events, which have the characteristics:

- Events have no time duration
- Events can be triggered and detected in different parts of the behavioral model
- Events do not block the execution of an analog block
- Events can be detected using the @ operator
- Events do not hold any data
- There can be both digital and analog events

There are two types of analog events: global events and monitored events. Null arguments are not allowed in analog events.

cross function

The cross() function is used for generating a monitored analog event. It is used to detect threshold crossings in analog signals when the expression crosses zero in the direction specified. cross() can control the timestep to accurately resolve the crossing. The format is

cross(*expr* [, *dir* [, *time_tol* [, *expr_tol*]]]);

where *expr* is required, and *dir*, *time_tol*, and *expr_tol* are optional arguments. The *dir* argument is an integer expression; the other arguments are real. If the tolerances are not defined, they are set by the simulator. If either or both tolerances are defined, then the direction of the crossing must also be defined. The direction can only evaluate to +1, -1, or 0. If it is set to 0 or is not specified, the event and timestep control will occur on both positive and negative signal crossings. If dir is +1 or -1, then the event and timestep control occur on rising edge (falling edge) transitions of the signal only. For other transitions of the signal, the cross() function will not generate an event. *expr_tol* and *time_tol* represent the maximum allowable error between the estimated crossing point and the actual crossing point.

Examples:

The following description of a sample-and-hold illustrates how the cross() function can be used.

```
module sample_and_hold (in, out, sample) ;
  output out;
  input in, sample;
  electrical in, out, sample;
  real state;
  analog begin
    @(cross(V(sample) -2.0, +1.0))
    state = V(in) ;
    V(out) <+ transition(state, 0, 10n) ;
  end
endmodule
```

The cross() function maintains its internal state. It has the same restrictions as other analog operators in that it can not be used inside an if(), case(), casex(), or casez() statement unless the conditional expression is a genvar expression. Also, cross() is not allowed in the repeat() and while() iteration statements but is allowed in the analog_for statements.

timer function

The timer() function is used to generate analog events. It is used to detect specific points in time. The general form is

timer (start_time [, period [, time_tol]]) ;

where start_time is a required argument but period and time_tol are optional. The timer() function schedules an event to occur at an absolute time (start_time). The analog simulator then inserts a time point within timetol of an event. At that time point, the event evaluates to True. If time_tol is not specified, the default time point is at (or just beyond) the time of the event. If the period is specified as greater than zero, the timer function schedules subsequent events at multiples of period.

Examples:

A pseudo-random bit stream generator is an example how the timer function can be used.

```
module bitStreamGen (out);
  output out;
  electrical out;
  parameter period = 1.0;
  integer x;
  analog begin
    @(timer(0, period))
    x = $random + 0.5;
    V(out) <+ transition( x, 0.0, period/100.0 );
  end
endmodule
```

Operators

Analog operators operate on an expression and return a value. Furthermore, they can operate on more than just the current value of their arguments as they maintain their internal state and so their output is a function of both the input and the internal state.

Because they maintain their internal state, analog operators are subject to several important restrictions. These are:

- Analog operators can not be used inside conditional (if and case) or looping (for) statements unless the conditional expression is a genvar expression (which can not change their value during the course of an analysis).
- Analog operators are not allowed in repeat and while looping statements.
- Analog operators can only be used inside an analog block; they can not be used inside an initial or always block, or inside a user-defined analog function.

Under most cases, you can not specify a null argument in the argument list of an analog operator.

Operator	Function
Time derivative	The ddt operator computes the time derivative of its argument. The form is ddt(*expr*)
Time integral	The idt operator computes the time-integral of its argument. The general form is idt(*expr*)
Linear time delay	absdelay() implements the absolute transport delay for continuous waveforms The general form is absdelay(*input*, *td* [, *maxdelay*])
Discrete waveform filters (transition, slew)	transition(*expr [, td [, rise_time [, fall_time [, time_tol*]]]]) The slew analog operator bounds the rate of change (slope) of the waveform. The general form is slew(*expr* [, *max_pos_slew_rate* [, *max_neg_slew_rate*]]). The last_crossing() function returns a real value representing the simulation time when a signal expression last crossed zero. The format is last_crossing(*expr*, *direction*)
Laplace transform filters	laplace_zp() implements the zero-pole form of the Laplace transform filter. The general form for each is laplace_zp(*expr*, *z*, *r* [,*e*]) laplace_zd() implements the zero-denominator form of the Laplace transform filter. The laplace_np() implements the numerator-pole form of the Laplace transform filter. laplace_nd() implements the numerator-denominator form of the Laplace transform filter.
Z-transform filters	The Z-transform filters implement linear discrete-time filters. Each filter uses a parameter T which specifies the filter's sampling period. The zeros argument may be represented as a null argument. The null argument is produced by two adjacent commas („) in the argument list. All Z-transform filters share three common arguments: T, t, and t0. T specifies the period of the filter, is mandatory, and must be positive. t specifies the transition time, is optional, and must be nonnegative. zi_zp() implements the zero-pole form of the Z-transform filter. The general form is zi_zp(*expr* , *z* , *r* , *T* [, *t* [, *t0*]]) zi_zd() implements the zero-denominator form of the Z-transform filter. zi_np() implements the numerator-pole form of the Z-transform filter. zi_nd() implements the numerator-denominator form of the Z-transform filter.

Contribution assignment statements

Sequential block

A sequential block is a grouping of two or more statements into one single statement. The format is:

begin [: *block_identifier* { *block_item_declaration* }]
{ *statement* }
end

where *block_item_declaration* is
parameter_declaration
integer_declaration | *real_declaration*

Indirect branch assignment

An indirect branch assignment is useful when it is difficult to solve an equation. It has this format:

V(n) : V(p) == 0;

Which can be read as "find V(n) such that V(p) is equal to zero." This example says that node n should be driven with a voltage source and the voltage should be such that the given equation is satisfied. V(p) is probed and not driven.

Indirect branch assignments are allowed only within the analog block.

Branch contribution statement

A branch contribution statement typically consists of a left-hand side and a right-hand side, separated by a branch contribution operator. The right-hand side can be any expression which evaluates to (or can be promoted to) a real value. The left-hand side specifies the source branch signal to assign the RHS. It consists of a signal access function applied to a branch. The form is:

V(*n1*, *n2*) <+ *expression*;

Branch contribution statements will implicitly define source branch relations. The branch is goes from the first net of the access function to the second net. If the second net is not specified in the call, the global reference node (ground) is used as the reference net.

Ports

Ports provide a way to connect modules to other modules and devices. A port has a direction: input, output, or inout, which must be declared. The ports are listed after the module declaration. The port type and port direction must then be declared in the body of the module.

Examples:

```
module resistor(p,n);
  inout p,n;
```

```
  electrical p,n;
…
module modName(outPort, inPort);
  output outPort;
  input inPort;
  electrical out, in;
…
```

Ports can support vectors (buses) as well.

Analog Functions

Analog functions provide a modular way for a user-defined function to accept parameters and return a value. The functions are defined as analog or digital and must be defined within modules.

The analog function is of the form:

analog function {real|integer} *function_name* ;
  input_declaration;
  statement_block;
endfunction

The *input_declaration* describes the input parameters to the function as well as any variables used in the statement block:

input *passed_parameters*;
real *parameter_list*;

The *statement_block* and analog function:

- Can use any statements available for conditional execution
- Can not use access functions
- Can not use contribution statements or event control statements
- Must have at least one input declared; the block item declaration declares the type of the inputs as well as local variables used
- Can not use named blocks
- Can only reference locally-defined variables or passed variable arguments

The analog function implicitly declares a variable of the same name as the function, *function_name*. This variable must be assigned in the statement block; its last assigned value is passed back.

Example:

```
analog function real B_of_T;
  input B, T, T_NOM, XTB;
  real B, T, T_NOM, XTB;
```

```
  begin
    B_of_T = B * pow(T / T_NOM, XTB);
  end
endfunction
```

The function is called by the line:

```
BF_T = B_of_T(BF, T, T_NOM, XTB);
```

System tasks and functions

System functions provided access to system level tasks as well as access to simulator information.

Environment parameter functions

These functions return simulator environment information.

Function	Return Value
$temperature	Circuit ambient temperature in Kelvin
$abstime	Absolute time in seconds
$realtime[(*scale*)]	$realtime can have an optional argument which scales the time. If no argument is given, $realtime's return value is scaled to the `time_unit of the module which invoked it. If an argument is given, $realtime shall divide the absolute time by the value of the argument (i.e., scale to the value specified in the argument). The argument for $realtime follows the semantics of the `time_unit, that is it shall consist of an integer followed by a scale factor. Valid integers are: 1, 10, and 100; valid scale factors are: s (seconds), ms (milliseconds), us (microseconds), ns (nanoseconds), ps (picoseconds), and fs (femtoseconds)
$vt[(*Temperature*)]	$vt can optionally have *Temperature* (in Kelvin units) as an input argument and returns the thermal voltage (kT/q) at the given temperature. $vt without the optional input temperature argument returns the thermal voltage using $temperature.

Input/output operations

These functions provide access to display and file operations.

Function	Return Value
$fopen(*file_name*)	$fopen opens the file specified as an argument and returns a 32-bit multichannel descriptor which is uniquely associated with the file. It returns 0 if the file could not be opened for writing.
$fclose(*file_id*)	$fclose closes the channels specified in the multichannel descriptor and does not allow any further output to the closed channels. $fopen reuses channels which have been closed.
$strobe(*args*)	$strobe provides the ability to display simulation data when the simulator has converged on a solution for all nodes using a printf() style format.
$monitor(*args*)	$monitor provides same capabilities as $strobe but outputs only when a parameter changes.

Eagleware Verilog-A Extensions

Eagleware has created several extensions to Verilog-A. These extensions are not required in any Verilog-A files, but they allow more complete information to be given to GENESYS about the model, making it easier to pass Verilog-A files between users. In GENESYS, a Verilog-A file gives a complete description of the model, and no other files are generally necessary to share between users.

Parameter Descriptions

First, parameter descriptions and units can be included in comments:

```
parameter real Vtr = 20.0; // Soft breakdown model parameter [V]
parameter real P3 = 0.0; // Polynomial coeff P3 for channel current [1/V^3]
parameter real Fnc = 0.0 from [0:inf); // Noise corner freq [Hz]
parameter real Cds = 0 from [0:inf]; // Zero-bias D-S junction capacitance [F]
```

Any comments on the same line as the parameter are assumed to be a description of the parameter. Additionally, units can be given inside square brackets. Currently supported units include: Hz, Ohm, mho, H, F, V, A, s, C, deg, m, W, and DB. If you use any of these units, then other related units (such as pF or dBm) can be specified when the parts are used. Unrecognized units (such as "1/V^3" above) are simply put into the description so that the user knows what units must be entered for the part.

Additionally, if the comment starts with "Unused" or "Alias", then the parameter is not shown to the user in GENESYS and the default value is used. If the comment starts with "Required", then GENESYS will give an error if the parameter is not specified. If the value `NOT_GIVEN is used as the default, then GENESYS will not show a default, but will just show "optional" for the default value.

Eagleware extension keywords

All Eagleware extension keywords are placed inside comments in the Verilog-A and are always placed between two pairs of percent signs, like:

// %%*KEYWORD*%%
// %%*KEYWORD=value*%%

Since these keywords are placed in comments, they will be ignored by other simulators. The keywords ***must*** be placed inside the module that they are to affect (after the module statement). Keywords only affect one module. If you have multiple modules in your VA file, you will need to duplicate any keywords which are to affect multiple modules.

DEVICE_CLASS keyword

The DEVICE_CLASS keyword tells GENESYS what type of device the module represent. This allows GENESYS to:

- Select a appropriate symbols
- Create multiple models for N or P class devices (such as NFET or PFET) from one Verilog-A source.
- Automatically reverse pins 1 and 2 for transistor class devices. This reversal is necessary since in SPICE (and in most Verilog-A source) the input is pin 2 and the output is pin 1. In GENESYS and most RF/Microwave simulation, the convention is for the input to be pin 1 and the output to be pin 2.

Some examples of device class statements are:

// %%DEVICE_CLASS=DIODE%%
// %%DEVICE_CLASS=FET(NFET,PFET)%%
// %%DEVICE_CLASS=MOS(NMOS:type=1,PMOS:type=-1)%%

The general format of the keyword is:

// %%DEVICE_CLASS=*type*(*option1*:*var1*=*value1*,*option2*:*var2*=*value2*...)%%

type is required and should be one of:

DIODE, BJT, BJT4, BJT5, FET, JFET, MOS, RESISTOR, CAPACITOR, CCCS, CCVS, VCCS, VCVS

BJT4 adds a substrate node, and BJT5 adds substrate and temperature nodes. MOS supports both three and four pin devices.

option1, *option2,* ..., are not required. If they are given, GENESYS will create one model for each option. Additionally, if *var* is not given for an option, the value of the option will be set to 1. For the FET example above, GENESYS will make two models, with "_NFET" and "_PFET" added to the base model name. For the NFET model, the value "NFET" will be set to one. Additionally, NFET and PFET will not be shown as parameters in GENESYS. GENESYS will also use appropriate symbols for any recognized option. The following options are recognized by GENESYS:

BJT, BJT4, BJT5: NPN and PNP
FET and JFET: NFET, NJF, PFET, and PJF
MOS: NMOS and PMOS

If *var1,var2,...,* are specified, they are set to the specified value (instead of the option being set to a value). Additionally, the parameters referenced are not shown as parameters in GENESYS. Otherwise, they behave identically to the case above.

EAGLEWARE_LAYOUT keyword

Advanced keyword which allows overriding of footprints or association entries.

EAGLEWARE_OPTIONS keyword

Advanced keyword which allows specification of additional model options.

EAGLEWARE_NAME keyword

Normally, the GENESYS model name is the same as the Verilog-A model (with any device class options added to this base name). The base name can be overridden by:

// %%EAGLEWARE_NAME=*modelname*%%

EAGLEWARE_SWAP12 keyword

Advanced keyword which reverses pins 1 and 2 in GENESYS. Can override DEVICE_CLASS swapping if placed later in the file.

EAGLEWARE_NOSWAP12 keyword

Advanced keyword which prevents pins 1 and 2 from being swapped in GENESYS. Can override DEVICE_CLASS swapping if placed later in the file.

EAGLEWARE_IGNORE keyword

Tells GENESYS to ignore a parameter. For example,

// %%EAGLEWARE_IGNORE=x%%

will cause the parameter x to not be displayed in GENESYS, and the default value will be used in Verilog-A.

Chapter 7 SPECTRASYS (System)

System Models

Linear (Y-matrix) models are created from the behavioral models in SPECTRASYS. These Y-matrix models are used to determine the impedance of each element along the propagating path of the signals. Once the impedance and gain are known, the correct node voltage for a given spectral component can be determined for every node for all elements. Using this technique, VSWR interactions are automatically accounted for.

For the system non-linear devices such as the RF amplifier and mixer (RFAMP, MIXERP, and MIXERA), a Y-matrix model is also used to determine the impedance and gain. However, the non-linear parameters of the models such as P1dB, PSAT, IP3, and IP2 are used to determine the non-linear behavior of the model (such as harmonic/intermod generation, mixing, and gain compression).

Glossary

Adjacent Channel – This is a channel that has the same bandwidth as the main channel but center frequency moved up or down by the channel bandwidth.

ACP - Adjacent Channel Power

Channel – The combination of the channel frequency and channel measurement bandwidth. For example, the channel 99.5 to 100.5 MHz would be specified as a channel frequency of 100 MHz with a channel bandwidth of 1 MHz.

Coherent Signal - Two signals which are at a constant phase offset are coherent. In SPECTRASYS, coherent signals must come from the same source.

Desired Spectrum – This is the spectrum that originated along the specified path and flowing in the same direction as the path.

IF – Intermediate Frequency.

IIP3 - Input Referenced Third Order Intercept.

Image Channel – This is the channel defined by the image frequency of the first mixer and the channel bandwidth. For example, the channel 1000 to 1001 MHz would have an image channel of 800 to 801 MHz if an LO Frequency of 900 MHz was specified for the mixer.

LO – Local Oscillator.

Offset Channel – User defined channel relative to the main channel. For example, an offset channel specified as –50 MHz for a main Channel Frequency of 125 MHz would result in a channel of 75 MHz ± ½ Channel Bandwidth.

OIP3 - Output Third Order Intercept.

LO Side Injection – The relative indication of the LO frequency with respect to the mixer RF frequency. The RF frequency can be either the input or the output of the mixer.

For example, if the mixer took a 1000 MHz and down converted it to a 100 MHz IF then an LO frequency of 900 MHz is Low Side LO injection and an LO frequency of 1100 MHz is High Side LO injection.

MDS – Minimum Detectable (Discernable) Signal which is equivalent to -174 dBm/Hz + System Noise Figure + 10 Log(Bandwidth)

Non-Coherent Signal - Two signals which are not at constant phase offset are not coherent.

Path – The course a signal takes from the source node to the destination node.

RBW - Resolution Bandwidth

SFDR – Spurious Free Dynamic Range

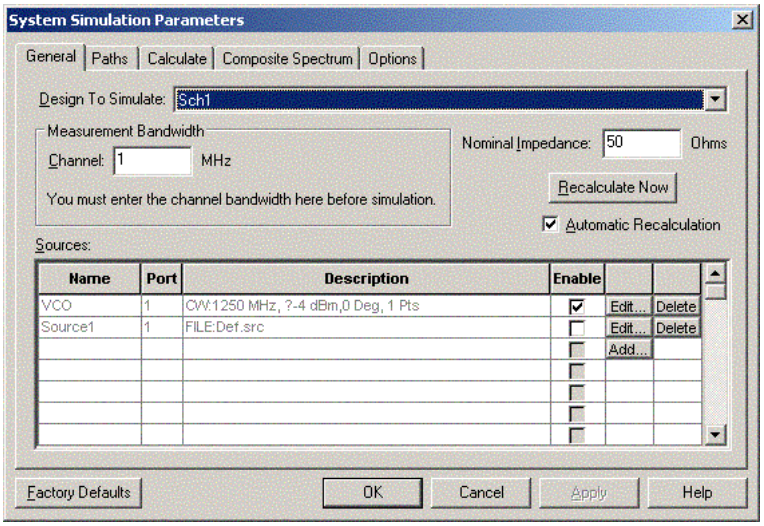
Undesired Spectrum - Any spectrum flowing in a direction opposite of the path direction.

Dialog Box Reference

System Simulation Parameters - General Tab

This page sets the general settings for a SPECTRASYS Simulation. To reach this page, add a System Simulation by right-clicking on Simulations in the Workspace Window.

> **Tip:** Any of the parameters in this dialog box can be made tunable by placing a '?' in front of the parameter.



Design to Simulate - The schematic to use for the system simulation.

Channel Measurement Bandwidth - Specifies the integration bandwidth of the all channels used in SPECTRASYS.

Nominal Impedance - The default system impedance.

Recalculate Now - Closes this dialog and initiates an immediate recalculation of the system simulation.

Automatic Recalculation - When checked, enables SPECTRASYS to automatically recalculate the simulation on an as-needed basis.

Sources - Grid that defines the system simulator signal sources.

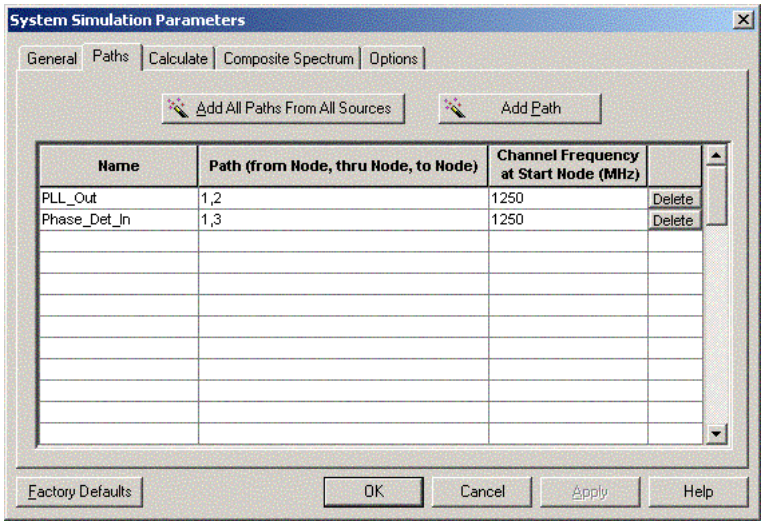
> **Name** - Name of the signal source.
>
> **Port** - Port to attach the signal to. Note that more than one signal can be present at a port.
>
> **Description** - Description of the signal source.
>
> **Enable** - Enables / disables the source in the system simulation.
>
> **Add** - Add a new source.
>
> **Edit** - Edit the current source.
>
> **Delete** - Delete the current source..

Factory Defaults - Will restore the factory default values and options for the system analysis.

System Simulation Parameters - Paths Tab

Many measurements require the definition of a path. For an overview of Paths, see the Paths section later in the Simulation manual.

Two functions exist on the "System Simulation" dialog box (shown below) to aid the user in specifying the path. The first is an "Add Primary Paths" button. All possible port-to-port paths will be added to the "System Simulation" for all ports that have a source defined. If no sources have been defined then no paths will be added. If the number of paths becomes very large then the user will be prompted before adding the paths. The second is an "Add Path" button which will prompt the user for the 1) Path Name, 2) From Node, and 3) To Node.



Add All Paths From All Sources - Automatically adds all possible paths between inputs (ports with signal sources) and input/output ports (all ports).

Add Path - Invokes a wizard to assist the manual creation of a path.

Name - Specifies the path name. This name is used in output graphs to select the path's data.

Path (from Node, thru Node, to Node) - a sequence of node numbers is specified here. The system simulator chooses the shortest path which goes through the specified nodes in order.

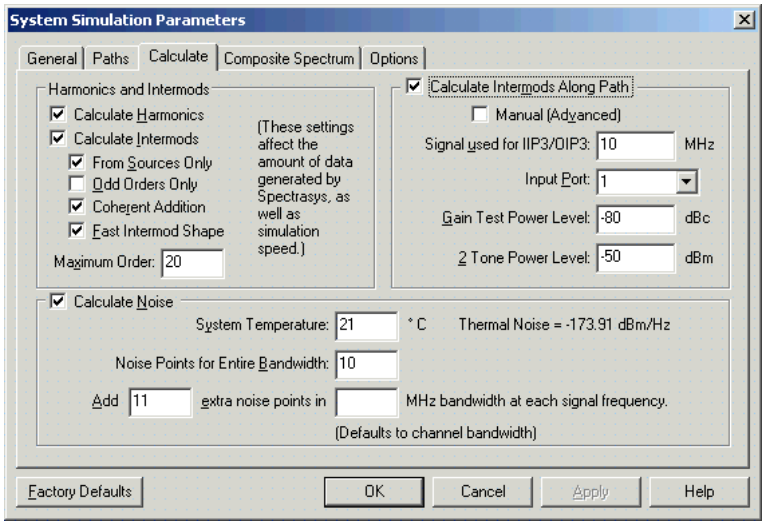
Channel Frequency (MHz) - Specifies the path frequency at the start node. By default this parameter is blank which means that SPECTRASYS will use the frequency of a single source located on the *start node*. If there is more than one source located on the start node the channel frequency is ambiguous and cannot be determined. An error will appear in this case and the user must specify the desired channel frequency. This frequency and the 'Channel Measurement Bandwidth' make up the main channel for this path.

Delete - Paths can be deleted by clicking the Delete button.

System Simulation Parameters - Calculate Tab

This page controls calculation of Intermods, Harmonics, and Noise.

> **Tip:** Any of the parameters in this dialog box can be made tunable by placing a '?' in front of the parameter.



Intermods and Harmonics - For speed calculation reasons the calculation of harmonics and intermods can be disabled. By default they are both enabled. The user can also specify what types of signals will create intermods and harmonics. See the 'Calculate Intermods, Harmonics' section for more information.

Harmonics - Harmonics will be created by the non-linear elements when this option is selected. Calculation time for harmonics is typically very quick.

Intermods - Intermods will be created by the non-linear elements when this option is selected. Intermod Simulation time depends on the number of input signal, levels of the resulting intermods, and number of non-linear stages.

From Sources Only - When this option is selected harmonics and intermods will only be created from source signals that initiated at a port. All undesired products created along the path will be excluded from the calculation of harmonics and intermods. Conversely, when unchecked.

Harmonics and intermods will be created from all signals appearing at the input to the non-linear element. This includes intermods, harmonics, and other undesired signals. This option typically requires longer simulation time since more spectral components are being created. For long simulation times see the 'System Simulation Tips' section.

Odd Order Only - When checked only odd order intermod and harmonics will be created. the fewer number of intermods created the faster the simulation will be.

Coherent Addition - When checked SPECTRASYS will determine coherency of cascaded signals such as intermod, harmonics, and mixed signals. Generally, cascaded intermod equations assume coherent

intermod addition. Otherwise, all intermods, harmonics, and mixed signals will assume non-coherent addition. When unchecked only source signals can be coherent and all other derived spectrums, including intermods, harmonics, and mixed signals are considered to be non-coherent. See the 'Coherency' section for more information.

Fast Intermod Shape - When checked Intermods and Harmonics will be represented by only 2 data points. In most cases this will be fine. However, if one desired to examine one of these signals though a filter then more points may be needed to accurately represent the shape of the signal. When unchecked the average number of points from all inputs signals is used to represent the intermod or harmonic.

Maximum Order - This option is used to limit the order of the spectrums created in the simulation. This limit applies to all non linear elements in SPECTRASYS; such as: amplifiers, mixers, multipliers, dividers, etc. Each model has a limitation on the maximum order that it can generate. Please refer to the element help to determine the order limit for each model.

Calculate Noise - When checked SPECTRASYS will calculate noise. The option must be enabled for SPECTRASYS measurements that use noise i.e. Cascaded Noise Figure. Every component in the schematic will create noise. A complex noise correlation matrix is used to determine the noise power for each element at every node.

System Temperature - This is the global ambient temperature of the entire schematic under simulation. This is the temperature needed to determine the thermal noise power level. For convenience SPECTRASYS will automatically calculate the resulting thermal noise power and display it just below the edit field.

Thermal Noise - Automatically calculated info display that shows the thermal noise power given the specified temperature.

Noise Points for Entire Bandwidth - This is the number of points used to represent the entire band of noise. Noise will automatically be created beginning at the frequency specified by 'Ignore Frequency Below' and ending at the frequency specified by 'Ignore Frequency Above'. These noise points will be uniformly distributed across this bandwidth.

Add Extra Points - This is the number of extra noise points that will be inserted across the 'In Bandwidth' parameter. These additional noise points will be uniformly distributed across this bandwidth. The center frequency of these noise points is the signal frequency. These noise points will be added to every desired spectrum created in SPECTRASYS. However, unused noise points will be removed to improve the simulation time. See 'Broadband Noise' for additional information about Smart Noise Point Removal.

In Bandwidth - This is the bandwidth where extra additional noise points can be inserted. The center frequency of these noise points is the frequency of each desired signal. This parameter is used when the user wants greater resolution of the noise ... like through a narrowband Intermediate Frequency (IF) filter. This bandwidth defaults to the channel bandwidth if this parameter is left blank.

Noise Simulation Tips: The more noise points used in the simulation the longer the simulation time will generally be. Since each component generates noise the more components in a schematic will also increase the simulation time. Better speed performance can be achieved for a large number of components by disabling noise calculations or reducing the number of simulation points.

Calculate Intermods Along Path - When checked SPECTRASYS will calculate input and output third order intercept points. Intercept calculations are based on the actual power of closest interfering tone and the total intermod power within the channel instead of the cascaded intermod equations. All intercept and intermod levels will be based on the actual tone levels. Cascaded intermod equations make dangerous assumptions that interfering tones are never attenuated. Consequently, cascaded intermod equations give erroneous results when modeling an entire receiver since IF filters typically attenuate the interfering tones. SPECTRASYS doesn't suffer from this weakness and all intermod spectrums and measurements will be based on actual spectrum powers.

Manual - When checked the 'Calculate Intermod Along Path' mode is entered and the user must specify the interfering tones as well as a desired signal in the main channel so the correct in-channel gain can be determined. When unchecked the automatic IIP3 mode is entered and SPECTRASYS will create the two interfering tones and calculate the results. Some of the following additional parameters are needed to complete the automatic IIP3 simulation.

Signal used for IIP3/OIP3 - (automatic and manual) Specifies the offset frequency from the path channel frequency where the interfering tone is located that will be used to calculate the Input and Output Third Order Intercepts (IIP3 and OIP3). (automatic) In the automatic mode this frequency offset specifies the spacing between the desired signal, which is in the channel, and the first interfering tone. Furthermore, this will also be the spacing between the two tones. This spacing guarantees that the two interfering tones will create intermods in the channel. (manual) If many interfering tones are used in an analysis then the user has the freedom to choose which interfering signal will be used for intercept calculations. All intercept calculations will be based on the signals located at this offset.

This offset is used to determine the 'Tone Channel Frequency' that is used to determine the 'Tone Channel Power'.

Input Port - (automatic only) This is the port number where the two interfering tones will be created.

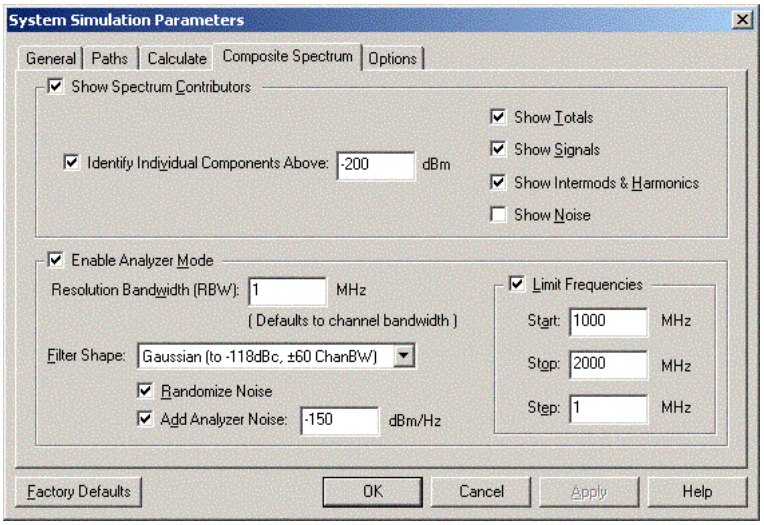
Gain Test Power Level - (automatic only) This is the power level of a signal that will be created within the channel to determine the gain of the main channel. This particular power level is not that critical. However, this level should be low enough so that there is no question that the nonlinear devices such as the amplifiers and mixers are operating in the linear range. This in-channel gain is needed to determine the correct input third order intercept point.

2 Tone Power Level - (automatic only) This is the actual power level of both interfering tones. Remember that the power level of these tones can drive the non-linear devices into compression.

System Simulation Parameters - Composite Spectrum Tab

This page controls calculation of Composite Spectrum, Components, and Spectrum analyzer mode.

Tip: Any of the parameters in this dialog box can be made tunable by placing a '?' in front of the parameter.



Show Spectrum Contributors - This is a graph and table viewing option that will allow the user to determine what is displayed. This option only affects the displayed output and will not affect any internal calculations. When checked the spectrum at a node can be broken down into general groups or individual components when displayed. When unchecked only totals will be shown. Spectrum identification will only occur when this options is enabled and the 'Identify Individual Components Above' options is also enabled.

Identify Individual Components, Above - This threshold is used to show only individual spectral components above the given threshold. This parameter is mainly used to reduce clutter on the graph such as the case when lots of spectral components appear on the same graph. ***NOTE: This option must be selected in order to view the origination and identification of spectral components.*** See the 'Identifying Spectral Origin' section for more information.

Show Totals - Shows a trace representing the total power traveling for each direction of travel through a node. For example, if three elements were connected at a particular node then power would be flowing in three different directions. A unique color would represent each trace.

Show Signals - Shows a trace for each intentional signal source that was applied to an input or output port.

Show Intermods and Harmonics - Shows a trace for each intermod and harmonic spectral component.

Show Noise - Shows a trace for each noise component.

Enable Analyzer Mode - This checkbox enables the analyzer mode and its settings. This mode can help the engineer visualize what the simulated spectrum would look like on a common spectrum analyzer. The analyzer mode has been added to allow the user to correlate the simulation data with spectrum analyzer data measured in the lab. This mode affects only the graphed results and in nowise will affect the integrated measurements.

Resolution Bandwidth (RBW) - The analyzer mode can be thought of just like a spectrum analyzer that has a sweeping receiver that peak detects the total power within the resolution bandwidth. For the analyzer mode the user can specify the resolution bandwidth of this sweeping filter. The default resolution bandwidth is the 'Measurement Channel Bandwidth' if no value has been specified.

Filter Shape - This parameter determines the shape of the resolution bandwidth filter that is used for integration. This filter shape is analogous to the resolution bandwidth filter shape in a spectrum analyzer. However, a brickwall filter can be created theoretically and is implemented in the software as a user selection. Furthermore, a more realistic filter can also be selected which is created from a Gaussian 3 element lowpass prototype. The user is able to select three widths for this particular filter which are based on an integer number of channel bandwidths. No spectrum integration will occur outside the width of this filter. This filter width is used to reduce the amount of data collected, saved, and processed by SPECTRASYS.

Brickwall (Ideal) - This filter is an ideal rectangular filter whose skirts are infinitely steep.

Gaussian (to -100 dBc, ± 30 Chan BW) - Data will be ignored that is farther than 30 channels away from

the center frequency. With this 3 element lowpass prototype the attenuation 30 channels from the center will be about -100 dBc.

Gaussian (to -117 dBc, ± 60 Chan BW) - Data will be ignored that is farther than 60 channels away from the center frequency. With this 3 element lowpass prototype the attenuation 30 channels from the center will be about -117 dBc.

Gaussian (to -150 dBc, ± 200 Chan BW) - Data will be ignored that is farther than 200 channels away from the center frequency. With this 3 element lowpass prototype the attenuation 30 channels from the center will be about -150 dBc.

Randomize Noise - This checkbox enables random noise. When enabled, random noise will be added around the resulting analyzer sweep. In this way the output will be more representative of a typical spectrum analyzer, at the expense of additional computation time.

Add Analyzer Noise - This checkbox enables the analyzer noise floor. All spectrum analyzers have a limited dynamic range. They are typically limited on the upper end by intermods and spurious performance at the internal mixer output. On the lower end they are limited by the noise of the analyzer. This noise is a function of the internal architecture of the specific spectrum analyzer and the internal RF attenuator. The user has the ability to enter a noise floor for the analyzer to more accurately represent the data that will be measured in the lab.

Analyzer Noise Floor - When the 'Add Analyzer Noise' is checked analyzer noise will be added to the resulting analyzer trace. This parameter will also aid the user in correlating the simulation results with what would actually be measured on a spectrum analyzer.

Limit Frequencies - This checkbox enables frequency limiting of the analyzer mode. By default the entire spectrum from the 'Ignore Spectrum Frequency Below' lower frequency limit to the highest frequency limit of 'Ignore Spectrum Frequency Above' will be processed by the analyzer for every node in the system. In some cases this may be very time consuming. In order to improve the simulation speed and just process the area of interest, frequency limits can be enabled to restrict the computation range of the analyzer.

Start - This is the beginning frequency of the analyzer.

Stop - This is the ending frequency of the analyzer.

Step - This is the frequency step size between analyzer data points. The step size can be reduced until the

maximum number of simulation points of 20,000 is reached. .

Number of Simulation Points: The number of simulation points used for the graph is determined internally in SPECTRASYS. This parameter cannot be changed by the user. Since SPECTRASYS can deal with large frequencies ranges, the amounts of data collected for a single spectrum analyzer trace could be enormous. Furthermore, the analyzer function is not a post processing function and the number of simulation points cannot be changed without rerunning the simulation. In order to better control the amount of data collected, which is proportional to the simulation time, SPECTRASYS internally determines the number of simulation points to use.

Simulation Speed-Ups: During the system simulation the analyzer will create an analyzer trace for direction of travel for every node in the system. Consequently, for systems with large number of nodes, the convolution routines used to calculate the analyzer traces alone can be time consuming if the analyzer properties are not optimized. If simulation speed is important then using the narrowest filter shape will have the best simulation speed.

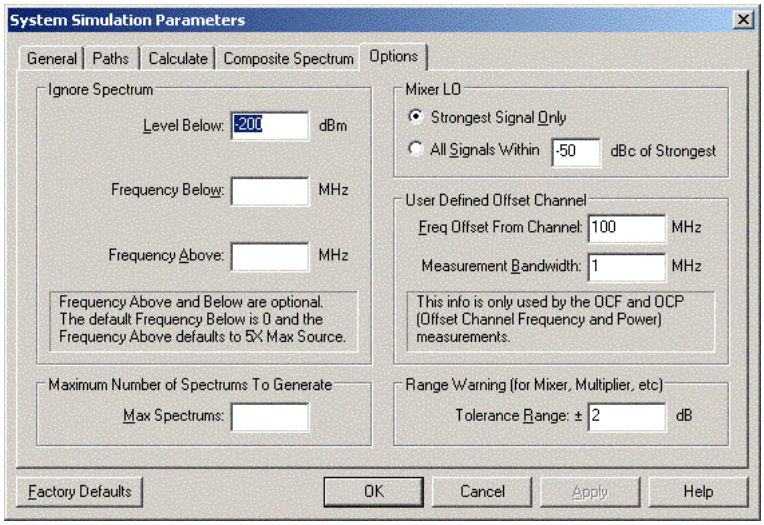
File Size: The size of the data file will increase when the analyzer mode is enabled. Furthermore, the file size can grow rapidly depending on the settings of the analyzer mode. For example, the smaller the resolution bandwidth the more data points are needed to represent the data, the larger the data file will be, and most likely the simulation time will increase.

Analyzer Troubleshooting: *What does it mean when the signal doesn't seem to be lined up with the integrated spectrum?* All this means is the frequency resolution isn't small enough to accurately represent the signal of interest. If this is the case, there are a few things that can be done to increase this resolution. First, the resolution bandwidth can be reduced. If this is inadequate, the 'Limit Frequencies' feature should be enabled and the user can specify the 'Start', 'Stop', and 'Step' frequencies used for the analyzer.

System Simulation Parameters - Options Tab

This page contains miscellaneous SPECTRASYS options.

Tip: Any of the parameters in this dialog box can be made tunable by placing a '?' in front of the parameter.



Ignore Spectrum - This group is used to limit or restrict the number of spectrums created by SPECTRASYS. These thresholds apply at every calculated node. Consequently, if a signal is heavily attenuated or outside the given frequency range during a portion of the path and are then amplified or frequency translated back into the given frequency range then these thresholds must be set so that the spectrums will not be ignored along the calculation path. Once an individual spectrum is ignored it will not continue to propagate. However, all spectrums previously calculated will still be available at the nodes where there were within the specified limits. For example, If we had a 2 GHz transmitter that had an IF frequency of 150 MHz and we set the 'Ignore Frequency Below' limit to 200 MHz then the entire IF signal would not be present and consequently neither would the 2 GHz RF signal.

Level Below (default = -200 dBm) - All spectrums that are below this threshold will not be created by SPECTRASYS. This threshold should be set to the highest acceptable level if optimal speed is an issue. Spectrums are not actually ignored if they are not more than about 20 dB below this threshold since several spectrums can be added together to give a total result that would be greater than this threshold.

Frequency Below (default = 0 Hz) - All spectral components whose frequency is below this threshold will be ignored and will not be created. Spectrums falling below this limit will not continue to propagate. However, there are several cases where negative frequencies may be calculated at interim steps (i.e. through a mixer) which will be folded back onto the positive frequency axis. This parameter will only affect the final folded frequencies and not the interim frequency steps. Likewise, this is the lower noise frequency limit.

Frequency Above (default = 5 times the highest source frequency) - All spectral components whose frequency is above this threshold will be ignored and will not be created. Spectrums falling above this limit

will not continue to propagate. Likewise, this is the upper noise frequency limit.

Simulation Speed-Up: As with any other type of simulation the larger the number of spectral components that need to be processed the more time the simulator will take. Setting these limits to only calculate the frequencies and amplitude ranges of interest can speed up the calculation process ... especially when calculating intermods. However, take caution when setting these limits so that intentional spectrums are not ignored.

User Defined Offset Channel - This group is only used in conjunction with the 'Offset Channel Frequency' and 'Offset Channel Power' measurements.

Freq Offset From Channel - This is the relative frequency offset from the current channel frequency.

Measurement Bandwidth - This is the integration bandwidth for the 'Offset Channel Power' measurement.

See 'Offset Channel' for additional information.

Maximum Number of Spectrums To Generate - This group is used to limit or restrict the maximum number of spectrums that will be created by SPECTRASYS.

Max Spectrums - Limits the maximum number of spectrums that SPECTRASYS will create. Once this limit is reached during a simulation no additional spectrums will be created. This option must be used with care since a premature limitation of the number of total spectrums may affect the accuracy of the measurements.

Range Warning (for Mixer, Multiplier, etc) - This group is used to control range warnings used by some elements in SPECTRASYS.

Tolerance Range - This threshold range is used by some elements in SPECTRASYS to warn the user when a given power level falls outside the specified range. This range applies to each element on a case by case basis. For example the total LO power for the given mixer will be determined by integrating the LO spectrum and then comparing this power level to the 'LO Drive Level' for the given mixer. If this power level is outside the 'Tolerance Range' window then a warning will be issued for this mixer either indicating that the mixer is being starved or over driven.

Mixer LO

Strongest Signal Only - When selected, the frequency of LO signal is used to determine the output frequency of all mixed signals regardless of the number of other signals that may be present on the LO.

All Signals Within X, dBC of Strongest - When selected, all frequencies of LO signals falling within the specified range of the peak LO signal will be used to create new mixed output spectrum. Reciprocal mixing (blocking) can be simulated with this option by creating a 2nd LO signal that represents the LO phase noise at a specified frequency offset. The LO signal range must be set to include this signal which will be below the peak LO signal. When the simulation is ran, the reciprocal mixed spectrum will appear at the mixer output.

Note: All parameters on this page will support equations and can also be made tunable by placing a question mark in front of the parameter value.

How it Works

Amplifiers

Amplifiers

This section will describe the fundamental operation of how SPECTRASYS simulates RF amplifiers.

General RF and VGA (Variable Gain) Amplifier Parameters

Gain - Small signal low frequency gain. The actual amplifier gain will change according to the gain compression and frequency rolloff of the amplifier.

Noise Figure - Amount of noise added to the circuit by the amplifier. The noise figure is assumed to be flat across frequency. A time domain simulation is performed to determine the noise figure If the amplifier is in compression.

Output P1dB - Output 1 dB compression point.

Output Saturation Power - Output saturated output power.

Output IP3 - Output third order intercept.

Output IP2 - Output second order intercept.

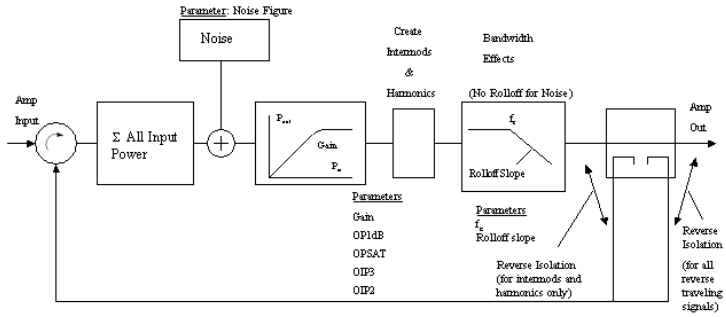
Reverse Isolation - Attenuation from the output to the input. The reverse isolation is assumed to be flat across frequency.

Reference Impedance - Input and output impedance of the amplifier.

Corner Frequency - The frequency at which the input signals will begin to be attenuated by the 'Rolloff Slope dB/Decade' parameter.

Rolloff Slope dB/Decade - The slope of the frequency rolloff specified in dB attenuation per decade in frequency.

The following diagram is a high level view of the operation of the amplifier.



RF Amplifier Model (using ideal elements)

The amplifier operation is as follows:

1. Determine Total Input Power - The entire input spectrum of the amplifier is integrated to determine the total input power.
2. Add Noise - Noise is added to the input spectrum. The noise may be modified as the amplifier enters compression.
3. Determine Amplifier Gain - The actual gain of the amplifier will depend on how close the amplifier is to compression and saturation. A polynomial curve fit is done between the small signal linear gain curve, and the output P1dB and saturation points to determine the actual gain curve of the amplifier. Using the input power and the non-linear polynomial gain curve, the actual gain of the amplifier can be determined.
4. Create Intermods and Harmonics - Using the non-linear parameters of output P1dB, Saturation Power, IP3, and IP2, intermods and harmonics will be created. See the Calculate Intermod and Harmonics section for additional information.
5. Frequency Rolloff - All signals, intermods, and harmonics will be attenuated as a function of frequency according to the attenuation slope that begins at the the corner frequency. NOTE: Noise will bypass this step and will not be rolled off with frequency.
6. Reverse Isolation for Internally Created Intermods and Harmonics - Once intermods and harmonics have been created and rolled off with frequency, these intermods and harmonics will appear at the amplifier input and continue to propagate backwards through the system.
7. Reverse Isolation for Reverse Traveling Signals - Reverse isolation will be applied to all reverse traveling signals that encounter the amplifier output before its input.

Hybrid Linear / Nonlinear Model

This section describes the fundamental operation of how SPECTRASYS simulates the hybrid linear / non-linear model. The hybrid model is available for certain non-linear models in SPECTRASYS. Some examples of non-linear models are the RF Amplifier, Switches, Attenuators, etc. This mode of operation is enabled when S-parameters or a

sub-network is substituted for the given SPECTRASYS non-linear model. For example, if an RF Amplifier behavioral model (RFAMP) was placed is a schematic and S-parameters where substituted for the behavioral model then the hybrid mode of operation would be enabled.

When the hybrid mode of operation is enabled the linear parameters (from sub-networks or S-Parameters) would be used for all linear characteristics of the model. The non-linear parameters of the behavioral model (such as 1 dB compression, saturation power, and intercept points) would still be used to calculated the intermods and harmonics.

Behavioral Parameters that will Use Linear Parameters

All non-linear SPECTRASYS models may not have all the following behavioral parameters. However, any of the following behavioral parameters will be substituted by the linear parameters of a sub-network or S-parameters.

- Gain
- Noise Figure (Noise parameters should be added to S-parameters files that do not contain them so that noise can be accurately modelled)
- Reverse Isolation
- Input / Output Impedance
- Corner Frequency
- Rolloff Slope

Behavioral Parameters that Cannot Use Linear Parameters

The following nonlinear behavioral parameters will still be used to generate nonlinearities in the hybrid mode of operation.

- Gain (Even though this is a linear parameter this value is needed to determine the operating point of the nonlinearites since they are created before being applied to the linear parameters. This gain is used to convert all output nonlinear parameters to input parameters. This gain should be set to the nominal gain used by the linear parameters for the frequencies of interest. All nonlinear warnings dealing with compression, saturation, and intercept points use this gain value not the linear parameter gain value.)
- 1 dB Compression
- Saturation Power
- Intercept Points

Key Assumptions

In real world nonlinear circuits a transfer function exists between the circuit input and the location where the nonlinearities are created. Furthermore, there will be another transfer function located between the nonlinearity creation point and the circuit output. Given only linear parameters (i.e. S-parameter data file) there is no way of knowing where the

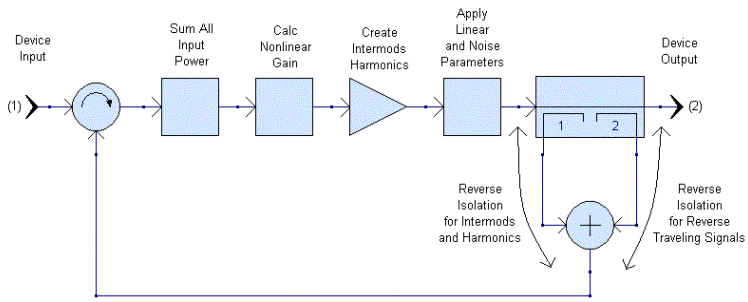
nonlinear creation point is located. Consequently, the assumption is made that all nonlinearities will be created first before being applied to the linear parameters.

Compression and saturation is another problem that has to be dealt with for linear parameters. By definition linear parameters contain no information about compression and saturation. Since harmonics and intermods are generated behaviorally they can follow a smooth Power In vs. Power Out transfer curve. 1 dB compression and saturation will appear as a smooth function with varying input power level. However, since the linear parameters are constant and do not model these nonlinear effects then a hard saturation algorithm was used to limit the maximum output power of the hybrid model.

Consequently, the following assumptions were made:

- All nonlineartities will be created at the device input before being applied to the linear parameters.
- The behavioral gain will be used to refer all output nonlinear parameters to the input.
- Compression, Saturation, and Intermod levels will be based on the behavioral parameters before being applied to the linear parameters.
- Harmonics and Intermods will follow a smooth Power In vs. Power Out curve since they do not use the linear parameters.
- All other signals excluding intermods and harmonics created by this device will have their maximum output power limited to the saturation power. Compression cannot be accurately modelled for this group of signals. However, a warning will be given when the device is in compression.

The following diagram is a high level view of the hybrid linear / nonlinear model.



The device operation is as follows:

1. Determine Total Input Power - The entire input spectrum of the device is integrated to determine the total input power.
2. Determine Gain to Refer Nonlinear Parameters to Input - The actual gain of the device will depend on how close the device is to compression and saturation. A polynomial curve fit is done between the small signal linear gain curve, and the

output P1dB and saturation points to determine the actual gain curve used for the nonlinearities only.

3. Create Intermods and Harmonics - Intermods and harmonics will be created for all input signals to the device. See the Calculate Intermod and Harmonics section for additional information.

4. Linear Parameters - All input signals, intermods, and harmonics will be attenuated as a function of frequency according to the linear parameters.

5. Reverse Isolation for Internally Created Intermods and Harmonics - Once intermods and harmonics have been created and rolled off with frequency using the linear parameters (S21), these intermods and harmonics will appear at the amplifier input after being processed by the linear parameters in the reverse direction (S12) and will continue to propagate backwards through the system.

6. Reverse Isolation for Reverse Traveling Signals - Reverse isolation will be applied to all reverse traveling signals by applying the linear parameters in the reverse direction (S12) that encounter the amplifier output before the input.

7. Add Noise - Noise is added to the device based on the noise parameters contained in the linear parameters.

Channel

Channelized Measurements and Measurement Bandwidth

Over 30 different types of measurements are available for SPECTRASYS. Many of these measurements integrate spectrum power. A frequency and bandwidth are required in order for SPECTRASYS to know where to integrate the spectrums. The Channel Frequency specifies the center integration frequency and the 'Measurement Channel Bandwidth' specifies the range of frequencies to integrate over.

For example, if a power amplifier was designed for a 5 MHz carrier operation in the 2 GHz band, then you must set the 'Measurement Channel Bandwidth' to 5 MHz. If a carrier is injected into the input of the amplifier at 1990 MHz, then all measurements along the path will integrate their spectrums from 1987.5 to 1992.5 MHz, i.e. 1990 ± 2.5 MHz.

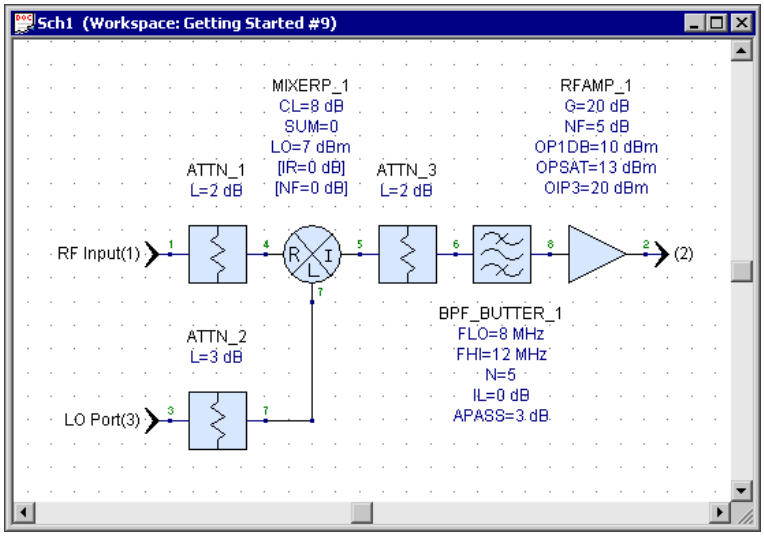
See: System Simulation Dialog box, General Tab

Channel (Path) Frequency

Since each spectrum can contain a large number of spectral components and frequencies, SPECTRASYS must be able to determine the area of the spectrum over which to integrate to determine power levels. A 'Channel Frequency' and a 'Measurement Bandwidth' define this integration area. SPECTRASYS can automatically identify the desired 'Channel Frequency' in an unambiguous case where only one frequency is on the 'from node'. An error will appear if more than one frequency is available. In this case the user must specify the intended frequency for the designated path.

A unique 'Channel Frequency' exists for each node along the specified path. Consequently, each node along the path will have the same 'Channel Frequency' until a

frequency translation element such as a mixer is encountered. SPECTRASYS automatically deals with frequency translation through a mixer. The individual mixer parameters of 'Desired Output (Sum or Difference)' and 'LO Injection (High of Low)' are used to determine the desired frequency at the output of the mixer. A mixer is the only device that causes a frequency translation of the center frequency. For the following schematic, the channel frequency (CF) is shown in the table. Notice that CF is 100 MHz at all nodes before the mixer, and 10 MHz after the mixer (i.e. IF frequency).



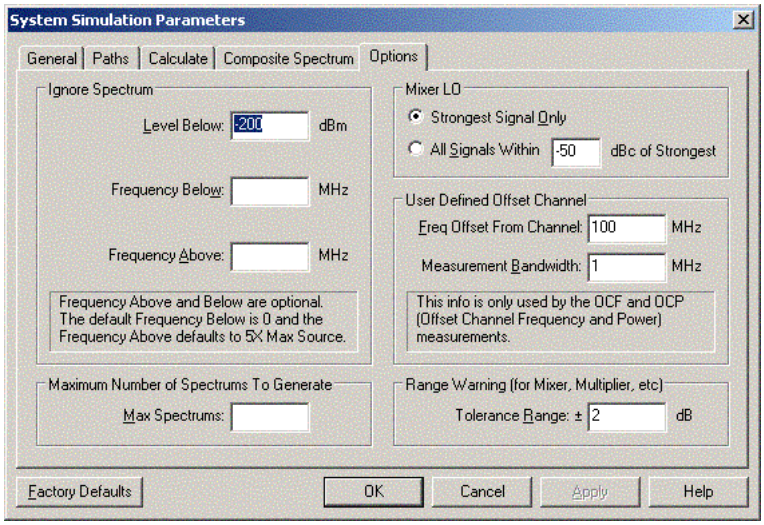
Budget Table (Workspace: Getting Started #9)

	Node	CF (MHz)	DBM[CP]	DBM[CNP]	DB[CNR]	DB[CGAIN]	DB[CNF]
1	1	100	-80	-173.913	93.913	0	0
2	4	100	-82	-175.442	93.442	-2	0.471
3	5	10	-90	-173.278	83.278	-10	10.635
4	6	10	-92	-175.278	83.278	-12	10.635
5	8	10	-92	-175.278	83.278	-12	10.635
6	2	10	-72	-149.301	77.301	8	16.613

The 'Channel Frequency' is a critical parameter for SPECTRASYS since most of the measurements are based on this parameter. If this frequency is incorrectly specified then all measurements using this frequency will be incorrect. The easiest way to verify the 'Channel Frequency' that SPECTRASYS is using is to look at the 'Channel Frequency' measurement in a Table or a Rectangular Graph.

Offset Channel

The offset channel is a special measurement that allows specification of a user defined channel and bandwidth relative to the main channel. The user can specify the 'Freq Offset from Channel' and 'Measurement Bandwidth' parameters on the "Options" page of the 'System Simulation' dialog box.



When these parameters are used in conjunction with the 'Offset Channel Frequency' and 'Offset Channel Power' measurements the user is able to determine the integrated channel power for an arbitrary channel relative to the main channel. Furthermore, both the 'Freq Offset from Channel' and the 'Measurement Bandwidth' parameters can be made tunable by placing a '?' in front the parameter to be tuned.

For example, perhaps you would like to determine the power of some signal at 100 MHz offset from the main channel. The 'Freq Offset from Channel' would be set to '100' MHz and the 'Measurement Bandwidth' could be set to the a user defined bandwidth; for example: 1 MHz. The 'OCF' (Offset Channel Frequency) measurement could be added to a table to show the user the actual frequency being used for the 'OCP' (Offset Channel Power) measurement. It's always a good idea to add this frequency measurement to a table so the user knows that all parameters have been correctly specified. Furthermore, when a mixer is encountered, the user will know exactly which frequency is being used for the offset channel power measurement. The 'OCP' measurement can be added to a level diagram or a table to show the power of the phase noise as it travels along the specified path.

Coherency

Coherency

Since all signals in SPECTRASYS are treated on individual basis so must coherency for each of the spectrums created during the simulation. Coherent signals will add in voltage and phase, whereas non-coherent signals will add in power. For example, if two coherent voltages had the same amplitude and phase the resulting power would be 6 dB higher. If they were exactly 180 degrees out of phase having the same amplitude the two signals would cancel each other. If the two signals where non-coherent then the power would only increase by 3 dB irrespective of the phase.

Some of the coherency of SPECTRASYS can be controlled by the user. The user can determine whether intermods and harmonics add coherently and whether mixer output signals consider the LO signal when determining coherency. See the 'Calculate Tab' of the System Simulation Dialog Box for more information on this setting.

How it Works

When a new spectrum is created a coherency number is assigned to each spectrum. These coherency numbers are used to group spectrums together to determine what the resulting total spectrum is after a coherent addition. Coherent additions are especially important at the input to non-linear devices since the total spectrum from coherent signals will yield a different power than individual spectrums. This total power is needed to correctly determine the operating point of the non-linear devices.

The coherency number of a new spectrum will use an existing coherency number if the two spectrums are coherent. Several rules are followed to determine if a newly created spectrum is coherent with an existing spectrum.

All of these rules must be followed before any two signals can be considered coherent:

NOTE: If the 'Coherent Addition' option is unchecked then all intermods and harmonics will always be non-coherent and well as any mixed products out of a mixer regardless of the following comments.

1. Each source is only coherent with itself.

There are no exceptions to this rule. If two signals are created, even on the same port, by definition they are non-coherent with each other and will always be. To create a coherent signal that drives more than one sub-circuit a single source must be created then split to the appropriate sub-circuits.

2. Signals must be of the same type.

Signals generally have the following categories: Source, Intermod, Harmonic, and Noise. Coherent signals only apply to the same category of signals. For example, a source spectrum can never be coherent with an intermod spectrum and vice versa. Source spectrum can be coherent with source spectrum and intermod spectrum can be coherent with intermod spectrum, etc.

3. Signals must have the same center frequency and bandwidth.

All coherent signals must have the same center frequency and bandwidth. For example, a 2nd harmonic cannot be coherent with a 3rd harmonic since both the center frequency and bandwidths are not the same. However, if we had a cascade of two amplifiers then the 2nd harmonic generated in 1st amplifier would be coherent with the 2nd harmonic generated in the 2nd amplifier for the same signal source. In this case both the center frequency and bandwidth are the same with both harmonics being created from the same signal source.

4. Must have the same LO source (mixers only).

When a new spectrum is created at the output of a mixer SPECTRASYS will determine the coherency of the mixer input signal as well as the LO signal. A new coherency

number will be assigned for unique input and LO signals. If there is more than one mixer in the simulation then coherency numbers for the second mixer may come from the first mixer if the all of the above rules are met for the input signal as well as the LO signal. A good example of this is an image reject mixer. A single input port is split 2 ways that drive the input to 2 mixers. A single LO signal is also split 2 ways and phase shifted before being applied to the mixer LO ports. The mixer outputs are combined back together to form the image reject mixer output. Since both mixers have the same input source as well as LO source then all signals that have the same type, frequency, and bandwidth will have the same coherency number.

NOTE: The coherency number is displayed in the spectrum identification information. This will aid the user in understanding their circuit operation as well debugging any problems. See the 'Spectrum Identification' section for more information.

Coherent vs. Noncoherent Addition

"Coherent addition is more conservative than noncoherent addition, i.e., the coherent assumption indicated a less linear system than the noncoherent equations indicated. In a worst-case scenario, coherent addition should be used."

"When designing low-noise receiving systems, it was found that well-designed cascades usually behave as though the distortion products are adding up noncoherently. For the most part, these system have achieved the equivalent of noncoherent summation plus one or two dB. With wide-band systems, the cascaded SOI [Second Order Intercept] or TOI [Third Order Intercept] will stay at noncoherent levels over most of the frequency range of the system. However, over narrow frequency ranges, the SOI and TOI will increase to coherent summation levels."

"In a well-designed system (where the equivalent intercept points of all the devices are equal), the difference between coherent and non-coherent summation is 4 to 5 dB. When designing a system, it is best to calculate the numbers for both the coherent and noncoherent cases to assess the variation likely to be expected over time and frequency." (McClanning, Kevin and Vito, Tom, *Radio Receiver Design*. Noble Publishing, 2000)

Intermods & Harmonics

Calculate Intermods and Harmonics

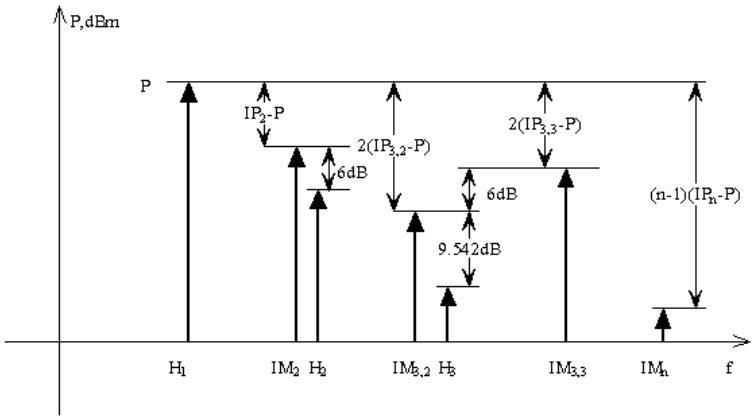
This example will help the user understand how SPECTRASYS deals with intermods and how the nonlinear devices handle these intermods. The user will also understand the difference between generated, conducted, and total third order intermod power. See the 'Amplifier' section for more information about the internal amplifier model used in SPECTRASYS.

Calculated Products

Each model in SPECTRASYS has a limitation on the maximum order that it can generate. Please refer to the element help to determine the order limit.

The following example show the nonlinear second and third order products will be created for each pair of input signals F1 and F2 (listed in increasing frequency assuming F2 is greater than F1):

F2 - F1	2F1	3F1
2F1 - F2	F1 + F2	2F1 + F2
F1	2F2	2F2 + F1
F2		3F2
2F2 - F1		



The relative levels of spectral components for the small signal regime and equal amplitudes of the signal's tones is shown above.

Definitions of symbols

P - Fundamental Tone Power

IP_n - Nth Order Intercept Point

H_1 - Fundamental Tone

H_2 - 2nd Harmonic

H_3 - 3rd Harmonic

IM_n - Nth Order Intermods

$IM_{n,m}$ - Nth Order Intermods due to M tones

SPECTRASYS uses the formulas for calculation of the nonlinear products which correspond to the small signal model (Taylor expansion of the nonlinear characteristics).

2nd Order Intermod Products

The amplitude of the second order intermod products (F2 - F1 and F1 + F2) are equal to the tone power level minus IP2 or in other words IM2 = Ptone - IP2.

2nd Harmonics

The amplitude of the second harmonics are calculated as follows . The amplitude of the second harmonic is equal to the tone power level minus the difference between IP2 (second order intercept) and the tone power level of the device.

3rd Order 2 Tone Products

The amplitude of the third order products (2F1 - F2, 2F2 - F1, 2F1 + F2, and 2F2 + F1) are equal to 2 times the quantity of the tone power level minus IP3 or in other words IM3 = 2 (Ptone - IP3).

Carrier Triple Beats (3rd Order 3 Tone Products)

When more that two carriers are present in a channel, 3rd order intermod products can be created by the multiplication of three carriers. These intermods are called carrier triple beats. SPECTRASYS will create triple beats for all combinations of 3 or more carriers. Working out the math, carrier triple beats will be 6 dB higher that the 3rd order 2 tone products. This calculation of the triple beat level assumes that the amplitude of all input signals is the same. The frequency combinations of the carrier triple beats are as follows:

F1 - F2 + F3
F1 - F2 - F3
F1 + F2 + F3
F1 + F2 - F3

3rd Harmonics

The amplitude of the third harmonics are 9.542 dB below the 3rd order 2 tone products.

Higher Orders

Some models in SPECTRASYS crate higher than 2nd & 3rd order products. The intermod levels and frequencies are calculated based on a complicated mathematical process. This process description is beyond the scope of this text. Please see other resources.[1]

Measurements

SPECTRASYS creates intermods for all input sources driving nonlinear elements such as amplifiers and mixers. Cascaded intermod equations are NOT used by SPECTRASYS. There are two serious drawbacks using the cascaded equations. See the section 'Cascaded Intermod Equations' for additional information. Also see the 'Calculate Intermods Along Path' section.

Linear elements will not create intermods. However, these elements will conduct them from prior stages where they were created.

The 'Total Third Order Intermod Power' (TIM3P) can be separated into two distinct groups of intermods. The first group is 'Generated' intermods and the second is 'Conducted' intermods from a prior stage. SPECTRASYS is able to separate intermods into these two groups. This allows the user to quickly determine the weak intermod link in a cascade of stages. This total is the non-coherent sum of the generated and conducted third order intermod power.

'Generated Third Order Intermod Power' (GIM3P) is the total third order intermod power that is created in a particular stage. This measurement will only show the intermod levels for the stages that created them.

'Conducted Third Order Intermod Power' (CIM3P) is the total third order intermod power conducted from the prior stage. This measurement, when used in conjunction with the 'Generated Third Order Intermod Power (GIM3P)', will identify the stages in the chain that are the weakest link and are the highest contributor to the total intermod power. The stage prior to the stage where the conducted intermods are dominant through the rest of the chain, is the weak link in the chain.

See the SPECTRASYS examples for an illustration of these measurements.

Tone Dissimilar Amplitude

SPECTRASYS automatically accounts for the amplitude of all input signals that create a given intermod. This yields accurate intermod results since frequency response is taken into account which cascaded intermod equations do not.

Channel Bandwidth and Intermods

The bandwidth of third order products is greater than the individual bandwidth of the sources that created them. For example, if two 1 Hz tones were used to create intermods, the resulting bandwidth would be 3 Hz. The bandwidth follows the intermod equation that determines the frequency except for the fact that bandwidth cannot be subtracted. For example, if the third order intermod equation is: Fim3 = F1 - 2*F2 then the equation for the resulting bandwidth would be: BWim3 = BW1 + 2*BW2. If BW1 = 30 kHz and BW = 1 MHz, then the resulting bandwidth would be 2.03 MHz. The user needs to make sure that the 'Channel Measurement Bandwidth' is set wide enough to integrate all of this energy.

Calculate Intermods and Harmonics

A checkbox named 'Calculate Intermods and Harmonics' located on the "Calculate" page of the 'System Simulation' dialog box can be used to disable/enable all calculation of all intermods and harmonics. Simulation speed will be increased for large number of carriers and nonlinear stages if intermods and harmonics are being calculated. The user can disable this option to increase the simulation speed if intermod and harmonic calculations are unimportant.

1. Jose Carlos Pedro, Nuno Borges Carvalho, "Intermodulation Distortion in Microwave and Wireless Circuits", Artech House, 2003

Cascaded Intermod Equations

Cascaded intermod equations are NOT used by SPECTRASYS. There are serious drawbacks using the cascaded equations.

1. They assume interfering input signals are never filtered and maintain the same gain as the desired signal through all cascaded stages. This may be fine for in-band intermod measurements but will be completely inaccurate for out-of-band intermod measurements. Generally, out-of-band interferers in a receiver are filtered in the IF stages. Continuing

the cascaded intermod analysis past the point where the interfering signals are filtered will result in erroneous results.

2. They assume all stages are perfectly matched.

3. They assume a two equal tone analysis.

4. They assume infinite reverse isolation.

5. They don't identify weak links in a cascaded chain.

In SPECTRASYS both desired and interfering signals are created and set to the frequency as they would appear in the real system. These interferers need not be limited to two tones. All intermods created from these signals will be passed on their actual power levels and not erroneous assumptions. Consequently, intermod measurements will be accurate no matter whether the interferers are in-band or out-of-band.

Calculate Intermods Along Path

NOTE - Intermods will always be created and appear in spectrum plots as long as 'Calculate Intermods' is checked.

When checked (located on the 'Calculate' page) enables cascaded ntermod measurements ('Calculate Intermods' must be checked in order to make intermod measurements). Cascaded intermod equations have serious limitations and are NOT used by SPECTRASYS.

Two intermod analysis modes exist in Spectrasys, **Automatic 2 Tone** and **Manual**.

Manual (Advanced) Mode

In this mode the user must create all interfering sources and at least a single desired source. This mode is not restricted to only two tones. The user can specify as many tones as desired and the location of the tone channel (for intercept measurements) so that cascaded intermod measurements will be made. A desired signal at the channel frequency must be created so that SPECTRASYS can determine the correct in channel gain needed to create input intermod measurements from output measurements. The user must specify the location of the 'Tone Channel' (or 'Signal used for IIP3/OIP3') for all intercept measurements.

NOTE - Ensure that interfering signal frequencies are chosen is so that intermods will appear in the main channel. If not intermod measurements will be zero.

NOTE - Set the channel measurement bandwidth to the widest intermod to be measured. Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results.

Check the 'Manual (Advanced)' option on the 'Calculate' page of the system analysis to enable this option.

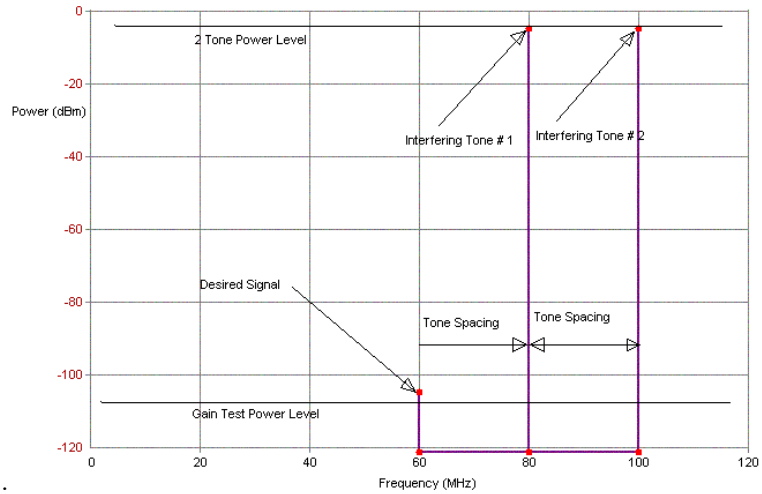
Automatic 2 Tone
In this mode SPECTRASYS will create the 3 sources needed to calculate intermods and intercept points. See the 'Calculate' page for a description of parameters needed to create these signals.

NOTE - These signals and their intermods will be totally transparent to the user when looking at any spectrum plots and can only be accessed through intermod measurements.

This analysis is run as a two step process. The first step will run the simulation as originally specified. However for the second step all signals on the specified input port will be disabled and the 3 new signals will be created. These signals consist of one small signal that falls within the channel that is used to measure channel gain and two other tones separated in such a way to produce itermods within the channel. A 2nd analysis is performed for intermod measurements only.

This mode will set the channel bandwidth to the correct bandwidth to measure all third order intermod power.

This mode is enabled as long the 'Manual (Advanced)' checkbox is unchecked.



.

Mixers

Passive and Active Mixers

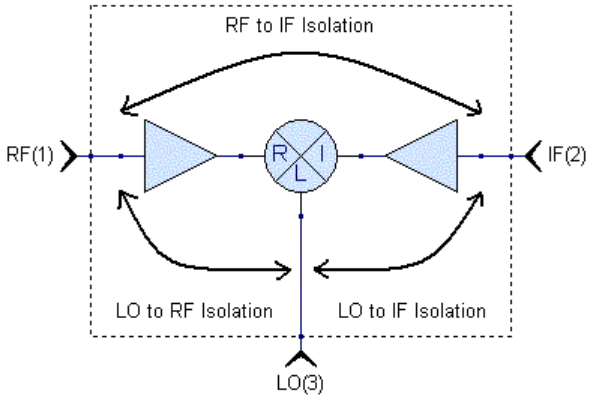
Mixers are key elements in any RF system that translates frequencies like super heterodyne receivers and transmitters. Many times their performance is critical to the proper operation of the system and can be one of the most challenging components to characterize and make behave properly under all system conditions.

SPECTRASYS will aid the user in understanding the output spectrum of the mixer and all of its non-ideal characteristics such as isolation.

Three types of mixers are available in SPECTRASYS, they are: **Passive**, **Active**, and **Intermod Table Mixer**. The only difference between the Passive and Active mixer is respectively the "Conversion Loss" and "Conversion Gain".

See 'RF Mixer' and 'Intermod Table Mixer' in the Element Manual for more specific information about each mixer.

Passive and Active Mixer Model - The mixer can be thought of as two amplifiers (one on the RF port and the other on the IF port) both facing an ideal frequency translation or mixer as shown in the following figure:



The isolation parameters appear port to port and bypass the effects of the amplifiers and ideal mixer. Any signal appearing on any of these ports will be propagated to all of the other ports through the respective isolation. Obviously, no frequency translation is taking place during the isolation calculations.

Signal Spectrum Arriving at the LO Port - All spectrums arriving al the LO port will be propagated (without a frequency translation) to the RF and IF ports with their respective isolations. However, only the highest power level LO signal is currently being used to create the mixed output spectrum of the mixer. Obviously, no mixed output spectrum will be created unless an LO signal is present on the LO pin of the mixer.

If the bandwidth of the LO signal is greater that 25 Hz the mixer will kick into a convolution mode and all highest power LO spectrum will be convolved will all input signals to create mixed output spectrum. The convolution process is much more time consuming and the simulation time will increase.

Currently, the LO power does not affect the conversion loss or gain of the mixer. Instead the power of the LO is determined and compared with the 'Warning Range' specified on the 'Options Tab' of the 'System Simulation Dialog Box'. The user will then be warned if the mixer is being starved or is over driven by the LO.

Signal Spectrum Arriving at the RF Port - All spectrums arriving at the RF input port will be propagated to the IF and LO ports through their respective isolations. Next the actual conversion gain (loss) of the mixer is determined by examining the total power appearing at this port. This total power is used to determine whether the internal amplifier connected to the RF port is being compressed or saturated. For more information on the amplifier model see the 'Amplifier' section. Any gain compression will be applied to that conversion gain (loss) specified in the model along with any VSWR effects. Before any sum and difference frequencies are created the entire non-linear spectrum due to all of the signals arriving at the RF input (intermods and harmonics) are created. Both the sum and difference spectrums will be created from this non-linear spectrum and the peak LO spectrum frequency. Any negative frequencies created during the difference calculations will be shifted by 180 degrees and folded back onto the positive frequency axis. For all signals traveling from the RF port to the IF port the IF port amplifier is effectively bypassed. The non-linear spectrum created by the internal amplifier on this port will also appear on the input port and be propagated backwards due to the reverse isolation of the mixer. It is assumed that the reverse isolation of the mixer is equivalent to the RF to IF isolation.

Signal Spectrum Arriving at the IF Port - Spectrums will be treated identically to the RF port except for the fact that the amplifier on the IF port is used to create the non-linear spectrum appearing at the IF port input and the amplifier on the RF port will be bypassed.

Mixer LO Level Warning - Maintaining proper mixer LO level is important to guarantee the performance of any mixer. Typically, this is a level that can easily be overlooked from one design turn to another. The user must specifically check the LO power level to ensure that the mixer is operating in the expected range. With SPECTRASYS this process is much easier and the user will automatically be notified if the mixer is being over or under driven. The user has control of the LO drive level of each mixer and a global system simulation parameter that will check that the LO power is within a user specified widow of the LO drive level. The user specified mixer LO parameter is the 'Mixer LO Range' specified on the 'Options' tab of the 'System Simulation' dialog box. During system calculations, SPECTRASYS will integrate the entire LO spectrum power and this power will be compared to the mixer LO drive level. If this LO power is outside the specified LO range a local error will be created and the mixer will change color indicating to the user that a potential error has occurred in the mixer.

Desired Output - The 'Desired Output' mixer parameter is only used by SPECTRASYS to determine the desired channel frequency along a path defined through the mixer. This parameter does not affect the operation of the mixer in any way.

LO Drive Level - The 'LO Drive Level' parameter is currently only used by the mixer to determine if the target LO power level of the mixer. This information combined with the 'Warning Range' on the 'Options Tab' in the 'System Simulation Dialog Box' is used to warn the user if the mixer is being starved or over-driven.

Noise Arriving at the RF and IF Ports - A noise source is treated just like any other signal source. However, since this is currently a time-independent simulator noise will not create intermods, harmonics, and be used with reverse isolation.

Broadband Noise

Broadband Noise

SPECTRASYS can process large blocks of spectrum very quickly and broadband noise is no exception. Noise can come from any of three different sources. These are:

1. Thermal noise of passive components
2. Added noise of all components
3. Noise source applied to a port

SPECTRASYS uses the parameters found on the 'Options' page and 'Calculate' page of the 'System Simulation' dialog box to determine the frequency range, power level, and number of points needed to represent the broadband noise.

In the real world noise occurs at all frequencies. Since SPECTRASYS is a continuous frequency simulator we need to provide a way to limit the frequencies of the noise. For example, the users may not be interested in noise at 10 GHz when looking at an 800 MHz system. In other cases this may be necessary. The frequency limits for noise are explained below:

Lower Noise Frequency Limit - Is determined by the frequency set by the 'Ignore Spectrum Frequency Below' parameter. The default for this parameter is 0 Hz.

Upper Noise Frequency Limit - Is determined by the frequency set by the 'Ignore Spectrum Frequency Above' parameter. The default for this parameter is 5 times the highest source frequency.

Thermal Noise Power - This power is determined by the 'System Temperature' parameter. This noise power is used by all of the elements in SPECTRASYS to create their noise power. Obviously, the total noise power in a given bandwidth is equal to the noise power density (dBm/Hz) multiplied by the integration bandwidth. For channelized measurements the 'Channel Measurement Bandwidth' is used. As a result, the total noise power will increase proportionally with the bandwidth.

Number of Noise Points - There are several parameters used by SPECTRASYS to determine the number of points needed to represent the noise. All noise measurements are integrated measurements whose accuracy is totally dependent on representing the noise with enough points. So how does one determine how many noise points are needed? At a first glance the easy solution to this problem is to allow the user to specify the total number noise points and then make the assumption that all noise points are uniformly distributed. This solution doesn't work very well for high frequency simulations that may have narrow bandwidths at a particular frequency ... like an intermediate frequency (IF). Obviously, in order to represent the noise correctly in the narrow bandwidth enough noise points need to be added to eliminate integration errors. On the other hand adding too many noise points will slow down the simulation.

The solution to the noise point problem is to insert noise points using three techniques:

1. Specify the total number of points to be uniformly distributed across the entire noise bandwidth.

2. SPECTRASYS automatically knows which frequencies in the noise spectrum need more points. It will automatically insert them into the noise spectrum so that the noise spectrum around signal sources will be accurately represented even through narrow band filters. It will do this based on the frequencies of all the known sources in the simulation.

3. Additional noise points can be inserted around the channel as specified by the user. The user can specify a noise channel bandwidth and the number of points that will be uniformly distributed in this bandwidth. Once again this is extremely useful when trying to examine the noise spectrum through narrowband devices like filters, etc.

Smart Noise Point Removal - Since the simulation time is proportional to the number of noise points then simulation time can be improved by removing unnecessary noise points. For every desired spectrum (signal source, desired mixer, multiplier, divider, etc product) noise points are added at the frequencies specified on the 'Calculate Tab' of the System Simulation Dialog Box. Once the noise has been processed by a particular element all noise points are examined to determine their amplitude and phase. If consecutive noise points have the same complex values then some of these consecutive noise points are removed.

Noise sources are not discussed in this section. Please refer to the 'Sources Section' for more information about noise sources.

Note: Noise will not be calculated unless the 'Calculate Noise' checkbox in the 'System Simulation' dialog box has been checked.

Paths

Measurements are Defined by Paths

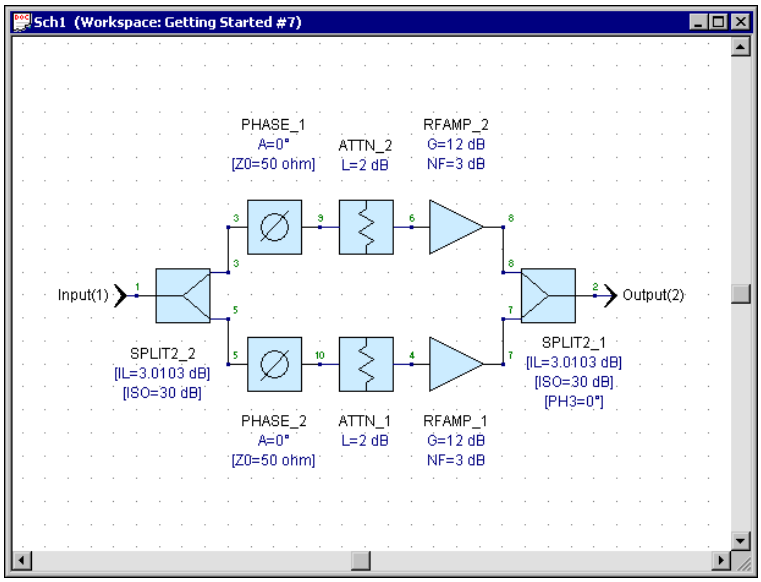
Since spectrums are propagated to every node in the schematic the user must have some way of indicating signal direction in order to make useful measurements. A path is used for just such purpose. Basically, a path is a node number sequence and is defined by specifying:

1. Name (or use the default i.e. Path1)
2. '**from node**'
3. '**thru nodes'** (optional)
4. and '**to node**'

Given the 'from node' and the 'to node' SPECTRASYS will pick the shortest path between the two node even though several paths may exist. If the user would like to look at an alternate path then 'thru nodes' can be specified.

In the example below, if the following two paths existed: a) 1,3,9,6,8,2 and b) 1,5,10,4,7,2 then specifying the path as: "1,2" SPECTRASYS would select path (a). However, if the user wanted to specify path (b) then by simply finding a unique node(s) then this can could be specified i.e. "1,10,2". There is no restriction on the number of nodes used to

specify a path. In some cases several 'thru' nodes may need to be specified to uniquely identify a path.



In order for SPECTRASYS to be able to locate the path a signal source must be present on the 'from node'. If a source has not been created or has been disabled then SPECTRASYS will not be able to locate the path.

The 'from node' and the 'to node' can be any node in the schematic and is not restricted to an input or an output port. However, the first node in the path (node sequence) must be the 'from node' and the last node must be the 'to node'. All nodes in the path must be separated by commas and the 'thru nodes' can be in any order.

Two functions exist on the "Paths" page of the System Simulation dialog box (shown below) to aid the user in specifying the path. The first is an "Add All Paths From All Sources" button. All possible port-to-port paths will be added to the "System Simulation" for all ports that have a source defined. If no sources have been defined then no paths will be added. If the number of paths becomes very large then the user will be prompted before adding the paths. The second is an "Add Path" button which will prompt the user for the 1) Path Name, 2) From Node, and 3) To Node.

See the 'System Simulation Parameters - Paths' section for additional information.

Path Frequency

This is the same as the 'Channel Frequency'. See Channel Frequency for more information.

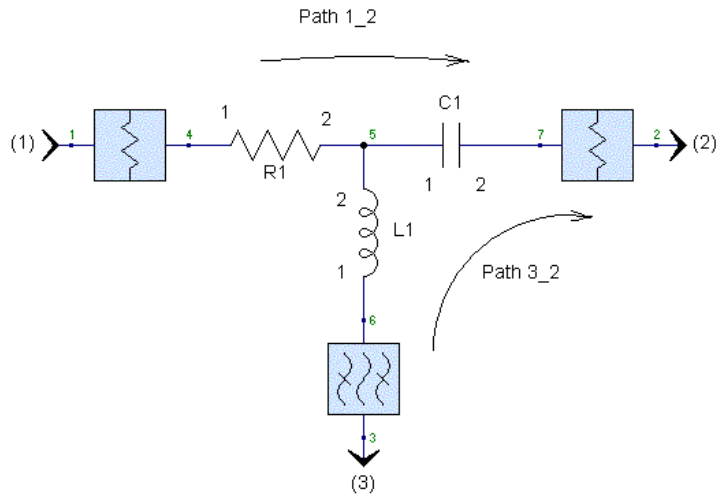
Directional Energy (Node Voltage and Power)

When more than two connections occur at a node a convention must be established in order to make sense of the information contained at the node for viewing a table or a level

diagram. The value that is reported for a node along a path that has more than two elements is the value seen by the series element in the path entering the node.

For example, in the following example we have defined two paths 'Path1_2' which is the path from node 1 to node 2 and 'Path3_2' which is the path from node 3 to node 2. On a level diagram or in a table the value reported at node 5 for 'Path1_2' would be the value of the measurement leaving terminal 2 of the resistor R1 entering node 5. Likewise, the impedance seen along this path is that seen looking from terminal 2 of the resistor R1 into node 5. Consequently, the impedance seen by R1 is the L1 to port 3 network in parallel with the C1 to port 2 network. In a similar manner the value reported at node 5 for the 'Path3_2' would be the value of the measurement leaving terminal 2 of inductor L1 entering node 5. The impedance for the node looking from terminal 2 of inductor L1 is most likely to be completely different from the impedance seen by R1 or even C1 because from the inductors perspective, the R1 to port 1 network is in parallel with the C1 to port 2 network.

SPECTRASYS knows about the direction of all of the paths and will determine the correct impedance looking along that path. As a result all measurements contain the correct values as seen looking along the path of interest. Remember, absolute node impedance and resulting measurements based on that impedance don't make any sense since they are totally dependent on the which direction from which we look into the node.



Transmitted Energy

Transmitted energy is only the energy flowing in the forward direction.

For example, lets suppose that we have a fixed attenuator of 3 dB in series with a bandpass filter that has 50 dB of rejection at 1 GHz which is outside the passband of the filter. Now lets suppose that we are going to look at the power level of this "out-of-band 1 GHz signal" along the path from the attenuator input to the output of the bandpass filter. Intuitively, we would expect to see 3 dB of attenuation of the 1 GHz signal across the 3 dB pad and then and additional 50 dB of rejection across the filter. However, when

we closely examine the impedances and power levels at each node we see things in a slightly different light.

1) The input impedance of the 3 dB pad will not be exactly 50 ohms since its load impedance is the input impedance of the bandpass filter at the input frequency of 1 GHz which can be very low or very high. Consequently, if the applied power level is 0 dBm then the actual power level that will be transmitted through the attenuator (node 1 power) will be lower than the applied power.

2) Since the input impedance of the bandpass filter at the out-of-band frequency of 1 GHz can be very high or low there will be very little power at the input of this filter for this particular frequency that will actually be transmitted through the filter. Most of the energy will be reflected by the filter. Since the input power to the attenuator is very high and the input power to the bandpass filter is very low then it appears that the entire attenuation of the filter appears across the 3 dB pad.

In other words the transmitted energy through any filter will be equal to the insertion loss.

When we realize that the power at each node is the actual power that is transmitted through the element to the next node in the path then the level diagrams make more sense. Another way that we can think of this node power is that this would be the actual power measured at that node with a power meter at that given frequency if the power meter was matched to the same impedance as seen by that load circuit.

Path Spectrum

Along every path there are 5 categories of spectrums that every signal will be part of. These spectrums are: desired, undesired, noise, intermod, and total.

Desired and Undesired Spectrum

The definition of 'Desired Spectrum' is spectrum that is traveling in the same direction as the desired path. All other spectrum originating from other sources will be present at the node of interest but will be specified as 'Undesired Spectrum' since it didn't originate along the desired path direction. Each and every node along the path contains both 'Desired' and 'Undesired' spectrums.

As with a real circuit board, all signal sources propagate their signals to every node in the system since perfect isolation is unrealizable with real components. Consequently, real signals propagate in both directions at every node. SPECTRASYS follows this same model and signals travel in both directions at a every node. However, the user is typically only interested in the RF power traveling in a particular direction. For example, if the user created a schematic of a single conversion super heterodyne receiver the cascaded gain for the primary receive path would only make sense looking in the direction from the receiver front end to the IF output. The direction of the LO radiation along the path from the LO to the receive antenna port would be in a direction opposite that of the received signal. As a result, the 'Desired Spectrum' for the received signal would be in the forward direction (from the receiver front end to the IF output) and the 'Undesired Spectrum' would be any other signal that didn't originate from the receiver front end that is traveling in the reverse direction. However, looking at the signals along the LO radiation path, the

LO signals would be the 'Desired Spectrum' and the received signals would be the 'Undesired Spectrum'.

All signals that are members of the 'Desired Spectrum' and 'Undesired Spectrum' are also members of the 'Total Spectrum'.

See the example "Getting Started #5.wsp" for a good illustration of 'Desired' and 'Undesired' spectrum.

Noise Spectrum

All spectrums created from noise sources in the schematic are placed in the 'Noise Spectrum' and also in the 'Total Spectrum'.

Intermod Spectrum

All spectrums created from intermods between two or more signals are placed in the 'Intermod Spectrum' and also in the 'Total Spectrum'.

Total Spectrum

Every spectrum passing through a node will appear in this spectrum category.

Outputs

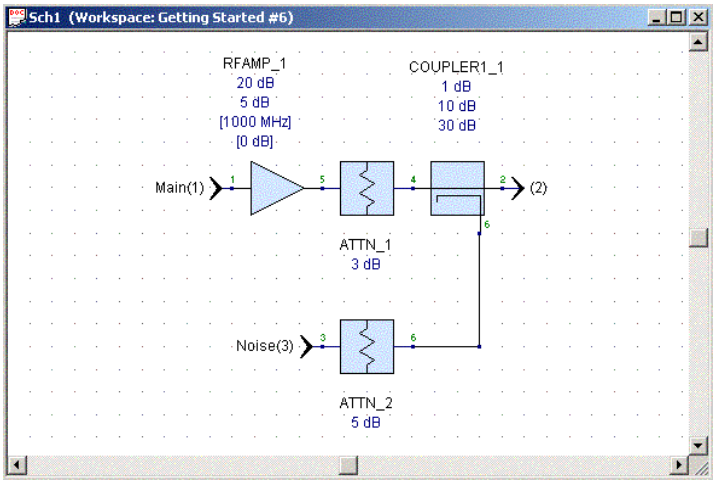
Level Diagrams

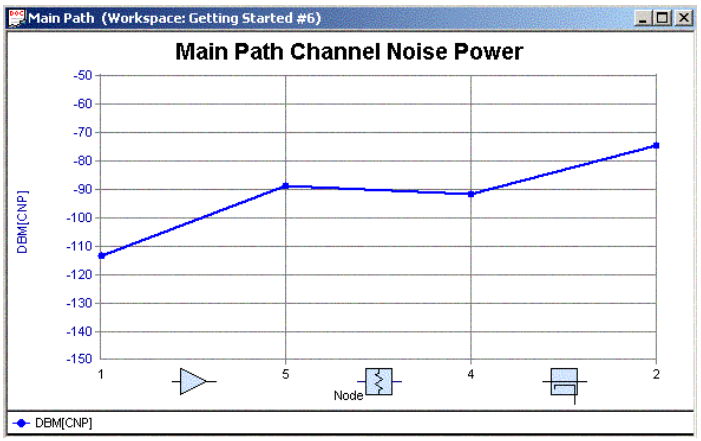
A level diagram is a diagram that can display measurements of cascaded stages along a user defined path. Each horizontal division of the x axis of the graph represents a stage along the path. The first division represents the input to the cascade and the last division represents the output of the cascade. The value of the measurements are displayed on the vertical axis.

The concept of level diagrams has been around for several years. RF designers have used level diagrams for decades to architect and design RF systems. These diagrams have not appeared in commercial RF simulation software until SPECTRASYS. Eagleware's implementation of a level diagram is unique and will help the RF engineer to optimize the RF system performance right from the diagram.

Level diagrams give the user a quick visual indication of the performance of the entire cascade. Node numbers are placed on the horizontal axis to show the node sequence of the path. Furthermore, schematic symbols are extracted from the schematic and placed at the bottom of the level diagram. These are the schematic symbols of the path of the level diagram. The user can change any of the schematic element parameters by double clicking on the desired symbol directly on the level diagram. The element parameters for that device will appear and the user can edit those parameters directly. The effects of these changes are shown immediately on the graph.

For the simple schematic below, the noise power along the main path was of interest. The nodes along the path are 1,5,4 and 2. The level diagram shows the noise power at each node and the schematic element between each node. This schematic symbol alleviates the need to refer back to the main schematic and allows changes to element parameters.





For multiple connections at a node, energy can be traveling in multiple directions. For details on understanding the meaning of the node measurements in this case please refer to the section on Directional Energy.

Level diagrams and tables contain only transmitted energy information. In other words, this is the transmitted energy traveling from node to node along the specified path. See the Transmitted Energy section for additional information.

Composite Spectrum

Composite spectrum is unique feature in SPECTRASYS. A composite spectrum allows the user the ability to view full node spectrums and identify the spectral component origins and their path of travel to the designated node. There are three general spectral categories in SPECTRASYS. They are: 1) Signal, 2) Intermods and Harmonics, and 3) Noise. Furthermore, each source and their derived components (harmonics, intermods, spurs, etc.) will be propagating in all directions when then arrive at a node.

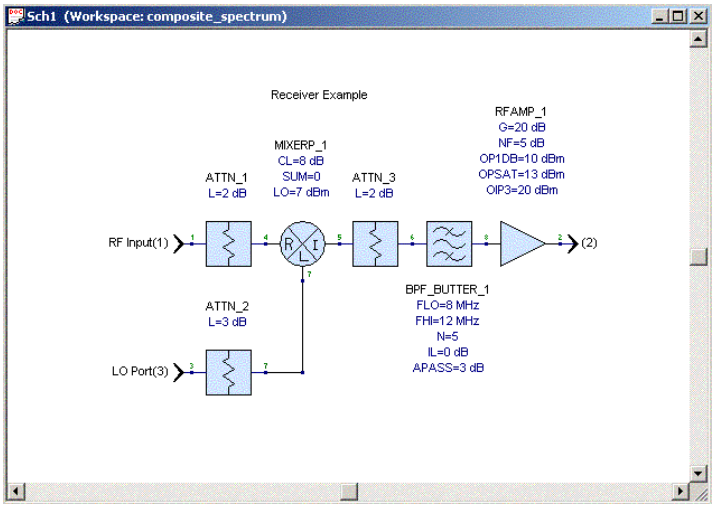
The makeup of composite spectrum consists of a trace for each:

1. Element representing the total power traveling from that element into the node
2. Signal component
3. Intermod and harmonic component
4. Noise component

Because of the way that SPECTRASYS keeps track of these spectral components independently and each component is represented by a trace on the graph, it is not uncommon to have hundreds and even thousands of spectral components (traces) at a single node. The user can determine which of the spectral pieces they would like to see. On the 'Composite Spectrum' tab of the System Simulation dialog box the user can check or uncheck which pieces of the spectrum they would like to see. The total from each element will always be shown and cannot be disabled. Furthermore, all spectral components can be ignored below a user specified level.

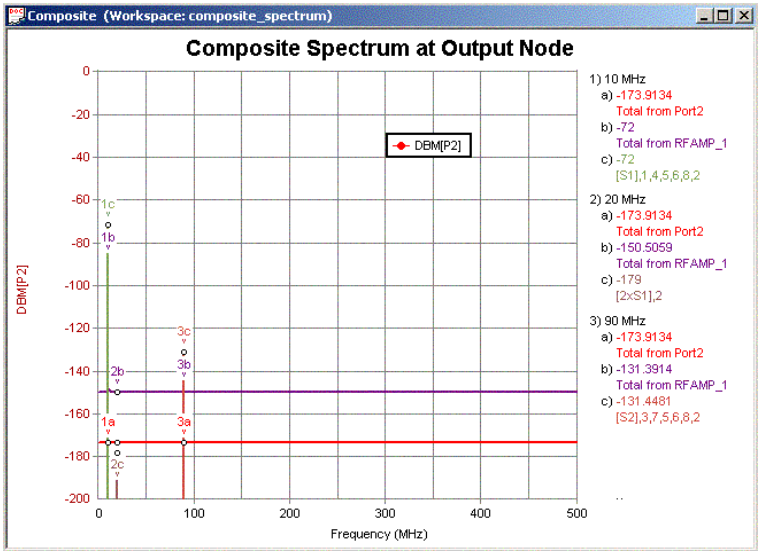
In order to view a composite spectrum plot, the user must select the 'System Simulation' and 'Composite' in the 'Default Simulation/Data or Equations' combo box of the rectangular graph properties (i.e. 'System1.Composite'). The user must then specify whether they want to display a voltage or a power measurement and the node number (i.e. P2 - power at node 2).

For example, a simple receiver (shown below) will have the antenna signal propagate forward through the receiver front end then through the mixer and the IF chain. However, after the LO arrives at the mixer it will propagate backwards through the receiver front end to arrive at the antenna input. In addition, this LO signal will also propagate forward through the IF chain. If we were to examine the receiver input on a spectrum analyzer we would see both the input signal from the antenna as well as the LO leakage. On the spectrum analyzer we would see both of these signals. However, we know that they are traveling in different directions. At the antenna input node we know that the received signal is traveling toward the IF chain and the LO leakage is traveling away from the IF chain.



Unlike a spectrum analyzer, on the SPECTRASYS composite plot we can actually distinguish the direction of travel of all spectral components. Furthermore, a trace that represents the total from all directions in a node is represented. By simply placing the mouse over the trace the user is able to identify which direction the signals are traveling by seeing which element they are coming from. Think of this is as an N-way directional coupler with infinite directivity, so that we only see the signals traveling in the direction of interest. All of these signals from each direction of travel is an independent trace on the composite spectrum plot. For example, if we had three elements connected to a node we would see signals traveling from each of these elements.

For the example shown, the components in the RF input are at 90 and 100 MHz. Notice that the 90 MHz component is identified as from source S2 (i.e. the LO drive). The leakage path is from the LO input port [S2] or node 3, through nodes 7 and 4 (i.e. the mixer LO to RF isolation), and then through attenuator (ATTN_1). The power of -25 dBm is the result of the LO power of 10 dBm attenuated by 5 db and passing through the mixer isolation path with -30 db.



One of the most useful features of composite spectrum is is the ability to identify the origin and path of each spectral component. See 'Identifying Spectral Origin' and 'System Simulation Parameters - Composite Spectrum' for more information.

Identifying Spectral Origin

Since each spectral component is tracked separately and SPECTRASYS knows the direction of travel of all signals, the user can find the origin and path of each spectral component by placing a marker on the graph or simply flying the mouse over the spectral component of interest. When a graph marker is added to a plot, the marker will attach itself to the closest data point. Also the mouse flyover text appears when the mouse is over the marker symbols (trace segment endpoints) or the marker text on the right side of the graph. These marker symbols can be enabled or disabled. The default marker symbols look like large round dots. If the user is having a difficult time trying to get the

mouse flyover text to popup it is because the mouse cursor is not near a marker symbol. The best solution to this problem is to enable the marker symbols so the user can see the marker locations and place the mouse cursor accordingly.

There exists a long and short form of spectral identification. The long form will appear on mouse flyover near the mouse cursor and will also be displayed on the status bar of the GENESYS window. The short form appears in the marker information on a graph.

The format of the spectral identification is as follows:

GENERAL FORMAT

Line 1 - **Marker Frequency, Marker Power (Voltage) Level** (the Frequency and Power appear on different lines for marker text on the right of the graph)

Line 2 - **{Coherency Number} Signal Type [Frequency Equation], Origin Node, Next Node, ..., Current Node** (Short Form)

OR

Line 2 - **{Coherency Number} Signal Type [Frequency Equation], Origin Element, Next Element, ..., Current Element** (Long Form)

Coherency Number

All signals in SPECTRASYS are grouped according to a coherency number. All signals with the same coherency number are coherent with each other and will be treated as such in the simulator. On the 'Calculate' page of the 'System Analysis' dialog box the user can control whether intermods and harmonics signals are added coherently or non-coherently.

Signal Type

D - Desired Signal. All signal types are categorized in SPECTRASYS. Furthermore, all signals are either marked desired or undesired. A good example of this is a mixer where the user will selected either the sum or difference output to be the desired signal and everything else will be undesired.

None - Undesired Signal. If there is no "D" displayed then the signal is an undesired signal.

Frequency Equation

From the frequency equation the user can identify which source frequencies created the spectrum. This equation is written like a typical mathematical equation. The long form of the equation will contain the actual name of the source, whereas the short form uses a short hand notation to indicate the source. The short hand notation is "S" for source plus the index number of the source. For example, S1 would mean the first source listed in the Sources Table of the 'System Analysis' dialog box; S2 the second, and so on regardless of the actual name of the source.

Frequency Equation Examples (Short Form)

[S1+2xS2] - Source1 (first source in the source table) + 2nd Harmonic of Source2 (second source in the source table)

[S1-S2+S3] - Source1 (first source) - Source2 (second source) + Source3 (third source)

[S2+3x[2xS3]] - Source2 (second source) + 3rd Harmonic of a 2nd Harmonic of Source3 (third source)

Sometimes the frequency of the equation may be negative. In this case, the user should simply use the absolute value of this frequency.

Path

The path of the spectral component can be determined by examining the comma delimited sequence of node numbers (short form) or reference designators (long form) which identify the node or element where the spectrum was created and the node or element sequence that the signal took to arrive at the destination node. The first node number or reference designator after the closing frequency equation bracket shows the reference designator or node number where the spectrum first appeared or was created. The subsequent node numbers or reference designators indicate the path that the spectral component took to arrive at the node under investigation.

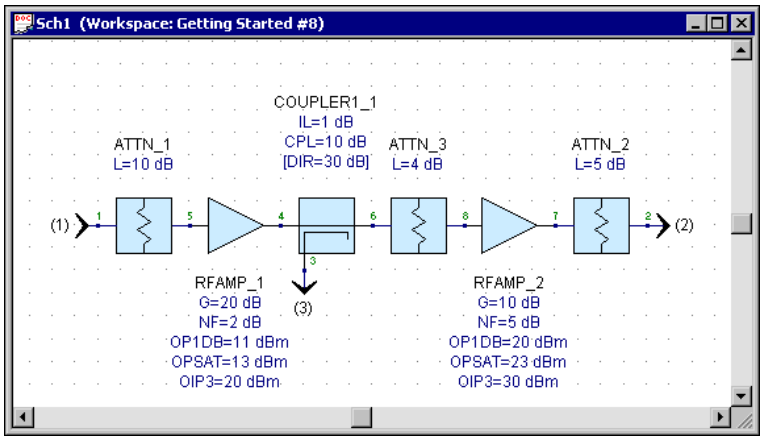
Path Examples

[S1-2xS2],6,7,8,12,5,2 Would indicated that a third order intermod (S1-2xS2) was created at node 6 then traveled through nodes 7, 8, 12, 5 and then arrived at node 2 (which is the current node under investigation).

Feature Activation

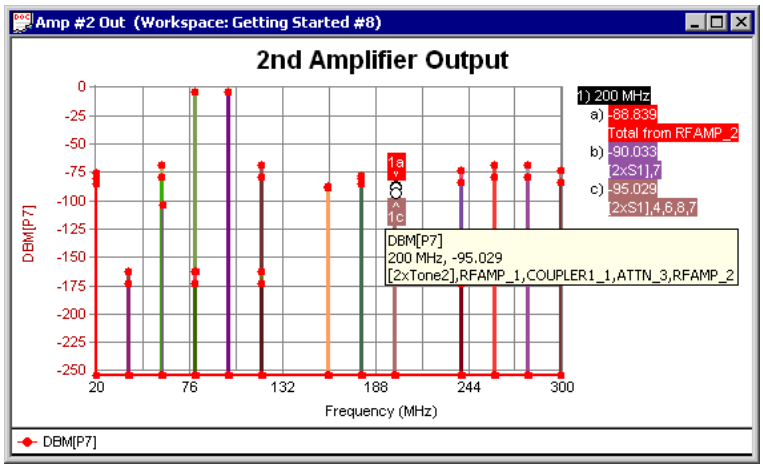
The spectral identification feature is activated from the "Composite Spectrum" page of the System Analysis dialog box. Please refer to that page for activation information.

EXAMPLE: Consider the following schematic that has cascaded amplifiers. Each can create second harmonics of the input signal. If the output power of the second amplifier is plotted, it is easy to identify source and amplitude of each of these harmonics. This is true even though they are of the same frequency. Consider the first source (S1) at 100 MHz at the input node # 1 in the schematic below.



For the power from RFAMP_2 at node # 7, consider the second harmonic (i.e. [2xS1]). One component is generated in the first amplifier and first appears at node # 4. The

designator is [2xS1],4,6,8,7. The amplitude is about -95 dBm as shown either in the flyover box or in the marker text. Notice that the flyover text has the long form of the identifier, whereas the marker text is the short form. The component generated in the second amplifier (RFAMP_2) has the designator [2xS1],7 since it first appears at node # 7. The amplitude is -90 dBm.



TROUBLESHOOTING: If the identifiers do not appear on the graphs make sure the mouse cursor is over the graph data point. Graph data points can be turned on and off through the graph menu or toolbar. Also make sure this option is enabled by checking the "Composite Spectrum" page of the System Analysis dialog box. Make sure that the "Identify Individual Components, Above" box is checked. If intermods or harmonics are desired, put a check mark in the "Show Intermods and Harmonics" box.

[Identification, Creation Equation]

Sources

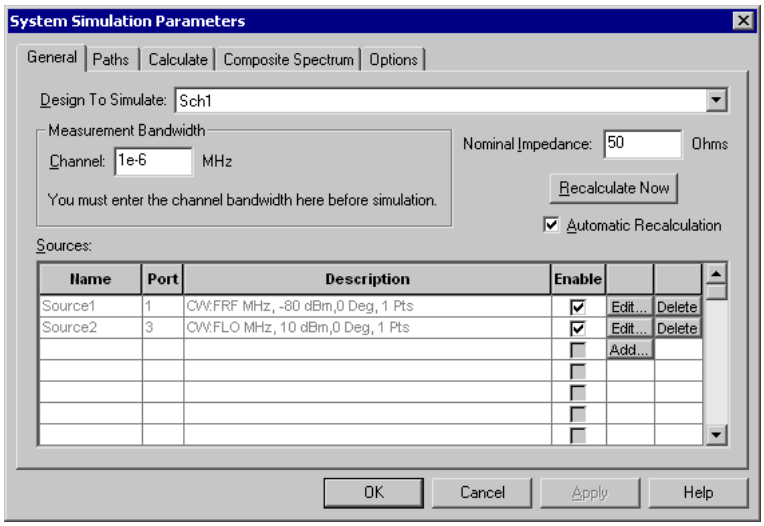
Sources

Sources are a very powerful feature of the SPECTRASYS. There are 4 basic types of sources. They are:

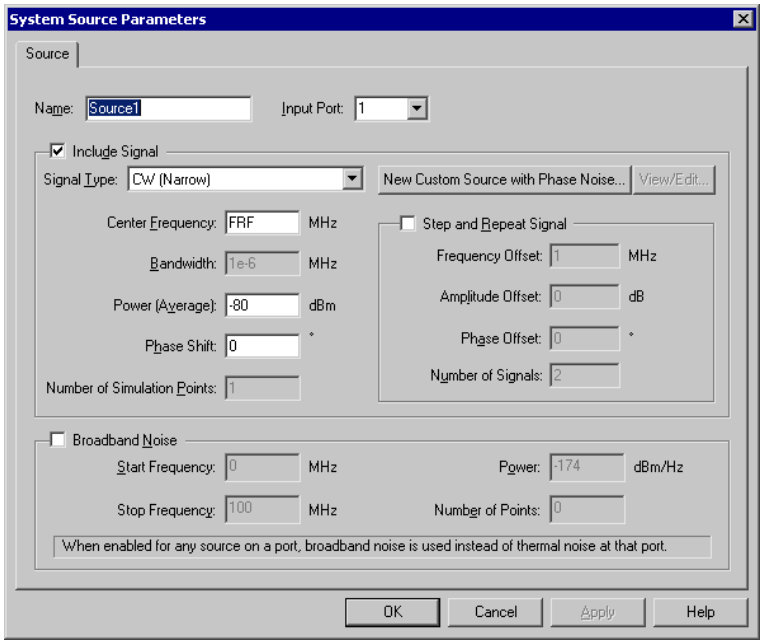
- Continuous Wave (CW)
- Modulated
- Noise
- User Defined

Signal sources (CW, Modulated, and User Defined) are defined by a center frequency, bandwidth, power level, phase shift, and number of points. Every one of these parameters can be tuned by placing a question mark in front of the parameter. All sources are assumed to have a uniform spectral density. Every source can be easily enabled or

disabled by checking or unchecking the “Enable” checkbox in the source table on the “General” page of the System Simulation dialog box.



By clicking on the "Edit" button of any source, the following "System Source Parameters" dialog box comes up. This page is used to enter parameters for each source, as described below.



Center Frequency – This is the center frequency of the source in MHz.

Bandwidth – This parameter is the bandwidth of the source in MHz. The lower frequency of the source is the center frequency minus ½ the bandwidth and the upper frequency of the source is the center frequency plus ½ the bandwidth .

Power Level – This is the average power level of the source in dBm.

Phase Shift – This is the phase shift of the source in degrees.

Number of Points – This is the number of points that represent the source. Most of the time 2 points is adequate to represent the source. However, in cases where the source bandwidth is large and the frequency response of the circuit may affect the bandwidth of the source, the user may want to increase the number of the points.

CW

All CW sources have their bandwidth defined to be 1 Hz. Furthermore, the number of points has been set to 2 points and can not be changed by the user.

MODULATED

A modulated source is currently represented by a uniformly distributed spectrum of constant amplitude. This type of spectrum is currently time-invariant. The user can set the following parameters: center frequency, bandwidth, power level, phase shift, and number of points.

USER DEFINED

The user defined source is a very powerful feature of SPECTRASYS. This is a time-independent source that is defined in the frequency domain. The user can specify the complete source frequency, amplitude, and phase in both relative and absolute values. Relative frequency, amplitude, and phase parameters are entered into a text file (*.src). The values stored in the *.src file are relative to the absolute frequency, amplitude, and phase parameters contained on the system source parameters dialog box.

Having relative values specified in a source file is a great advantage because the absolute center frequency, power level, and phase shift can be tuned from within the system source dialog box. Absolute values would not allow the user the ability to tune these parameters. Furthermore, the "Step and Repeat" function can also be used with source files that contain relative values. The number of points parameter is not needed for this particular type of source since each frequency point is specified in the data file.

There are several keywords used in this source file. Their descriptions and functions are:

> **!** - The exclamation mark is a special character used for commenting the file. All text located after this mark is ignored.
>
> **UNITS** - These are the frequency units used in the source file. The acceptable units are HZ - Hertz, KHZ - KiloHertz, and MHZ - MegaHertz
>
> **FREQ** - This is the nominal frequency of the data in the source file. All frequencies in the DATA section will be relative to this value. If this value is set to 0 (default) then all of the frequencies in the DATA block will be single sideband. When the actual source is created both sidebands will be created from the single sideband data.

POWER - This is the nominal power in dBm of the source. All power levels in the DATA section are relative to this value.

PHASE - This is the nominal phase in degrees of the source. All phases in the DATA section are relative to this value.

RANDPHASE - This is a special keyword that will randomize the phase of all data points in the source file irrespective of their values set in the DATA section.

BW - This is the full power bandwidth of the source. The units used for this bandwidth are specified by the UNITS keyword. If this value is set to 0 (default) then all power for the source is considered to be in a 1 Hz bandwidth (this is the definition of a CW signal). When the 'Channel Measurement' bandwidth in SPECTRASYS is set to this bandwidth then a 'Channel Power' measurement on this source signal should give the same power level as specified by this source

DATA - This section is used to specify the actual frequency points of the source along with their amplitudes and phases. This data is entered in three columns: Frequency, Power, and Phase.

ENDDATA - This is a keyword that indicates the end of the DATA block.

NOTE: All frequencies, power levels, and phases in the source file are relative to the source frequency, power, and phase contained on the system source properties dialog box.

The following data is a source file example:

```
! This file is an example of phase noise on a CW carrier
UNITS  HZ                    ! HZ, KHZ, or MHZ

! All final values in this table will be relative to Frequency,

! Power, and Phase specified in the Source dialog box

FREQ          0              ! Nominal Frequency in UNITS

POWER         0              ! Nominal Power in dBm

PHASE         0              ! Nominal Phase in Degrees

!RANDPHASE                   ! Remove initial comment character to randomize phase

BW            0              ! Full Power Bandwidth in UNITS (0 for CW Signals)
```

```
DATA

! All data is Relative to Nominal Values

! If FREQ is 0, data is Single Sideband offset.

! Freq          Power     Phase

!(Units)       (dBc)     (Degrees)

0.51           -30       0

10             -50       0

100            -70       0

1000           -80       0

10000          -90       0

100000         -95       0

1e6            -100      0


ENDDATA
```

NOISE

Broadband noise can be added to any CW, Modulated, or User Defined source. Noise is added to these sources by checking the "Broadband Noise" checkbox. Noise is specified by upper and lower frequency limits and spectral density. The noise spectral density is the power in dBm in a 1 Hz bandwidth. For a noise-only source, the user can un-check the "Include Signal" checkbox.

Source Parameter Tuning

Every source parameter can be tuned by placing a question mark in front of the parameter.

Summary

All sources in SPECTRASYS have bandwidth and spectral density. Sources have a center frequency, bandwidth, power level, phase shift, and number of points.

All sources are defined in the frequency domain.

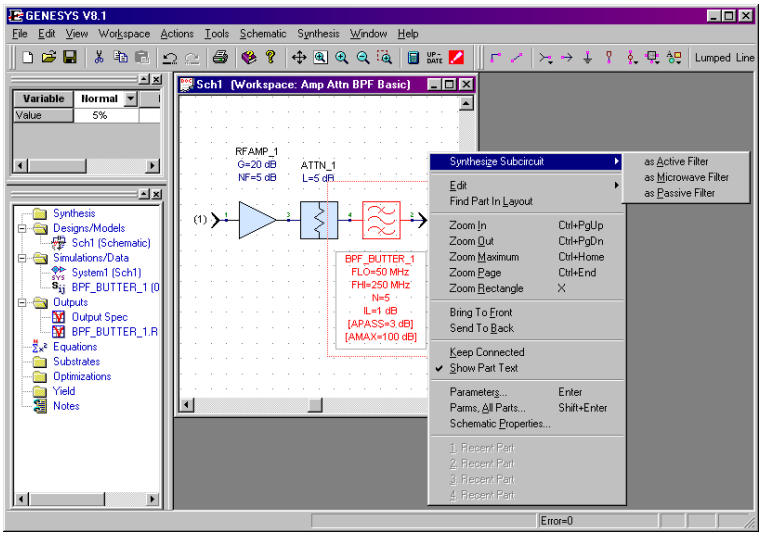
- Sources are Modeled in the Frequency Domain

- Currently, Time Varying Sources are not Supported
- CW Sources are Defined to have 1 Hz Bandwidth
- Modulated Sources can have any Bandwidth
- A Modulated Source is Represented by a Uniform Spectral Density

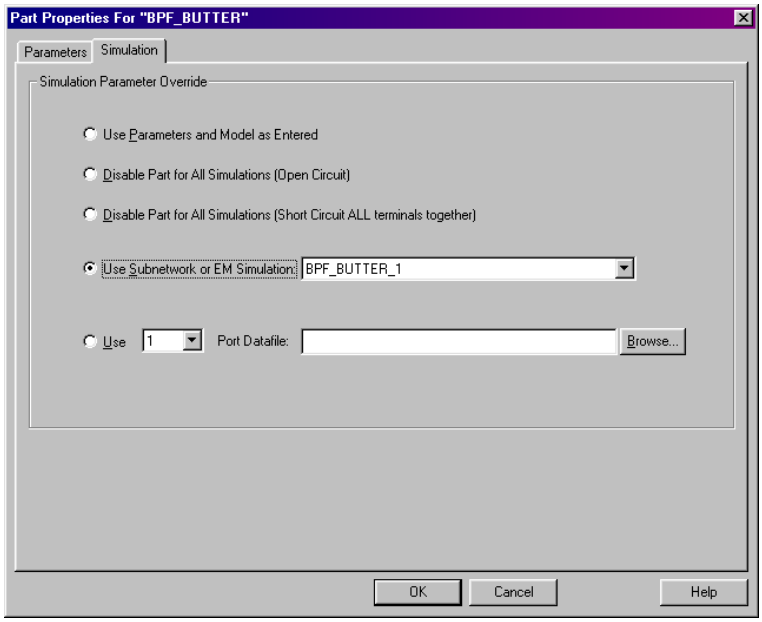
Synthesis

Synthesis

Circuits can directly synthesized from SPECTRASYS. Right clicking on the behavioral model will bring a context sensitive menu. This menu will list the synthesis modules available for the given element. The selected synthesis module will be invoked and the parameters of the behavioral model will be passed to this synthesis module. See the specific synthesis section for more information about each synthesis tool.



Once the user is satisfied with the synthesis results, these results can be substituted back into the behavioral model. If a behavioral model have been directly synthesized the subnetwork substitution will be automatic as show below.



At this point the parameters for the behavioral model will be disabled. For additional information on model substitution see the 'Simulation Tab' section of the Schematic Element Properties page in the User's Guide.

Troubleshooting

How come my noise figure decreases through a cascade?

The equation for Cascaded Noise Figure measurement in SPECTRASYS is:

CNF[n] = CNP[n] - CNP[0] - CGAIN[n] (dB), where n = stage number and CNP is the Channel Noise Power

and the Cascade Gain measurement is:

CGAIN[n] = DCP[n] - DCP[0] (dB), where n = stage number and DCP is the Desired Channel Power

Cascade Gain is therefore a function of all forward traveling power in the channel which is subject to VSWR effects.

-Verify that Gain and Cascaded Gain are as expected

Another issue usually is that the Channel Measurement Bandwidth is much wider than the channel signals. This is ok but extra noise points may need to be added to improve the accuracy of the Channel Noise Power measurement. SPECTRASYS interpolates between all noise and signal data points. If there is a lot of amplitude ripple in the circuit sufficient noise points must be added for each signal to properly account for these variations. If the noise spectrum looks very stick-figure-ish then extra noise points may need to be added.

If cascaded noise figure is being examined through a hybrid combining network the cascaded noise figure will appear to artificially peak at the internal nodes to the hybrid network. This occurs because cascaded gain used is only for the current path and not all parallel paths used in the hybrid network.

Adding extra noise points can be done on the 'Calculate' tab of the 'System Analysis' dialog box.

[SPECTRASYS, Cascaded Noise Figure, NF, CNF, Negative]

Why don't I get the same answer as my spreadsheets?

SPECTRASYS accounts for VSWR between stages, sneak paths, reverse isolation, frequency response, channel bandwidth, gain compression, broadband noise and image noise. Spreadsheets do not! Cascaded noise figure equations assume no image noise and perfect matching between stages. Cascaded intermod equations assume no frequency rolloff for the interfering tones. This can be a bad assumption especially for a receiver blocking test.

In order to correlate SPECTRASYS data with a spreadsheet (or other math packages or programs) the schematic must be reduced to the spreadsheet case. That is:

1. Remove VSWR and frequency effects

a. Behavioral filter have return loss which is a function of the ripple. Set the ripple to something really small like 0.001 dB

b. Set all ports and stages to the same impedance

c. Replace S Parameters elements (or other frequency dependent elements) with attenuators or amplifiers of the equivalent gain.

2. Remove sneak path effects

a. Set isolations very high (100 dB)

b. Set reverse isolation very high (100 dB)

3. Remove gain compression effects

a. Gain compression is based on total node power not channel power. All unwanted signals including noise will contribute to the total node power.

b. Increase the P1dB, PSAT, IP3, and IP2 points of all non-linear stages

4. Remove image noise effects

a. Set the image rejection high (100 dB) in all mixers (be sure to reject the image frequency band not the desired channel band)

After making these changes you will get excellent correlation.

[Spectrasys, Gain, Cascaded Gain, Cascaded Noise Figure, Intermods, GAIN, CGAIN, CNF, NF, IM3]

System Simulation Tips

The basic operation of SPECTRASYS involves the propagation of individual source spectra and all of their derived products (intermods, harmonics, etc.) to every node in the system. These spectrums will keep propagating until no additional spectrums are created. For instance, any new inputs arriving at the input of an amplifier will cause intermods and harmonics to be created at the amplifier output at that particular time. If additional signals arrive at the amplifier input at a future time then new intermods, harmonics, and other spurious products will be created at the amplifier output. This process continues until no additional spectrums are created. If loops exist in the system schematic, then the output from one element will feed the input of the next element and spectrum propagation could continue forever unless special features are placed within the software to limit spectral creation in this infinite loop. SPECTRASYS has special features to control loops and limit the total number of created spectrums.

Loops

Elements in parallel (parallel amplifiers connected via a 2 way splitter at the input and combined back together with a 2 way combiner at the output) can cause spectrums to be created that will propagate around this parallel path (or loop). If the gain of the amplifier is greater than its reverse isolation the spectrums will keep on growing as they travel around the path and will never die out (we would have an oscillator). The key point here is that if there are loops in the system schematic then it is very important to make sure that the element parameters are entered correctly so that signals don't grow in amplitude as they traverse around a loop. The simulation will only be as good as those parameters in the model. If the user is suspicious that the simulation is taking extra time then isolation parameters of the components that make up that loop can be increased to large values to see if that is the cause. The user can then start decreasing the isolation of the interested components until the desired response is achieved.

Maximum Number of Spectrums to Generate

As a last resort, you can limit the number of spectrums that will be generated. The number of spectrums generated at any time is shown in the simulation status window while SPECTRASYS is running. A typical number to force a limit to is 100000. See the Options Tab for more information.

Ignore Spectrum

As mentioned previously, SPECTRASYS will continue to process new spectrums until no additional spectrums have been calculated. However, in the case where a loop exists, spectrums will continue to be created around the loop until the 'Ignore Spectrum Below' threshold is reached at which time spectrums are not calculated below this threshold. The higher this threshold the fewer the number of calculated spectrums. In order to minimize the simulation time the user should set this threshold to calculate the least number of spectrums which will accurately represent the output. For example, if the user is not interested in seeing anything below -100 dBm and simulation speed is an issue, then setting the threshold to -100 dBm will improve the simulation speed.

Linear Elements

The more nodes in the system schematic, the more spectrums that will be created and propagated. This spectrum creation and propagation takes time. If several linear elements are used in the system schematic and simulation speed is an issue, then linear element circuits can be moved to new schematics and then linked into the system schematic using a 'Network' block. This 'Network' block will then point to this newly created schematics.

Intermods

One of the largest time consuming operations in SPECTRASYS is the calculation of a large number of intermods due to a large number of input signals into a non-linear device such as a mixer or amplifier. Various options can be selected to improve the simulation speed. Fasted intermod and harmonic simulation occurs when 'From Sources Only', ' Odd Orders Only', and 'Fast Intermod Shape' are enabled. Limiting the maximum order to the order of interest will eliminate time wasted in calculating irrelevant intermods and harmonics. You can also disable the calculation of intermods and harmonics until the initial architecture and basic budget parameters are set. These settings can be changed on the 'Calculate' tab of the 'System Simulation' dialog box.

If a large number of intermods are to be calculated (due to a large number of input signals) the best thing to do is to first establish the architecture and make sure that system is performing as expected for a small number of input signals. It is much faster to optimize the architecture with a small number of input signals, rather than wasting time waiting for complete system analysis for issues that can be resolved with far fewer input signals.

There are also different intermod and harmonic calculation modes that can increase the simulation speed. See the 'Calculate Intermods and Harmonics' section for additional information of these modes.

Analyzer Mode

During the system simulation the analyzer will create an analyzer trace for every node in the system. Consequently, for systems with large number of nodes the integrated analyzer traces alone can be time consuming if the analyzer properties are not optimized.

The simulation speed can be reduced by a careful selection of "Analyzer Mode" settings. If large frequency ranges are integrated with a small resolution bandwidth then the amount of data collected will be much larger and the simulation speed will decrease. Furthermore, enabling the 'Randomize Noise' feature may also slow down the simulation. In order to increase the simulation speed with the 'Analyzer Mode' enabled the user can disable the 'Randomize Noise' feature, increase the 'Resolution Bandwidth', and/or limit the frequency range over which a spectrum analyzer trace will be created. See the 'Analyzer Mode' section for additional information.

Ports

The standard input (*INP) and standard output (*OUT) ports are much more flexible than other ports used in GENESYS. Sources can be created and managed through the system simulation dialog box. Sources can be applied to both input and output ports. Functions like 'Step and Repeat', 'Added Noise', and 'Phase Noise' are not available except through the system simulation dialog box.

Noise

Noise points can be added in such a way to see filter shapes in spectrum plots. Potential noise interpolation problems can be eliminated when enough noise points have been added at the correct places so that filtered noise spectrum represents the shape of the filter. However, the more noise points that are simulated, generally the longer it takes the simulation to run.

See the 'Calculate' tab for more information on inserting noise points and noise calculations.

Mixers

The more Low Signals used to create new mixed spectrum (when 'All Signals Within x dBC of strongest' is selected) the longer simulation time will result since more spectrums are being calculated.

Chapter 8 Parameter Sweeps

Parameter Sweep

3D graphs in GENESYS require parameter sweeps to generate a third dimension for plotting. Parameter Sweeps give you this third dimension by adjusting a tuned variable, repeating another simulation for each adjustment. For example, to see how the response of a circuit changes when a capacitor is adjusted, you can add a Parameter sweep which sweeps the linear or electromagnetic simulation while adjusting the capacitor value. You can then view the results on a 3-D graph.

To add a parameter sweep:

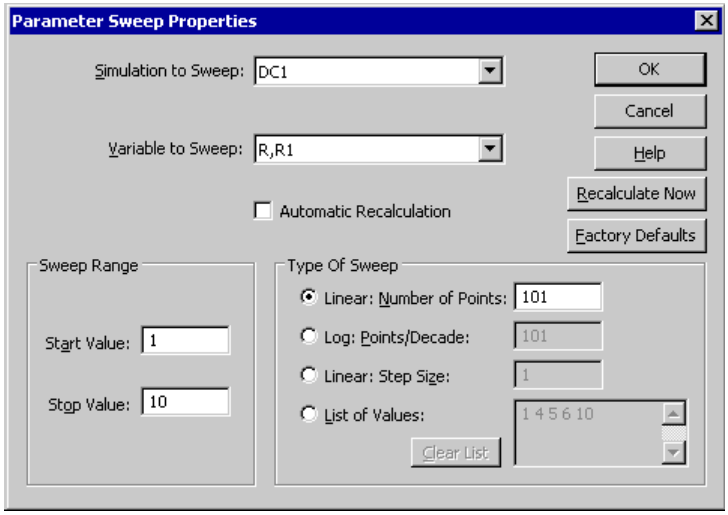
1. Right-click the Simulation/Data node on the Workspace Window.
2. Select "Add Parameter Sweep".
3. Add a Table or 3-D graph to display the results.

For advanced applications, you can nest Parameter sweeps, creating 4-D, 5-D, or higher data. This data can then be viewed on a table.

For information on the Parameter Sweep Properties dialog, see the Reference manual.

Parameter Sweep Properties

To open: double-click or create a Parameter Sweep.



Simulation to Sweep - Chooses which simulation to use for the parameter sweep. The selected simulation will be recalculated for each different value of the variable chosen below.

Variable to Sweep - Specifies which variable gets changed to create the sweep. All variables which appear in the tune window (marked with '?') are available to be swept.

Automatic Recalculation - Checking this box will cause the harmonic balance simulation to be run any time there is a change in the design. If the box is not checked, the simulation must be run manually either by right clicking on the simulation icon and selecting "Recalculate Now" or by clicking the recalculation button on the main tool bar.

Recalculate Now - Dismisses the dialog box and starts the simulator if required. If the circuit has already been simulated and has not been changed, the simulator will not calculate again.

Factory Defaults - Resets the sweep properties to the original, known-good settings.

Sweep Range

- **Start Value** - The lower bound (minimum frequency) of the sweep.
- **Stop Value** - The upper bound (maximum frequency) of the sweep.

Type of Sweep

- **Linear: Number of Points** - Allows specification of start value, stop value, and number of points
- **Log: Points/Decade** - Allows specification of start value, stop value, and number of points
- **Linear: Step Size** - Allows specification of start value, stop value, and space between points.
- **List of Values** - Allows the explicit specification of variable values. These points are entered into the List of Points box separated by spaces.

Chapter 9 Equation Reference

Statements

Each line in the EQUATION window must be in one of 5 formats: assignment, REF, comment, IF, THEN, GOTO, FUNCTION, RETURN, or BASE. The formats are described below.

Assignment

The assignment line assigns a value to a variable. The assignment statement calculates the value of the expression and then gives the value to the specified variable. Variables are not case sensitive (for example, VAR and var are the same). Accuracy is IEEE double precision (about twelve digits). The format is:

Variablename = *Expression*

Examples:

```
X=2
R=4*3/2^4*(9+8)
Voltage=(2+R)*Current
```

Assignments can define a value to be a variable, which allows that variable to be tuned, optimized, or included in the Monte Carlo analysis. All variable names must start with an alpha character. The rest of the name may contain letters, numbers and the underscore ("_") character. The tune statement format is:

VariableName = ?*Value*

Examples

```
X=?2
Large_R=?3.54e+16
```

The tune statement must be a single assignment, not an expression. Therefore, the following statement is illegal

```
X=?2+2 (WRONG!)
```

REF

This statement creates a reference to an expression. *expression* must be a simple variable, array element, or post-processed data. This can make your equations faster and easier to write. The format of the reference statement is:

REF *Variable* = *expression*

Example:

```
B=5
REF A=B 'A now points to B
```

```
C=A+A 'C now equals 10
A=C 'B (and, indirectly, A) now equals 10
D=VECTOR(20)
REF A=D[C] 'A points to D[10]
A=3.14 'D[10] now equals 3.14.
```

Comment

A line is considered a comment if the first character in the line is an apostrophe ('). Any part of a line can be a comment and everything after the apostrophe is ignored. The comment line format is:

'*Comment*

Example

'This line will be ignored.

LABEL

The label statement identifies a section of the EQUATION window for use in GOTO or IF THEN GOTO statements. After the GOTO is executed, the statement following LABEL is the next statement executed. If LABEL is the last statement in the window, the equations end after the GOTO. The format is:

LABEL *Labelname*

GOTO

This statement causes the EQUATION interpreter to jump in its calculations to the statement following the corresponding LABEL statement. The format of the GOTO statement is:

GOTO *Labelname*

IF

This statement is perhaps the most powerful one included in GENESYS. This statement causes the following steps to occur.

1. The value of the expression is calculated. Any true comparison results in a value of -1. For example, the expression 1>0 gives a value of -1, while the expression 0>1 gives a value of zero.
2. The value obtained in step one is compared to zero. If the value is not zero, then the interpreter performs the statement specified.

The format of the IF statement is:

IF *expression* THEN *statement*

Example:

```
IF Q>1000 THEN GOTO HIGHQ
RVal = 100
GOTO DONE
LABEL HIGHQ
RVal = 500
LABEL DONE
```

Warning: You cannot use IF/THEN with post-processed variables. Use the IFF and IFTRUE functions instead.

Since GENESYS uses approximate calculations (as any computer program must), round-off errors are inevitable. This could cause a problem if you are using equality checks. If this is the case, change

```
IF value = 5 THEN GOTO LABEL
```

to

```
IF ABS(value-5)<0.00001 THEN GOTO LABEL
```

or something similar. If you are using relational operators such as greater than (>) or less than (<), this point does not need to be considered.

FUNCTION

This statement is used to define functions. Functions take zero or more parameters as input and return exactly one value as output. All variables used within a function are local; that is, variables cannot be shared across functions or with the main block. See User Functions for detailed information on this statement. The format of a FUNCTION statement is:

```
FUNCTION name(parm1,parm2...)
equations
RETURN expression
```

An example function to calculate the inductance that resonates with a capacitor at a given frequency:

```
FUNCTION RESL(C,F)
'L is in nH, C is in pF, F is in MHz
FHz=1e6*F
CFarads=1e-12*C
Omega=2*PI*FHz
LHenries=1/(Omega*Omega*CFarads)
Return LHenries*1e9
```

RETURN

This statement returns a value from a function and exits the function. Note that this statement does not mark the end the function declaration, and a function with IF/THEN

statements can have more than one RETURN statement. The format of the return statement is:

RETURN *expression*

BASE

This statement defines the beginning index of arrays. The default base is 1, meaning that the first data point in an array is accessed using the number 1. The statement can appear more than once in an EQUATION window. A new base statement changes the beginning index of all arrays, whether they were defined before or after the base statement. The form of the statement is:

BASE 0

or

BASE 1

Viewing Variable Values

Values calculated in the EQUATION Window may be viewed to verify that the equations yield expected results. Right-click on Data Outputs in the Workspace Window and select "Add Variable Viewer".

Operators

Operator descriptions in precedence order are:

Operator	Meaning	Comments
[]	Array Index	
^	Exponentiation	Raises a number to a power. For example, 2^3 is 8, and 3^2 is 9
*	Multiplication	
/	Division	
\	Integer Division	The quotient is truncated to an integer result. For example, 10\3 is 3 and 3\4 is zero.
%	Modulo	The numbers are divided, and the remainder is returned. For example, 10%3 is 1 and 7.6%2 is 1.6.
+	Addition	
-	Subtraction	
=	Equality Check	Left and right values are compared. If the results are equal, the value is 1 (true); otherwise, the value is zero (false). For example, 1+1=2 gives 1 and 1+1=3 gives zero.
>	Greater Than	
<	Less Than	
>=, =>	Greater Than or Equal	
<=, =<	Less Than or	

	Equal	
!	Not	
&	And	
|	Or	This symbol is also referred to as "pipes". It is normally located on the back-slash (\) key using Shift.
;	Array Concatenation	Concatenates values to form vectors an matrices. See Arrays in this section.

Sample Expressions

Expression	Value
1+2*3	7
(1+2)*3	9
4^3	64
3*4^3	192
19/4	4.75
19\4	4
19$4	3
1+19%2*2^2	5
5>4	1 (True)
5<4	0 (False)
2*4>1+3 & 4*4<17^2	1 (True)
2*4>1+3 @ 4*4<17^2	0 (False)
SIN(180)<.5	1 (True)

Built-in Functions

Caution: Standard trigonometric functions must have an argument in degrees, and inverse standard functions return values in degrees. Hyperbolic trigonometric functions use pure numbers (not degrees).

ABS(expression) - absolute value of expression. For complex values, returns magnitude. Alternate form: MAG(expression)

ANG(expression) - phase of a complex number, returns between -180 and 180 degrees.

ANG360(expression) - phase of a complex number, returns between 0 and 360 degrees.

ARCCOS(expression) - inverse cosine ($\cos^{-1}$). Range: Argument must be between -1 and +1.

ARCCOSH(expression) - inverse hyperbolic cosine

ARCSIN(expression) - inverse sine ($\sin^{-1}$). Range: Argument must be between -1 and +1.

ARCSINH(expression) - inverse hyperbolic sine

ARCTAN(expression) - inverse tangent ($\tan^{-1}$). Alternate form: ATN(expression)

ARCTANH(expression) - inverse hyperbolic tangent

BESSELJ0(expression) - Calculates Bessel function J0 of expression.

COMPLEX(real,imag) - returns a complex number real + j imag

CONTOUR(expression, minContour, maxContour, stepSize, smoothParm, minX, maxX, minY, maxY, primaryGridSize, secondaryGridSize) - Generates contour plot of 2D data (expression) on a Smith Chart. Expression is a mandatory parameter, the remainder are optional parameters. Expression must contain coordinates (real, imag) and values for a given measurement, and a Thin Plate Spline is generated from which the contours are generated. The smoothParm is generally a number between 0 (no smoothing) and 1 (very strong smoothing). minX, maxX, minY, and maxY control the domain for which data is generated. primaryGridSize and secondaryGridSize control the resolution of the contour generation (primary should be a smaller number than secondary).

COS(expression) - cosine

COSH(expression) - hyperbolic cosine

COUNT(expression) - returns the number of data points contained in post-processed data, or the size of an array. See Arrays or Post Processing later in the equations reference.

DB10(expression) - returns 10*log(|expression|)

DB20(expression) - returns 20*log(|expression|)

EXP(expression) - value of "e" raised to expression

FIX(expression) - truncates the expression. Examples: FIX(5.6) is 5 and FIX(-1.4) is -1

FN_E(expression) - Calculates the complete elliptic integral of the second kind.

FN_K(expression) - Calculates the complete elliptic integral of the first kind.

GET(string) - Gets a measurement from a string variable. Can be useful for constructing a measurement from pieces of text. See Post Processing later in the equations reference.

GETINDEPVALUE(expression,index,dim) - returns the independent data point for dimension dim of a post-processed expression. See Post Processing later in the equations reference.

Note: If the independent data is frequency, GETINDEPVALUE returns the values in Hz (not MHz).

GETVALUE(expression,index) - calculates and returns a value of a post-processed expression. See Post Processing later in the equations reference.

GETVALUEAT(expression,indep) - calculates and returns a value of a post-processed expression at a given independent value. Only works on 2 dimensional data (X vs.Y). See Post Processing later in the equations reference.

> **Note:** If the independent data is frequency, GETINDEPAT requires values in Hz (not MHz).

IFF(condition,trueValue,falseValue) - returns trueValue if condition is true, and falseValue if condition is false. Can be used with any data, including post-processed data.

IFTRUE(condition,trueValue) - returns trueValue if condition is true, and zero if condition is false. Can be used with any data, including post-processed data.

IMAG(expression) - returns the imaginary part of a complex number. Alternate form: IM(expression)

INT(expression) - greatest integer less than or equal to the expression Examples: INT(5.6) is 5 and INT(-1.4) is -2

LOG(expression) - base 10 logarithm

LN(expression) - natural logarithm

MATRIX(rows, columns) - Returns a 2-dimensional array of size rows x columns. See Arrays later in the equations reference.

MIN(expression) - Finds the minimum value of a post-processed expression.

MAX(expression) - Finds the maximum value of a post-processed expression.

PLOTPOINTS(expression) - Plots 2D input data points on a Smith Chart without connecting the dots. Useful for seeing what data points have defined measurement values in a load-pull file.

REAL(expression) - returns the real part of a complex number. Alternate form: RE(expression)

RND - returns a pseudo-random number between zero and one

SIN(expression) - sine of the argument

SINH(expression) - hyperbolic sine

SQR(expression) - square root

TAN(expression) - tangent. Range: Argument must not be ±90, ±3*90, etc.

TANH(expression) - hyperbolic tangent. Range: Same as TAN(expression)

VALUEAT(expression, real, imag, smoothParm) - Returns the value of the thin plate spline generated from 2D data in expression, at a given coordinate (real, imag). The smoothParm parameter is optional (see function CONTOUR).

VECTOR(expression) - returns a vector (array) of size expression. See Arrays later in the equations reference.

Constants

Name	Value
PI	p, 3.14159265
_EPS0	8.854e-12
_ETA0	376.7343
_MU0	1.256637e-6
_VAIR	c, 2.997925e8
_LN2	ln(2), 0.6931471805599
_EXP1	e, 2.718281828459
_RTOD	Radians to degrees multiplier, 180/pi
_DTOR	Degrees to radians multiplier, pi/180

Strings

String variables can be used in the Equation Window:

```
A="ABC"
B="DEF"
C=A+B
```

After this code, C="ABCDEF". Concatenation (+) is the only operator currently defined for string variables, all other operations give undefined results.

If you create a model and want it to take a string variable as a parameter, put a ~ (tilde) in front of the parameter name in the Model Properties dialog box to indicate that it is a string. Furthermore, if the parameter starts with the word or is the word "~FILENAME", a browse button will be given to the user in the schematic part dialog box.

Arrays (Vectors and Matrices)

GENESYS allows you to create vectors and matrices in the Equation Window. Each element in a vector or matrix can hold any type of data: real, complex, string, swept, or even a nested array. Equations are made most easily with the array concatenation operator ; (semi-colon). For example:

```
X = 3;4;5
Y=X[1]+X[3]    'Y contains 8 (3+5)
```

places an array of values (3, 4, and 5) into variable X, and uses these values in Y. To make a two dimensional array, use parenthesis:

```
X = (1;2;3) ; (4;5;6) ; (7;8;9)
Y=X[2,2]    'Y contains 5
```

The ; operator can be combined with other operators and complex values:

```
W = 1;2;3
X = 8;sqr(4)+3;5
Y=COMPLEX(W,X)    'Y array ((1+j8), (2+j5), (3+j5))
```

There are also two functions which you can use to create arrays in your equations:

VECTOR(x) - returns a vector (1-dimensional array) of x real zeros. Elements are accessed using square brackets and are base one (numbering starts at one) by default:

```
A = VECTOR(3)
A[1] = 1
A[2] = 5
A[3] = A[1] + A[2] 'A[3] now contains 6.
```

MATRIX(x,y) - returns a matrix (2-dimensional array) of x by y real zeros. Elements are accessed using square brackets and are base one (numbering starts at one) by default:

```
B = MATRIX(2,2)
B[1,1] = complex(1,3)
B[1,2] = 3
B[2,1] = 3
B[2,2] = complex(1,-3)
```

Note for advanced users: Elements can also be accessed linearly in row-column order, which can be useful in some situations. Thus, the following equations work identically to the equations just given above:

```
B = MATRIX(2,2)
B[1] = complex(1,3)
B[2] = 3
B[3] = 3
B[4] = complex(1,-3)
```

GENESYS currently contains no special matrix mathematical operators. All operations simply work on each element individually. For example:

```
C=VECTOR(2)
C[1]=5
C[2]=-3.4
D=VECTOR(2)
D[1]=C[2]+3 'D[1] now equals -0.4
D[2]=COMPLEX(5,6)
E=C+D
```

E is now a two-element vector, E[1]=4.6, E[2]=-1.6 + j6.

Scalar/matrix combination operators also work. For instance, adding a complex number to a vector adds the complex number to every element of the vector:

```
F=VECTOR(2)
F[1]=1
F[2]=2
```

```
G=COMPLEX(3,4)
H=F+G 'H[1]=4+j4, H[2]=5+j4
```

Matrices and vectors are safe; out of bounds access is always caught. If an out-of-bounds index is used, the first element is used instead. If the variable being indexed is not an array, its value is used instead.

If two matrices of different sizes are added, then the operation is only performed up to the size of the smallest matrix. These operations are performed as if the matrices were vectors; see the example of linearly accessing a matrix as a vector above.

All operators and builtin functions will work properly on arrays, so, for example, taking the hyperbolic sine of matrix A using SINH(A) will take the hyperbolic sine of each element of A. Also, arrays can be passed to user models and functions, so you can create a user model that takes a matrix or vector as a parameter.

Strings can be used in vectors, and the addition operator will work. For example:

```
J=VECTOR(3)
J[1]="One"
J[2]="Two"
J[3]="Three"
K="Element "
M=J+K 'M[1]="Element One", M[2]="Element Two", etc.
```

Note: Vectors and matrices are now base one in GENESYS (first element is number one). To use base zero, put the statement "BASE 0" on a blank line at the top of your equations and at the top of any function.

To find out how many elements an array has, use the COUNT function:

```
N=VECTOR(71)
P=COUNT(N) 'P=71
Q=MATRIX(100,75)
R=COUNT(Q) 'R=7500
```

Post Processing

One of the more powerful features of the GENESYS Equation Window is post-processing (sometimes referred to as Output Equations). This allows you to perform calculations on the results of your analysis. These results can then be displayed, optimized, or even used in another design. For example:

```
Gain=Linear1.Filter.DB[S21]
AddToGain = ?5
TotalGain = Gain + AddToGain
```

This example takes the gain in dB, DB[S21], of the design "Filter" using the simulation setup in "Linear1" and places the result into the variable "Gain". For a complete explanation of this syntax, see the Measurements section of this manual. Note that Gain now contains swept data, DB[S21] vs. frequency. Next, the variable AddToGain is added

to each data point. The variable AddToGain can be tuned or optimized, which will directly affect the value of TotalGain.

There are several important things to know about post-processed data:

- Any measurement described in the measurements section of this manual is available for use in post-processing.
- To get simulation data, the expression must contain a period. For example A=DB[S21] will not work, but A=.DB[S21] will. This is most important if you take advantage of the USING statement (see below).
- Post-processed equations can be used directly in a graph or other output by putting = in front of the measurement. All rules of this section, including the period rule above, must be followed.
- To get simulation data, you must always use a measurement operator. For example, A=Linear1.Filter.S21 will not work, but A=Linear1.Filter.DB[S21] will.
- Post-Processed variables can be mixed with "regular" variables as in the example above.
- Frequency-dependent post-processed variables can be used in part values. The data will be sampled/interpolated/extracted as necessary, and the resulting part value may be different at each frequency. For example, R=50+.1*FREQ can be used to create a frequency-dependent resistor.
- Post-Processed variables can be combined. For example, the statement "Difference=Linear1.Filter.DB[S21]-Measured.Data.DB[S21]" gives the difference between the measured and the calculated DB[S21].
- For any operator or built-in function, swept data will be linearly interpolated if needed, and the resulting sweep will contain all frequency points from both the measured and the calculated data. In the item above, the difference variable will contain all data points from both the linear analysis and the measured data.
- All operators and built-in functions will work on post-processed data. For example, the statement "SineS=SIN(Linear1.FILTER.ANG[S21])" will take the sine of the phase of S21 at each data point.
- If the simulation data is itself a matrix, everything will still work fine. For example, the statement "Difference=Linear1.Filter.RECT[S]-Measured.Data.RECT[S]" will take the difference of all s-parameters. The Difference variable will now behave like an array (see the previous section), with the addition that all operations will operate at all frequencies. For example, Difference[2,1] returns the difference of S21 at all frequencies.
- FREQ is a post-processed variable. For each frequency point, the value is that frequency. All frequencies are in MHz. Exception: In a user model, if the freq variable is used, the model is calculated once per frequency, and FREQ is just a normal number.

- Post-processed variables cannot be used in IF-THEN statements. For example, "IF .DB[S21]>5 THEN Gain=Gain+10" is not legal. Instead, you should use the IFF and IFTRUE functions. In this case, you can state "Gain=Gain+IFTRUE(.DB[S21]>5, 10)". This is because the equations are only calculated once (not at each frequency).
- *All calculations are deferred until requested.* This means that when any of the statements shown above are encountered, the required calculation is simply noted. Later, when the data is needed, the calculation is performed. What does this mean to most users? Simply that post-processed calculations are very fast, do not require a lot of memory overhead, and only calculate when necessary.
- The USING statement is a big convenience if you are writing many expressions. With it, you only need to specify the simulation/data and design once. The USING statement applies for all measurements specified after it, and it does not carry over into functions. For example:

```
USING Linear1.FILTER
Gain=.DB[S21]
InputReflection=.DB[S11]
OutputReflection=.DB[S22]
Delay=.GD[S21]
```

Note: You must specify the period before the measurement. This tells GENESYS that you are getting post-processed data. If you leave out the period, you will get errors like "Unknown Variable S11".

Several functions in GENESYS are for use with post-processed calculations:

Note: These are advanced functions which are not required my most users. If you are not sure if you need to use them, then you probably don't.

COUNT(expression) - For post-processed data, this function will return the number of data points in the swept data. For example, if Linear1 is a linear simulation with 101 frequency points, then COUNT(Linear1.Sch.DB[S21]) is 101. This function is most useful if you want to loop post-processed data points with IF/THEN/GOTO Statements.

GET(string) - Gets a measurement from a string variable. The statements A=.DB[S21] and A=GET("DB[S21]") are identical. This statement exists so that you can pass a string containing the name of a measurement into a function, allowing the function to get the data.

GETINDEPVALUE(expression,index,dim) - returns the independent data point for dimension dim of a post-processed expression. Expression is the post-processed data, index is the point number, and dim is the independent dimension number to use. For normal frequency sweeps, dim should be 1. For parameter sweeps with multiple independent sweeps, you must use dim to specify whether you want to get frequency (dim=1) or the parameter (dim=2, or higher for nested parameter sweeps).

Note: If the independent data is frequency, GETINDEPVALUE returns the values in Hz (not MHz).

GETVALUE(expression,index) - calculates and returns a value of a post-processed expression. This allows you to get the value of an expression at a particular data point (index). This function is most useful in combination with the COUNT function for looping over values. Most users should not use this function, preferring the GETVALUEAT function instead. Note that this function causes immediate calculation of the value, and the value it returns is not swept; it is the actual value of a particular data point (real or imaginary). Advanced note: If the independent data is multi-dimensional, then index can contain an array specifying the index for each dimension.

GETVALUEAT(expression,indep) - calculates and returns a value of a post-processed expression at a given independent value. For example, this allows you to get the value of an expression at a particular frequency, such as Q=GETVALUEAT(.QL[S21],1e9) which gets the loaded Q of S21 at 1 GHz. If no data has been calculated at 1 GHz, the data will be interpolated or extrapolated as needed. While this function is somewhat slower than GETVALUE, it is much easier to use because you do not have to know the index of the point you want. Note that this function causes immediate calculation of the value, and the value it returns is not swept; it is the actual value of a particular data point (real or imaginary). Advanced note: This function only works on 2 dimensional data (X vs.Y).

Note: If the independent data is frequency, GETVALUEAT requires values in Hz (not MHz).

Equation Wizard

The equation wizard is a valuable tool that can help the user create the proper syntax for post-processed equations. The equation wizard can be accessed in one of two locations: 1) from the graph properties dialog box and 2) from the main menu under the "Tools" submenu.

Note: The equation window must be the active window before the "Equation Measurement Wizard..." menu selection will become active.

Graphing an Equation

The equation wizard can be used to create an equation that can be plotted on graph. See graph properties and the Equation Wizard dialog box for more information on graphing equations.

Equations in the Equations Section

While typing equations in the Equations Section of the workspace the Equation Measurement Wizard can be accessed from the "Tools" submenu located in the main menu. Selecting "Equation Measurement Wizard..." will bring up the Measurement Wizard allowing the user to select the workspace and the desired simulation. Another

Measurement Wizard dialog box will then appear where the user will be able to select the desired function and operator.

Note: Equations will be inserted into the Equations Section at the current location of the cursor.

Logical Operators

The NOT, AND, OR operators are called logical operators. They can be used to combine relational tests, such as "A<5 & B>6". They can also be used in binary math as described below.

Note: The information below is for advanced users and assumes that you are familiar with basic concepts of binary arithmetic and logical operators.

Whenever a logical operation (such as & or |) is performed, the values used are first converted to 32-bit signed integers (truncated). The operation is performed, and then the numbers are converted back to floating point format. This causes logical operators to work as expected when combined with relational operators: true is given a value of -1, which corresponds to all ones in binary notation; false is 0, which corresponds to all zeroes. So, when a logical operation is performed after a relational test, the value is either -1 (true) or 0 (false). This is the rationale for having the IF THEN GOTO Statement branch on a nonzero value. Relational operators act as expected on binary numbers, although there are no facilities included for conversion between binary and decimal format. So, the value of 5&4 is 4 and the value of 128|64 is 192.. The not operator (!) changes each 0 in the binary representation to a 1, and changes each 1 to a 0. Here are logical operator truth tables:

A	**B**	**!A**	**A&B**	**A|B**
0	0	1	0	0
0	1	1	0	1
1	0	0	0	1
1	1	0	1	1

User Functions

Functions can be created in GENESYS. Their format is:

FUNCTION *name*(*parm1*,*parm2*...)
equations
RETURN *expression*

Functions take zero or more parameters as input and return exactly one value as output. All variables used within a function are local; that is, variables cannot be shared across functions or with the main Equation Window. An example function to calculate the inductance that resonates with a capacitor at a given frequency:

```
FUNCTION RESL(C,F)
'L is in nH, C is in pF, F is in MHz
FHz=1e6*F
CFarads=1e-12*C
Omega=2*PI*FHz
LHenries=1/(Omega*Omega*CFarads)
Return LHenries*1e9
```

An example which uses this function is:

```
L=RESL(100,50) 'Find L to resonate 100pF at 50 MHz.
```

You could also type RESL(100,50) into a part value in SCHEMAX.

Functions should go at the end of the global equations in your workspace. If you have functions you want to save permanently, save your workspace in the \EAGLE\MODEL directory. (Multiple functions can be placed in one file.) The functions will then be automatically loaded when GENESYS is started.

For advanced uses, you can pass variables by reference, which means that the function can directly modify the variables passed in. To pass a variable by reference, put the word BYREF in front of the name. For example:

```
FUNCTION DOUBLE(BYREF X,BYREF Y)
X=X*2
Y=Y*2
RETURN 0
```

Calling this function doubles the variables passed in. For example:

```
A=5
B=6
IGNORE=DOUBLE(A,B) 'After this call, A=10 and B=12.
```

Notice that all functions must return a value, even if you will ignore it as in this case.

Calling Your FORTRAN/C/C++ DLLs

GENESYS has the capability to call programs you have written. The techniques for doing this are beyond the scope of this manual. If you are interested in this capability, contact Eagleware, and we will be happy to provide you with an application note with instructions.

Equations Overview

This chapter gives complete reference for user equations. For basic operation of equations, see the GENESYS Environment/Equations section of the User's Guide.

Chapter 10 Device Data

Linear vs. Nonlinear Device Models

S-parameters for RF and microwave devices are commonly available and easy to measure with a network analyzer. They are the most accurate way to model the small-signal performance of circuits. However, they are only valid at a particular operating point (bias level). Nonlinear device models are also commonly available from manufacturers but they are harder to extract from measurements. The advantage of nonlinear models is that they model circuit performance at all bias levels and frequencies. Moreover, the model characterizes the complete linear and nonlinear performance of the devices, including effects such as compression and distortion.

Linear Data Overview

Within GENESYS are a wide range of element models. Also, the model and equation features provide for user creation of models. However, it is often necessary or desirable to characterize a device used in GENESYS by measured or externally computed data. This function is provided for by the use of the ONE, TWO, THR, FOU, and NPO elements which read S, Y, G, H, or Z-parameter data.

Note: The information provided in this section applies to linear devices.

When using linear simulator, circuits are assumed time-invariant (element values are not a function of time) and thus sub-components are uniquely defined by a set of port parameter sets, such as two-port S-parameter data.

Although ONE, TWO, THR, FOU, and NPO are typically used for active devices, they may be used for any devices for which you can compute or measure data. For example, they could be used to characterize an antenna, a circuit with specified group delay data, or measured data for a broadband transformer or a pad.

Using a Data File in GENESYS

Data files can be used in GENESYS in two different ways:

- By adding a Link to a Data File in a simulation. This allows measurements to refer directly to the data file without the need to create a design.
- By using ONE, TWO, THR, FOU, or NPO elements in a circuit file or schematic.

In both cases, you must know in advance how many ports the device data represents. For transistors, this is almost always 2.

Link To Data File

A "Link to Data file" allows you to plot data from a device data file without drawing a schematic or creating a netlist.

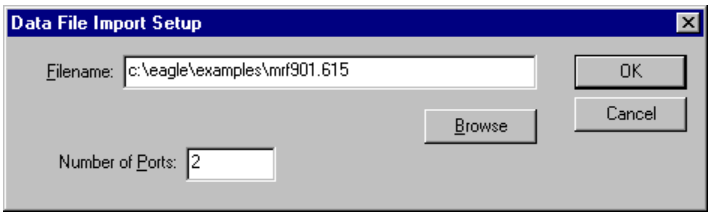
To add a data file import:

1. Right-click the Simulation/Data node on the Workspace Window.
2. Select "Add Link To Data File".

For information on the Data File Import Setup dialog, see the Reference manual. For an example, see "Model Extract.WSP".

Link to Data File Setup

To open: double-click or create a Link to a Data File.



Filename - Specifies the file containing the Device Data to load.

Browse - Opens a File Open Dialog box so that you can locate the desired data file.

Number of ports - Specifies the number of ports the data file has.

Provided Device Data

GENESYS includes over 25,000 data files for many different device types. Device data was provided directly by the manufacturers in electronic format.

Caution: Eagleware could not test every file that was provided. Through random sampling, we edited errors found in some files. It is the user's responsibility to test each file for accuracy.

Creating New Linear Data Files

You may easily add other devices to the library by using a text editor (such as NOTEPAD) to type the data into a file with the name of your choice. Be sure to save the file in standard ASCII format.

The first line in the file after any initial comments is a format specifier in the form:

units type format R *impedance*

where:

units is either Hz, kHz, MHz, or GHz
type is the type of the data file, either S, Y, G, H, or Z
format is DB for dB/angle data, MA for linear magnitude/angle data, or RI for real/imaginary data
impedance is the reference impedance in ohms, commonly 50 or 75

One of the most common format specifiers is:

MHZ S MA R 50

This indicates that the data is in S parameter form normalized to 50 ohms. The data is given in linear polar format (magnitude & angle). The frequencies are in megahertz.

The data follows after the format specifier. A typical line for this two-port file is:

500 .64 -23 12.5 98 .03 70 .8 -37

In this case, 500 is the frequency in megahertz. The magnitudes of S11, S21, S12 and S22 are .64, 12.5, .03 and .8, respectively. The phases are -23, 98, 70 and -37 degrees, respectively.

Alternatively, Y-parameter data may be used. The format specifier could be:

GHZ Y RI R 1

This would indicate rectangular, unnormalized Y parameter data with frequencies in GHz. A typical line is:

30 0 3E-4 9E-3 -8E-3 2E-5 0 -1E-4 1E-3

In this case, the frequency in gigahertz is 30. The real values of Y11, Y21, Y12 and Y22 are 0, 9E-3, 2E-5 and -1E-4 mhos, respectively. The imaginary values are 3E-4, -8E-3, 0 and 1E-3 mhos, respectively.

A sample S-parameter data file is shown below. The only portion of the file required for GENESYS is the segment in the middle with frequencies and S-parameter data. Lines in the data file beginning with "!" are comments and are ignored. The noise data at the end of the file is used for noise figure analysis. (Noise is discussed in a later section.)

```
! AT41435 S AND NOISE PARAMETERS
! Vce=8V Ic=10mA
! LAST UPDATED 06-1-89
# GHZ S MA R 50
!FREQ S11 S21 S12 S22
0.1 .80 -32 24.99 157 .011 82 .93 -12
0.5 .50 -110 1 2.30 108 .033 52 .61 -28
1.0 .40 -152 6.73 85 .049 56 .51 -30
1.5 .38 176 4.63 71 .063 59 .48 -32
2.0 .39 166 3.54 60 .080 58 .46 -37
2.5 .41 156 2.91 53 .095 61 44 -40
3.0 .44 145 2.47 43 .115 61 .43 -48
3.5 .46 137 2.15 33 .133 58 .43 -58
4.0 .46 127 1.91 23 .153 53 .45 -68
4.5 .47 116 1.72 13 .178 50 .46 -75
5.0 .49 104 1.58 3 .201 47 .48 -82
```

```
6.0 .59 81 1.34 -17 .247 36 .43 -101

! f      Fmin     Gammaopt      Rn/Zo
! GHz    dB         MAG   ANG    -

0.1      1.2      .12    3       0.17
0.5      1.2      .10    14      0.17
1.0      1.3      .05    28      0.17
2.0      1.7      .30    -154    0.16
4.0      3.0      .54    -118    0.35
```

A sample 1 port Z parameter data file is shown below. This data file could be used to specify a port impedance that varied over frequency. Notice that the data is real and imaginary (RI) impedance (Z) data taken across several frequency points (13.90 to 14.45 MHz) that has been normalized to 1 ohm (R 1).

```
# MHZ Z RI R 1
13.90 30.8 -29.2
14.00 31.6 -6.6
14.05 32.0 4.7
14.10 32.4 16.0
14.15 32.7 27.2
14.20 33.1 38.4
14.25 33.5 49.5
14.30 33.9 60.7
14.35 34.3 71.7
14.45 35.1 93.7
```

File Record Keeping

Most device files provided with GENESYS are S-parameter files in the usual device configuration, typically common emitter or common source. Devices you add to the library may use the ground terminal of your choice. However, if you always keep data in a consistent format, record keeping chores are greatly minimized.

Exporting Data Files

"Export/S-Parameters" in the File menu writes S-parameter data from any simulation or data source. This output data file has exactly the same format as S-parameter files used to import data. This allows the user to analyze, tune and optimize sub-networks which are then stored as S-parameter data files for use later in other circuit files. The S-parameter data file written by GENESYS has one line of data for each simulation frequency. If there are two or more available simulations or designs in the circuit file, GENESYS displays a dialog box to allow you to select the simulation or design to use.

Noise Data in Data Files

Some of the data files provided with GENESYS also include noise data used for noise figure analysis. This data includes the optimum noise figure (NFopt), the complex source impedance to present to the device to achieve the optimum noise figure (Gopt), and the effective noise resistance (Rn).

The best noise figure in a circuit is achieved when the device is presented with an optimum source impedance. The optimum input network to achieve this objective does not in general result in an excellent return loss match. Balanced amplifiers and isolators are sometimes used to achieve both the optimum noise figure and a good match.

Losses in the input network, feedback networks around the transistor, emitter feedback and multiple stages all effect the noise figure of the circuit. All of these effects are accurately simulated in GENESYS using the noise correlation matrix technique [5,6].

Noise parameters can be added to the two port data files after the S, Y, G, H, or Z parameters. See the section Creating New Data Files for information about entering S, Y, G, H, or Z parameters. Each line of a noise parameter has the following five entries:

Frequency NF(dB) Mag_Gamma_Opt Ang_Gamma_Opt Rn/Zo

Frequency - Frequency in units

NF(dB) - Minimum noise figure in dB

Mag_Gamma_Opt - Magnitude of the optimum source reflection coefficient for minimum noise figure

Ang_Gamma_Opt - Angle of the optimum source reflection coefficient for minimum noise figure

Rn/Zo - Normalized effective noise resistance

Here is an example of noise data in a file along with the device S parameters

```
! BFP620 , Si-NPN RF-Transistor in SOT343
!  Vce=2 V,  Ic=8 mA
! Common Emitter S-Parameters:          01. February 2000
#  GHz  S  MA  R  50
! f        S11          S21          S12          S22
! GHz     Mag   Ang     Mag   Ang     Mag   Ang     Mag   Ang
 0.010  0.8479   -1.3  21.960  179.3  0.0024   27.9  0.9851    0.4
 0.020  0.8424   -1.9  21.606  178.2  0.0021   34.2  0.9676   -1.5
 0.050  0.8509   -5.7  21.650  175.6  0.0047   66.0  0.9693   -3.8
 0.100  0.8391  -10.7  21.434  171.7  0.0092   74.1  0.9662   -7.7
 0.150  0.8420  -16.8  21.349  167.3  0.0138   74.1  0.9584  -11.6
 0.200  0.8312  -21.8  21.109  163.1  0.0183   76.1  0.9477  -15.4
```

```
 0.250  0.8150  -27.0  20.679  159.1  0.0221   72.0  0.9326  -19.2
 0.300  0.8049  -32.8  20.328  155.0  0.0267   70.8  0.9157  -22.9
 0.500  0.7349  -52.7  18.378  140.2  0.0409   63.9  0.8256  -36.4
 0.700  0.6653  -71.1  16.211  127.7  0.0531   57.2  0.7212  -47.9
 0.900  0.5930  -87.2  14.148  117.7  0.0614   52.2  0.6237  -57.4
 1.100  0.5403 -101.2  12.427  109.4  0.0695   48.8  0.5440  -65.0
 1.300  0.4982 -113.9  11.019  102.6  0.0747   46.1  0.4736  -72.0
 1.500  0.4710 -125.4   9.834   96.5  0.0795   44.0  0.4168  -77.5
 1.700  0.4495 -135.7   8.861   91.3  0.0850   41.9  0.3619  -83.3
 1.900  0.4312 -145.4   8.013   86.4  0.0893   40.4  0.3209  -88.2
 2.000  0.4229 -150.0   7.670   84.1  0.0917   40.1  0.3021  -90.9
 3.000  0.4130  172.7   5.243   64.8  0.1147   34.7  0.1662 -120.6
 4.000  0.4749  144.9   3.914   48.5  0.1359   27.8  0.1538 -173.1
 5.000  0.5311  125.9   3.037   34.4  0.1524   21.3  0.1899  151.8
 6.000  0.5797  113.2   2.457   22.5  0.1699   15.8  0.2239  124.4
!
!  f      Fmin      Gammaopt      Rn/Zo where Zo = 50
! GHz      dB       MAG   ANG    -
  0.900     0.64    0.22     25    0.12
  1.800     0.71    0.09     97    0.08
  2.400     0.75    0.06    139    0.09
  3.000     0.87    0.09   -175    0.10
  4.000     0.99    0.19   -147    0.08
  5.000     1.17    0.26   -125    0.11
  6.000     1.34    0.38   -101    0.17
!
!              INFINEON TECHNOLOGIES, Munich
```

See reference [38] for more information on the relationship between noise figure and noise parameters.

Nonlinear Device Library

GENESYS is supplied with a large number of nonlinear parts in its libraries. Models for these parts are based on data supplied by the manufacturers-the best source of the latest part information.

The libraries consist of several workspace files that contain models for a range of parts from each manufacturer. You can examine the exact contents of any of the files by simply opening the workspace files, which are located in the Model subdirectory.

Most models are represented as a link to a SPICE file. The spice files are located in the model directory in their native ASCII form. This link is transparent to you when you place a model in a schematic.

To use a model from the library, just enter the base nonlinear model that you would like (for example, a PNP), and then change the model to the desired part using the "Model..." button on the part parameter dialog.

You can use your nonlinear device with four methods:

- Link to an existing SPICE file
- Create a single part model
- Create a schematic-based User Model
- Enter the parameters directly into a nonlinear model on the schematic.

Chapter 11 User Models

Overview

User models allow the creation of new elements by the user. These models behave just as if they were built into GENESYS. This capability is one of the more powerful features in GENESYS.

To create a new model, you must generally know three things:

1. An equivalent circuit for the model.
2. Equations which define the component values in the equivalent circuit.
3. The parameters that will be specified (if any) each time that the model is used. You can name and give descriptions for each of parameters.

A model can be created from any existing schematic or from scratch.

In addition, the user can create a new symbol for this model which will aid in distinguishing the new model from other schematic elements. See the section on symbols for details.

Creating A Model

Note: You must have purchased SCHEMAX to create and save a schematic model. If you have not, you may create a text model definition using the process described later in this section.

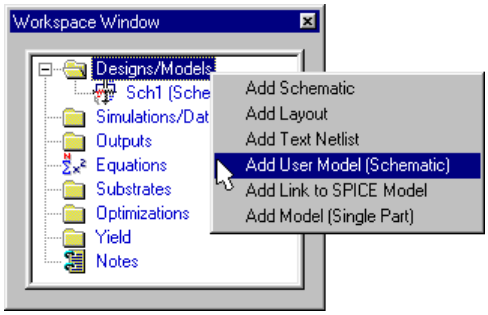
There are 2 ways to create a new model:

1. Without an existing schematic.
2. From an existing schematic of the model.

Creating A Model Without An Existing Schematic

With the Workspace Window:

1. Right-click on the Workspace Window "Designs" node as shown in the figure below.
2. Click "Add User Model (Schematic)".
3. Name the new model. Continue with step 5 in the Model Example below.



With the Design Manager:

1. Open the Design Manager by choosing "Designs/Models..." on the Workspace menu.
2. Click the "New" button.
3. Select "Add User Model (Schematic)".
4. Name the new model. Continue with step 5 in the Model Example below.

For more information on how to use the Workspace Manager dialogs, see the Reference manual.

Creating A Model From An Existing Schematic

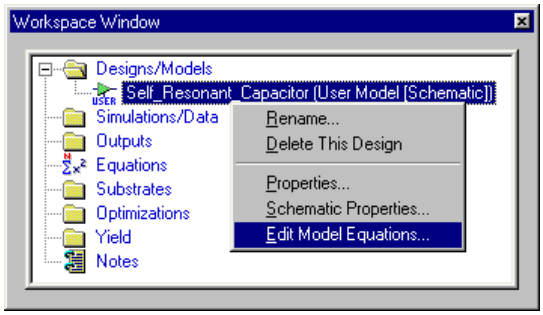
To create a model from an existing schematic:

1. Follow the instructions in "Creating A Model Without An Existing Schematic" above to create a blank model schematic.

Note: You do not have to define model parameters when the Model Properties dialog appears. By clicking OK, you can continue to create the model. However, the parameters (if any) must be defined, and a LAYOUT association chosen before the model can be used in a design.

---

2. Copy the existing schematic by selecting the entire schematic and choosing "Copy" from the Edit menu.
3. Paste the copied schematic into the model window by selecting the window and choosing "Paste" from the Edit menu.
4. Copy any equations from the Global Equations window by selecting them and choosing "Copy" from the Edit menu.
5. Right-click the model in the Workspace Window. (See the figure below.)



6. Choose "Edit Model Equations".
7. Paste the equations into the model by selecting the Model Equations window and choosing "Paste" from the Edit menu.

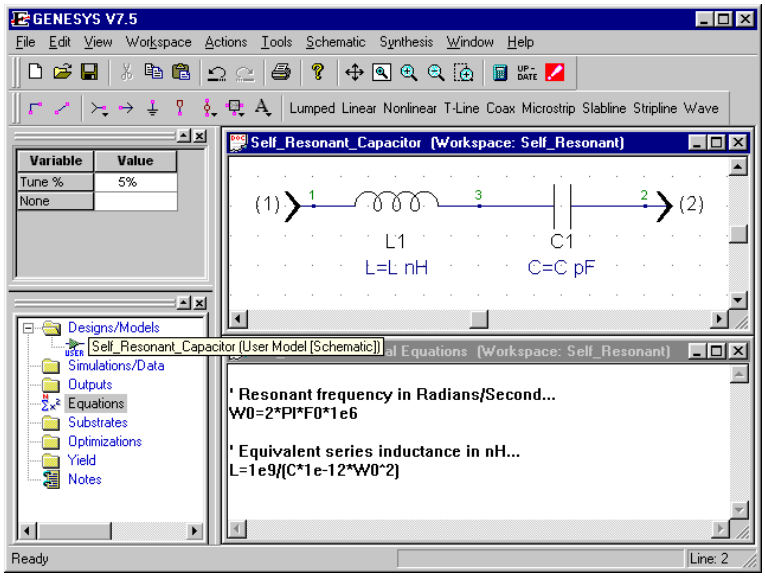
The model has now been created. If you chose to save the workspace into the MODEL directory, the model will load automatically each time GENESYS is started. This is the recommended method to share models with others.

User Model Example: A Self Resonant Capacitor

This example describes how to create a model for a self-resonant capacitor.

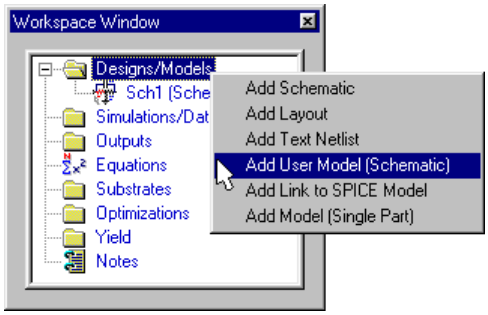
> **Note:** This example assumes that you are familiar with drawing schematics and entering parameters.

The figure below shows the model used in this example along with its equations.



To create this model:

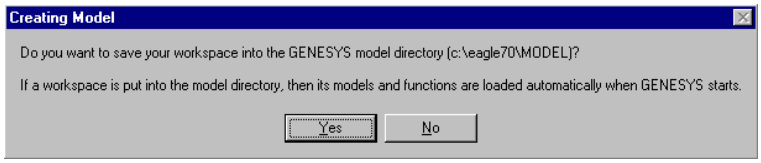
1. Create a new workspace by selecting "New" on the File menu.
2. Right-click the Designs/Models node in the Workspace Window as shown below:



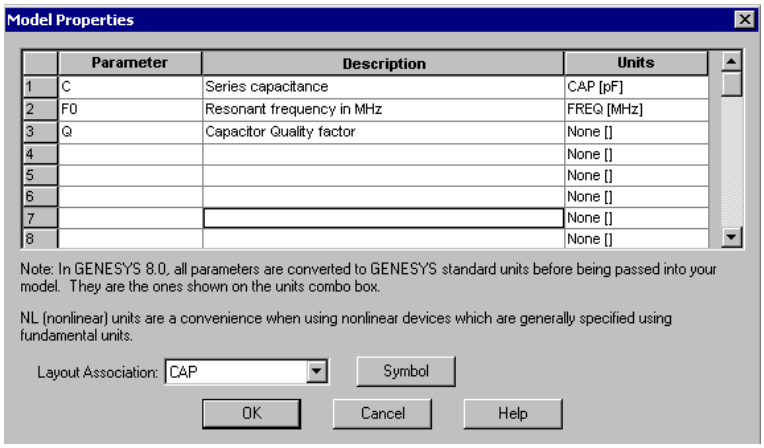
3. Select "Add User Model (Schematic)".
4. Name the model "Self_Resonant_Capacitor".

Note: Spaces are not allowed in model names, so it is important to use the underscore character(_) as shown. It is next to the zero on most American keyboards (with shift).

5. The following dialog appears:



6. If you answer "Yes" to this dialog, GENESYS will automatically load the model in the future, making it available for quick use.
7. In the Model Properties dialog, enter the following information:

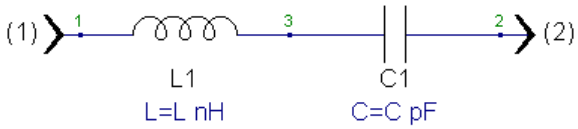


This box lists the parameters which must be passed to the model whenever it is used. The parameters for this example are:

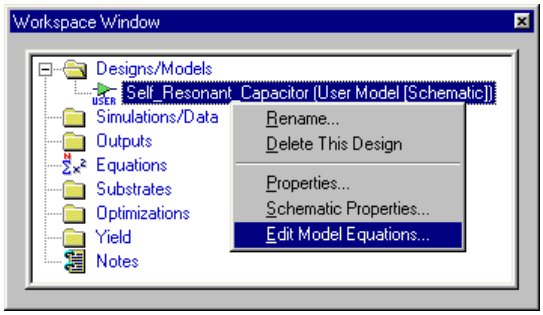
C - the actual capacitor value.
F0 - the frequency at which the capacitor self-resonates.
Q - the quality factor of the capacitor.

The "Layout Association" box associates this model with a normal capacitor when choosing footprints for board layouts.

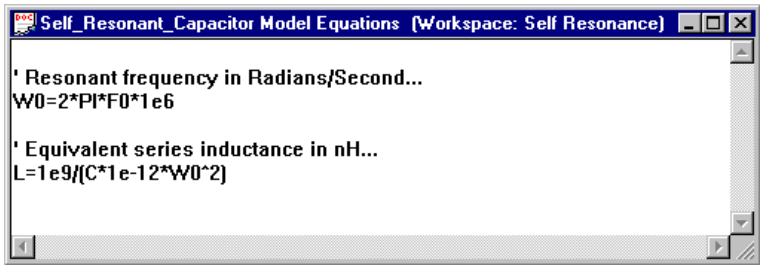
8. Press "OK" to close the Model Properties dialog.
9. Draw the schematic as shown in the figure below.



10. The inductor Q can be left blank, which defaults the value to 1 million. The capacitor Q should be set to "Q", which is one of the model parameters entered into the Model Properties dialog in step 5.
11. Right-click on the model in the Workspace Window as shown below.



12. Choose "Edit Model Equations".
13. Enter the equations as shown below:



14. This completes the model creation. Choose "Save" from the File menu to update the model file.

Next, let's create a schematic using the new model.

15. Choose "New" from the File menu.

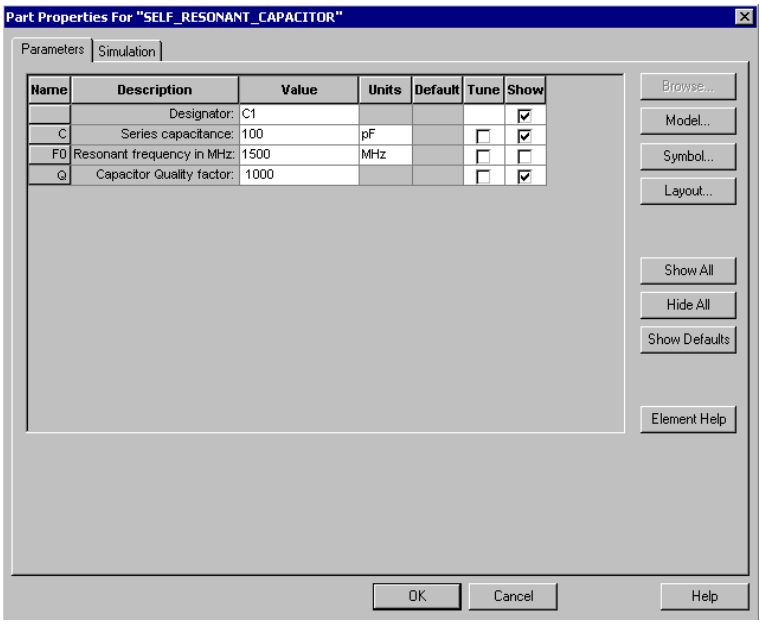
16. Draw a schematic consisting of only an input, a series capacitor, and an output, as shown below (don't set any parameters yet):

17. Double-click the capacitor symbol to display its Properties dialog.

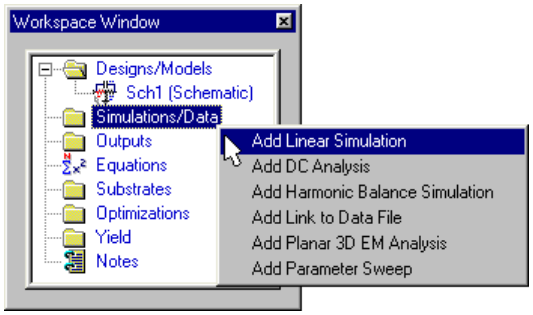
18. Click the Model button to open the Change Model dialog.

19. Set the category to SELF_RESONANT.wsp (or <All> to see every available model.) Then change the model to SELF_RESONANT_CAPACITOR as shown below, and click OK.

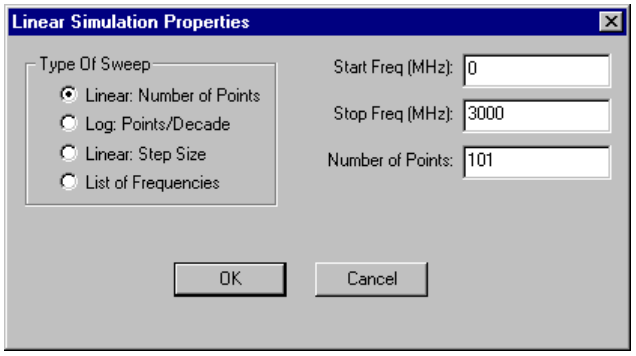
20. Now, the capacitor dialog changes to contain all the new model parameters, as shown below.



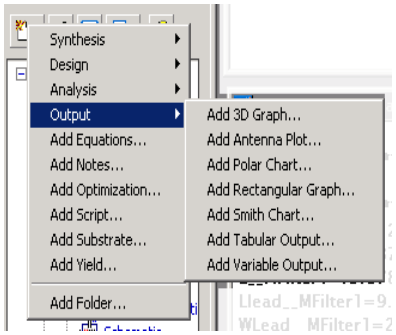
21. Right-click on the Simulations/Data node in the Workspace Window as shown below:



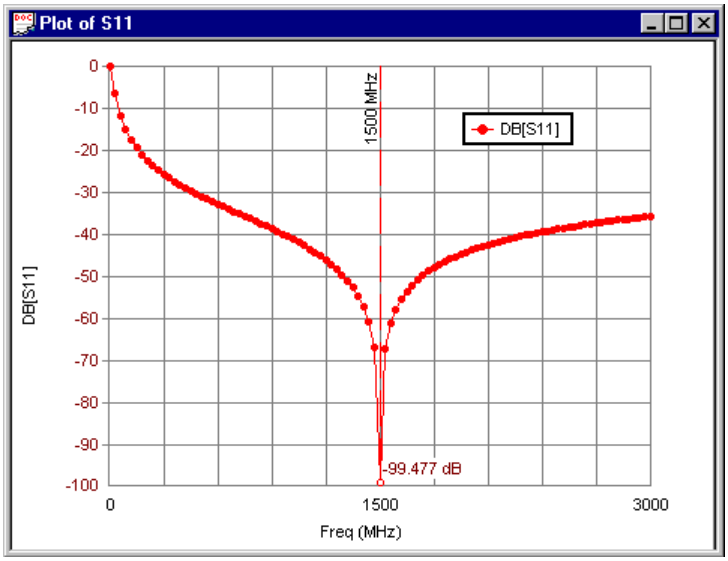
22. Add a linear simulation, and enter the parameters as shown below:



23. Right-click the Outputs node in the Workspace Window, as shown below:

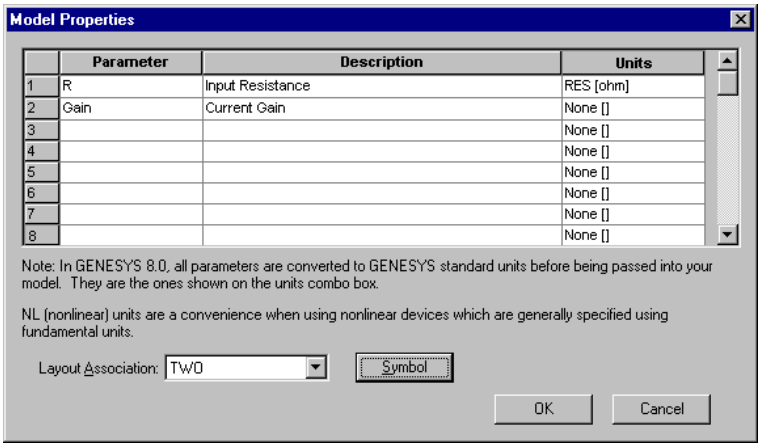


24. Add a rectangular graph, and plot S11. The figure below shows the plot from this example. This plot of S11 shows a return loss minimum at 1500 MHz, the capacitor's self resonant frequency.



Model Properties

To open: Create a new User Model.



Parameter - These entries are the parameter variables that can be used in model equations. When referring to these parameters in a model, they must appear precisely as entered here, with the exception of upper/lower case (they are not case-sensitive).

Description - Human-readable description of each parameter. This description is shown in SCHEMAX part dialog boxes.

Units - Describes what type of units that each parameter uses.

Layout Association - Defines which association table entry to use for this model. This defines the default footprint which will be used for this model when it is on a layout.

Symbol - Allows the user to select the schematic symbol associated with the model.

Using A Model In SCHEMAX

You can replace any element with a user defined model in SCHEMAX. To do this:

1. Double-click on an existing symbol that you have already drawn to change its model.
2. Press the Model button.
3. Choose the model to use from the combo box. An example is shown below.

4. Press OK.
5. Enter the parameters required for the model and press OK.

To place a user-defined model and special symbol on a schematic, follow the instructions for the "More" button in the previous section.

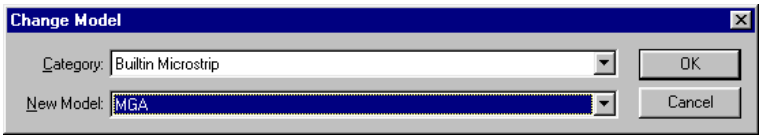
Single Part Models

Selecting "Add Model (Single Part)" from the Designs/Models right-click menu displays the dialog box shown below. This box defines the underlying part that will be used as the model. This dialog is the same as the Change Model dialog box in SCHEMAX.

The complete process to enter a single part model is:

1. Right click the Designs/Models icon in the workspace manager, and select "Add Model (Single Part)."
2. Name the part. The system asks if you want to store the file in the model directory. When stored in this directory, it is easy to reuse the part in other designs.
3. Choose a base model. The single part model can be based on any part. Typically this will be a nonlinear part such as a BJT model.
4. Enter the parameters for the part. The parameters that you enter will be used as the default for the part.
5. Use the part in a schematic. Enter a part that has a desired symbol. Change it to use your new Single Part model using the "Model..." button on the part dialog. Alternately, use the "More" button on the toolbar. Then change the model and symbol as prompted.

One advantage of the single part model is that default values can be easily overridden when used in a design. If you are used to Model statements in other simulators, single part models allow you to follow this paradigm while giving more flexibility.



Text Model Definitions

Note: The preferred method for creating models is to use the schematic based model editor described in User Model Example: A Self Resonant Capacitor.

If you do not have SCHEMAX, you may create a text description of your models. The format is as follows:

MODEL *name*(*parm1*,*parm2*,...)
[*model equation lines*]
[*model description lines*]
DEF*n*P *node1 node2...noden name*

where:

- *name* is the name of the model
- $parm_n$ are the parameters specified by the user
- *model equation lines* contain the equations for the model
- *model description lines* contain elements which make up the model
- *n* is the number of external nodes on the model
- $node_n$ are the external nodes used in the model descriptionlines

The text equivalent for the model given in \GENESYS\Model\Varactor.wsp is:

```
MODEL VARACTOR(Vt,Co,Gamma,Lp,Cp,Q)
Cv=Co/(1+Vt/0.7)^Gamma
C4=Co/(1+4/0.7)^Gamma
Rs=1/(3.14168e8*C4*1E-12*Q)
CAP 1 2 C=Cv
RES 2 3 R=Rs
CAP 1 3 C=Cp
IND 3 4 L=Lp
DEF2P 1 4 VARACTOR
```

This model can be typed or copied into a text file. You must then edit the Default.mod file in the \GENESYS\BIN directory: Add the line LIBRARY *filename*, where *filename* is the complete path and filename of your model. It can then be used as follows:

```
VARACTOR n1 n2 V=x Co=x G=x Lp=x Cp=x Q=x
```

Chapter 12 Link to Spice File

Overview

One of the easiest ways to get nonlinear device models into GENESYS for use with HARBEC is to use a link to a manufacturer supplied SPICE file. SPICE files have the following advantages over other methods of using nonlinear device data:

- They are often supplied by manufacturers.
- Entering device data manually is tedious and error-prone.
- SPICE files often contain very complete macromodel device characterizations.

They also have a few disadvantages:

- Model parameters cannot be tuned directly in GENESYS.
- SPICE data provided by manufacturers are often intended for low-frequency use and may not adequately characterize high-frequency behavior. (This is generally not a problem for devices intended for use at high frequencies.

To create a link to a SPICE file:

1. If workspaces using the spice link will be shared with co-workers, then we recommend placing the spice file either on a network drive which has the same letter for all co-workers, or, better, into your GENESYS\model directory or a subdirectory there.
2. Right-Click on Designs in the Workspace Tree and select Add Link to SPICE File.
3. If you want this link to be available automatically everytime you start GENSYS, you should answer Yes and then save your file into the model directory.
4. Click the "..." button on the SPICE Link dialog box and choose your spice library from the browse box. Press OK.
5. Choose the desired model or subcircuit from the combo box.
6. Normally, the only other necessary change in this box is checking or unchecking "Reverse Nodes 1 & 2". This box tells GENESYS that the spice subcircuit uses the spice node-numbering convention (Input=2, Output=1). Normally, you will check this box if the data represents a transistor or amplifier.
7. Click OK.
8. Generally, you should allow GENESYS to rename your model to be the same name as the spice model to avoid confusion.

Note: GENESYS will not allow 2 models with the same name to be loaded. If you create a SPICE model with the same name as an existing part, GENESYS will give an error at startup. If this happens, simply load your workspace and rename the spice link.

To use a link to a SPICE file:

1. On a schematic, place a part with the symbol you want for the link. For example, if you are placing a Bipolar Transistor, place a bipolar symbol.
2. Double-click on the part.
3. From the Schematic Element Properties Dialog box, click the model button.
4. From the Choose Model dialog box, choose the file and model with the spice link. Click OK, and Click OK.

SPICE File Compatibility

GENESYS is compatible with Berkeley SPICE3. Where possible, GENESYS has also been made compatibile with PSpice. The following devices can be used in a SPICE link:

B - Arbitrary Source. Note: SPICE 2 uses B for MESFET's. If you have a file using this convention, you must change the B prefix to Z and change the model name from MESFET to NMF or PMF

C - Capacitor

D - Nonlinear Diode

E,F,G,H - Controlled Sources

I - Current Source

J - Nonlinear JFET: Only JFET; JFET2 is not yet available.

K - Mutual Inductance (Only works in DC and HARBEC, not linear simulation)

L - Inductor

M - Nonlinear MOSFET: As of GENESYS Version 7.5 release, only level 1 (MOS1) is available.

Q - Nonlinear BJT

R - Resistor

V - Voltage Source

X - Subcircuit

Z - Nonlinear MESFET Transistors. Model types NMF and PMF are available. You must add a level parameter to the model to indicate which type of MESFET model will be used: 1=Curtice Quadratic, 2=Statz, 3=TOM, 4=Original SPICE 3F5 MESFET, 5=TOM2, 6=Curtice Cubic. For parameter details see the corresponding element in the Element Catalog.

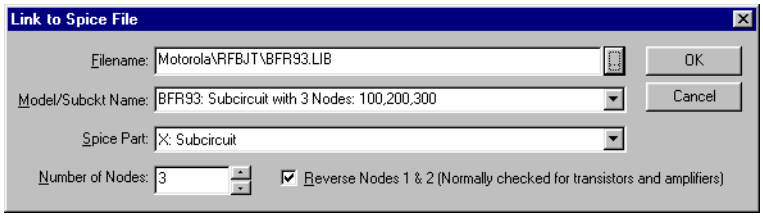
For example, a SPICE model for an XYZ143 device using a TOM N-Channel model in it might look like:

.model XYZ143 NMF (LEVEL=3 VTO=-2.5 CGS=1e-12)

If there are any compatibility errors in the SPICE file, the errors will appear in the GENESYS error window when a DC or HARBEC simulation is run which uses the link.

Link to Spice File

To open: Create a new Link to Spice Model.



Filename - The name of the library file containing the spice model. If you will be sharing files with other GENESYS users, you must use a uniform location for these files. Good locations include in the GENESYS\Model directory or a standard directory every user has on their C: drive.

... (Browse) - Clicking this button brings up a browse dialog box allowing you to search for your file. Even if you already have the correct file, this button is useful, as the browse dialog will show you the contents of the file.

Model/Subckt Name - This combo box is automatically filled in with available models and subcircuits parsed from the spice file. Select the model you want to use.

Spice Part - This box is automatically filled in when the model/subckt is selected. In SPICE, a model must be referred to using the correct type of part. Occasionally, you may need to override this selection.

Number of Nodes - This is automatically detected from the type of model, but can be overridden if necessary.

Reverse Nodes 1 & 2 - Unfortunately, Berkeley chose the convention that, for transistors, node 1 is the output and node 2 is the input. If this model is a transistor, or is a subcircuit with an amplifier that uses this convention, you should check this box. This box ensures that the GENESYS pictures and node numbering conventions are modified accordingly when using this part.

The Designs: Link to Spice File section in the User's guide has more detailed information.

Chapter 13 Measurements: Overview

Overview

GENESYS supports a rich set of output parameters. All parameters can be used for any purpose, including graphing, tabular display, optimization, yield, and post-processing.

Linear Measurements

The following table shows the available Measurements. Where *i* and *j* are shown in the chart, port numbers can be used to specify a port. Some parameters (such as A*i*) use only one port, e.g., A1 or VSWR2. Or, on a tabular output, the ports can be omitted (ie, S or Y), and measurements for all ports will be given.

Tip: All available measurements and their operators for a given circuit or sub-circuit with their appropriate syntax are shown in the measurement wizard. To bring up the measurement wizard select "measurement wizard" from the graph properties dialog box.

Note: The section in this manual on S Parameters contains detailed information about many of these parameters.

Meas.	Description	Default Operator	Shown on Smith Chart
S*ij*	S Parameters	DBANG	S*ij*
H*ij*	H Parameters*	RECT	--
YP*ij*	Y Parameters	RECT	--
ZP*ij*	Z Parameters	RECT	--
ZIN*i*	Impedance at port *i* with network terminations in place	RECT	S*ii*
YIN*i*	Admittance at port *i* with network terminations in place	RECT	S*ii*
ZPORT*i*	Reference Impedance at port *i*	RECT	
VSWR*i*	VSWR at port *i*	Linear (real)	S*ii*
E*ij*	Voltage gain from port *i* to port *j* with network terminations in place.	DBANG	--
N*ij*	Noise correlation matrix parameters	RECT	--
GMAX	Maximum available gain*	dB (real)	--
NF	Noise figure*	dB (real)	--
NMEAS	Noise measure*	Linear (real)	--

NFT	Effective noise input temperature*	Linear (real)	--
GOPT	Optimal gamma for noise*	DBANG	GOPT
YOPT	Optimal admittance for noise*	RECT	GOPT
ZOPT	Optimal impedance for noise*	RECT	GOPT
RN	Normalized noise resistance*	Linear (real)	--
NFMIN	Minimum noise figure*	dB (real)	--
ZM*i*	Simultaneous match impedance at port *i**	RECT	GM*i*
YM*i*	Simultaneous match admittance at port *i**	RECT	GM*i*
GM*i*	Simultaneous match gamma at port *i**	DBANG	GM*i*
K	Stability factor*	Linear (real)	
B1	Stability measure*	Linear (real)	
SB1	Input plane stability circle* Note: Filled areas are unstable regions.	None (Circle)	SB1 Circles
SB2	Output plane stability circle* Note: Filled areas are unstable regions.	None (Circle)	SB2 Circles
NCI	Constant noise circles* (shown at .25, .5, 1, 1.5, 2, 2.5, 3, and 6 dB less than optimal noise figure)	None (Circle)	NCI Circles
GA	Available gain circles**	None (Circle)	GA Circles
GP	Power gain circles**	None (Circle)	GP Circles
GU1	Unilateral gain circles at port 1**	None (Circle)	GU1 Circles
GU2	Unilateral gain circles at port 2**	None (Circle)	GU2 Circles

**Can only be used on 2-port networks*
***Gain circles are only available for 2-port networks. Circles are shown at 0, 1, 2, 3, 4, 5, and 6 dB less than optimal gain. In GA and GP, if K<1, then the 0dB circle is at GMAX, and the inside of this circle is shaded as an unstable region.*

Note: On a graph or in optimization, measurements which use DBANG by default show the dB part, measurements which use MAGANG show the magnitude, and measurements which use RECT show the real part.

Note: For port numbers greater that 9 a comma is used to separate port numbers. For example, on a 12 port device some of the S-Parameters would be specified as follows: S1,11 S12,2 S12,11 S12,2 .

Nonlinear Measurements

Tip: All available measurements and their operators for a given circuit or sub-circuit with their appropriate syntax are shown in the measurement wizard. To bring up the measurement wizard select "measurement wizard" from the graph properties dialog box.

Meas.	Description	Default Operator	Shown on Smith Chart
V*node*	Peak Voltage at *node (node is the node number or the name of the node as specified by the voltage test point designator name)*	MAG	--
I*probe*	Peak Current through *probe (probe is the current probe designator name)*	MAG	--
P*port*	RMS Power delivered at *port (port is the port number)*	DBM	--

Operators

Measurements are combined with operators to change the data format. The general format for combining operators with measurements is:

operator[*measurement*]

or

operator(*measurement*)

where *operator* is one of the operators listed in the table below and *measurement* is one of the measurements listed in the table in the previous section.

Also available is the @ operator which may be combined with any other measurement to select a subset of a sweep. Its format is:

operator[*measurement*]@*value*

where value is the independent value or range to pull data from. For ranges, separate values by : (colon).For multidimensional data, multiple ranges can be specified, separated by commas. The values can be the actual independent (frequency, etc.) data or can be #*index*, where index is the zero-based index of the data to use (such as a harmonic number in a nonlinear simulation). Some examples:

S21@900

Gives all data from S21 at 900 MHz. If the data comes from a parameter sweep, then the result will be a sweep of values, all at 900 MHz, vs. the swept parameter.

P2@#3

Returns the power in dBm at port 2 at the fourth data point (counting DC, that is the third harmonic for a single tone simulation).

MAG[V5]@0:3,1:3

Returns the magnitude of the voltage at node 5, from 0 to 3 for the first swept parameter and from 1 to 3 for the second parameter.

All measurements have default operators. For instance, on a table, using S21 will display in dB/angle form and Z32 will display in rectangular (real & complex) form. Likewise, on a graph, S21 graphs in dB, while Z32 graphs the real part of Z32.

Note: To avoid confusion, measurements used in equations for post-processing *must* specify an operator.

Operator	Description	Meas. must be	Result Is
MAGANG[]	Linear magnitude and angle in range -180 to 180	Complex	Complex*
MAGANG360[]	Linear magnitude and angle in range 0 to 360	Complex	Complex*
DBANG[]	dB magnitude and angle in range -180 to 180	Complex**	Complex*
DBANG360[]	dB magnitude and angle in range 0 to 360	Complex**	Complex*
RECT[]	Rectangular (real + imag)	Complex	Complex
MAG[]	Linear magnitude	Real/Complex	Real
ANG[]	Angle in range -180 to 180	Complex	Real
ANG360[]	Angle in range 0 to 360	Complex	Real
RE[]	Real part of complex measurement	Complex	Real
IM[]	Imaginary part of complex measurement	Complex	Real
DB[]	dB Magnitude	Real/Complex**	Real
GD[]	Group delay	Complex	Real
QL[]	Loaded Q [QL = (2 pi f) GD / 2]	Complex	Real
TIME[]	Converts Frequency domain to Time domain via inverse Fourier Transform. Intended for use with Voltage/Current to get time waveforms.	Complex	Real

**For post-processing equation purposes, the magnitude is in the real part of the result, and the angle is in the complex part of the result.*
***Only the following parameters can be displayed in dB form: S, GM, E, GOPT, GMAX, NF, NFMIN, and NMEAS.*

Note that not all operators can be used with all measurements. The "Measurement must be" column above indicates which type of parameter each operator can use. For example, ANG[] (Angle) cannot be used with a real-valued parameter, such as GMAX, so ANG[GMAX] is not allowed.

Note: All available measurements and their operators for a given circuit or sub-circuit with their appropriate syntax are shown in the measurement wizard. To bring up the measurement wizard select "measurement wizard" from the graph properties dialog box.

Sample Measurements

Measurement	Result in graph, Smith chart, optimization, or yield	Result on table
S22	dB Magnitude of S22	dB Magnitude plus angle of S22
QL[S21]	Loaded Q of S21	Loaded Q of S21
MAG[S21]	Linear Magnitude of S21	Linear Magnitude of S21
IM[Zin1]	Input reactance at port 1. On a Smith chart, S11 will be displayed, while IM[Zin1] will be used for the marker readouts.	Input reactance at port 1
S	---	Shows dB Magnitude plus angle of all S Parameters
RECT[S]	---	Shows real/imaginary parts of all S Parameters
SB1	On Smith or polar chart, shows input plane stability circles	Displays center, radius, and stability parameter of input plane stability circles
NCI	On Smith or polar chart, shows constant noise circles	Displays center, and radius of all noise circles (27 numbers per frequency)

Using Non-Default Simulation/Data

In all dialog boxes which allow entry of measurements, there is a "Default Simulation/Data or Equations" combo box. Any measurement can override this default. The format to override the network is:

simulation.design.operator[*measurement*]

where *simulation* is the name of the Simulation/Data from the Workspace Window, *design* is the name of the design to use, and *operator*[*measurement*] are as described in previous sections. An override is most useful for putting parameters from different networks on the same graph.

Additionally, the workspace can be overridden by using the following format:

workspace.simulation.design.operator[*measurement*]

where *workspace* is the short name of the workspace as given in the Workspace Window. This allows direct comparison of results from different workspaces.

Some examples of overrides are:

Meas.	Meaning
Linear1.Filter.DB[S21]	Show the dB magnitude of S21 from the Linear1 simulation of the Filter design
EM1.Layout1.S11	Show the dB magnitude of S11 from the EMPOWER analysis of Layout1
Filter.QL[S21]	Shows the loaded Q of the Filter design using the current simulation. Note that the simulation was not overriden, only the network.
DB[Linear1.FILTER.S21] **(wrong)**	ILLEGAL. The operator must go around the measurement, not the override.
Equations.X	Shows the global equation variable X, which must contain post-processed results.
TUNEBP.Linear1.Filter.DB[S21]	Overrides the workspace. Shows the dB magnitude of S21 from the Linear1 simulation of the Filter design from workspace TUNEBP.
Data1.A	Show all input admittances from a "Link to data file". Note that in this case, the design name is not required.

Using Equation Results (post-processing)

Anywhere that a measurement is used, post-processed equation variables can be used. The format is:

EQUATIONS.*variableName*

where *variableName* is a variable from the equations for that workspace. For example:

EQUATIONS.X

uses variable X from the global equations. A workspace name can also be used:

TUNEBP.EQUATIONS.Y

shows variable Y from the global equations of workspace TUNEBP.

Inline equations can also be used anywhere a measurement can be used. Start the measurement with = to indicate an inline equation. For example:

=.MAG[V1] - .MAG[V2]

will use the difference of V1 and V2. Notice that, as in the global equations, the periods and the operators (MAG[]) are required for inline equations. This measurement is actually equivalent to the following equations:

USING *MeasurementContext*
TEMP=.MAG[V1] - .MAG[V2]

and then requesting the measurement

EQUATIONS.TEMP

where *MeasurementContext* is the Default Simulation/Data specified in the measurement dialog.

Chapter 14 Measurements: Linear

S-Parameters

This S-parameter (or *scattering* parameter) measurements are complex functions of frequency. The frequency range and intervals are as specified in the Linear Simulation dialog box. The s-parameters assume a 50-ohm reference impedance unless otherwise specified. The s-parameters for an n-port network are of the form:

S_{ij} for i, j equal 1, 2, ... n

Details on the S-parameters and their application are found in Section x.x of this Manual.

Values: Complex matrix versus frequency.

Simulations: Linear, EMPOWER

Default Format: Table: dB, angle Graph: dB Smith Chart: dB, angle

Commonly Used Operators:

Operator	Description	Result Type
ANG[S11]	Angle in range -180 to 180 degrees	Real
GD[S22]	Group Delay	Real
QL[S21]	Loaded Q	Real

{ Other Operators: DB[], MAG[], RECT[], ANG360[], RE[], IM[], MAGANG[], MAGANG360[],DBANG[] }

Examples:

Measurement	Result in graph, Smith chart, optimization, or yield	Result on table
S22	dB Magnitude of S22	dB Magnitude plus angle of S22
QL[S21]	Loaded Q of S21	Loaded Q of S21
MAG[S21]	Linear Magnitude of S21	Linear Magnitude of S21
S	---	Shows dB Magnitude plus angle of all S Parameters
RECT[S]	---	Shows real/imaginary parts of all S Parameters
GD[S21]	Group delay of S21	Group delay of S21

Note: For port numbers greater that 9 a comma is used to separate port numbers. For example, on a 12 port device some of the S-Parameters would be specified as follows: S1,11 S12,2 S12,11 S12,2 .

H-Parameters

This H-parameter (or *hybrid* parameter) measurements are complex functions of frequency. The frequency range and intervals are as specified in the Linear Simulation dialog box. The H-parameters are only defined for a two port network, and are of the form:

H_{ij} for i, j equal 1, 2

The equations relating the input voltage (V_1) and current (I_1) to the output voltage (V_2) and current (I_2) are:

$$V_1 = H_{11} I_1 + H_{12} V_2$$

$$I_2 = H_{21} I_1 + H_{22} V_2$$

Values: Complex matrix versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
RECT[H11]	real/imaginary parts	Real
RE[H22]	real part	Real
MAGANG[H21]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[], MAGANG360[] }

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
H22	RE[H22]	real part of H22
RECT[H]	---	Shows real/imaginary parts of all H Parameters
MAG[H21]	Linear Magnitude of H21	Linear Magnitude of H21
H	---	Shows real/imaginary parts of all H Parameters

* Not available on Smith Chart

Y-Parameters

This Y-parameter (or *admittance* parameter) measurements are complex functions of frequency. The frequency range and intervals are as specified in the Linear Simulation dialog box. The Y-parameters for an n-port network are of the form:

YP_{ij} for i, j equal 1, 2, ...n

For a two port network, the equations relating the input voltage (V_1) and current (I_1) to the output voltage (V_2) and current (I_2) are:

$I_1 = YP_{11} V_1 + YP_{12} V_2$

$I_2 = YP_{21} V_1 + YP_{22} V_2$

Values: Complex matrix versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
RECT[YP11]	real/imaginary parts	Real
RE[YP22]	real part	Real
MAGANG[YP21]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[] , MAGANG360[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
YP22	RE[YP22]	real part of YP22
RECT[YP]	---	Shows real/imaginary parts of all Y Parameters
MAG[YP21]	Linear Magnitude of YP21	Linear Magnitude of YP21
YP	---	Shows real/imaginary parts of all Y Parameters

* Not available on Smith Chart

Z-Parameters

This Z-parameter (or *impedance* parameter) measurements are complex functions of frequency. The frequency range and intervals are as specified in the Linear Simulation dialog box. The Z-parameters for an n-port network are of the form:

ZP_{ij} for i, j equal 1, 2, ...n

For a two port network, the equations relating the input voltage (V_1) and current (I_1) to the output voltage (V_2) and current (I_2) are:

$V_1 = ZP_{11} I_1 + ZP_{12} I_2$

$V_2 = ZP_{21} I_1 + ZP_{22} I_2$

Values: Complex matrix versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
RECT[ZP11]	real/imaginary parts	Real
RE[ZP22]	real part	Real
MAGANG[ZP21]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[] , MAGANG360[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
ZP22	RE[ZP22]	real part of ZP22
RECT[ZP]	---	Shows real/imaginary parts of all Z Parameters
MAG[ZP21]	Linear Magnitude of ZP21	Linear Magnitude of ZP21
ZP	---	Shows real/imaginary parts of all Z Parameters

* Not available on Smith Chart

Voltage Standing Wave Ratio (VSWR)

The VSWR measurement is a real function of frequency. The measurements are made looking into the network from the port with other network terminations in place. The frequency range and intervals are as specified in the Linear Simulation dialog box. A port number "*i*" is used to identify the port:

VSWR*i* is the Voltage Standing Wave Ratio looking in from port *i*.

The VSWR is a measure of the energy reflected back to the port. The $VSWR_1$ is related to the s-parameter S_{11} by:

$$VSWR_1 = [\,1 + |S_{11}|\,] \;/\; [\,1 - |S_{11}|\,]$$

Therefore, as the reflected energy goes to zero, $|S_{11}|$, goes to zero and the VSWR approaches unity. As the reflected energy increases, $|S_{11}|$ approaches unity, and VSWR goes to infinity.

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: RE (Real) Graph: RE (Real) Smith Chart: S_{ij} (plots s-parameters)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
VSWR1	VSWR1	VSWR

VSWR	---	Show VSWR for all ports

* Not available on Smith Chart, plots s-parameters

Input Impedance / Admittance (ZINi, YINi)

The port impedance and admittance measurements are complex functions of frequency. The measurements are made looking into the network from the port with other network terminations in place. The frequency range and intervals are as specified in the Linear Simulation dialog box. A port number "*i*" is used to identify the port:

ZIN*i* is the input impedance looking in from port *i*.

YIN*i* is the input admittance looking in from port *i*.

Values: Complex value versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: S_{ij} (plots s-paramters)

Commonly Used Operators:

Operator	Description	Result Type
RECT[ZIN1]	real/imaginary parts	Real
RE[YIN2]	real part	Real
MAGANG[ZIN3]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[], MAGANG360[] }

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
ZIN2	RE[ZIN2]	real part of ZIN2
RECT[ZIN]	---	Shows real/imaginary parts for all ports
MAG[YIN1]	Linear Magnitude of Y21	Linear Magnitude of YIN1
ZIN	RE[ZIN1]	Shows real/imaginary parts of all ports

* Not available on Smith Chart

Voltage Gain

This voltage gain measurements are complex functions of frequency. The frequency range and intervals are as specified in the Linear Simulation dialog box. The voltage gain, E_{ij} , is the ratio of the output voltage (V_j) to the input voltage (V_i).

$E_{ij} = V_j / V_i$

Note that due to reflections, the gain E_{ii} may not be unity.

Values: Complex matrix versus frequency.

Simulations: Linear

Default Format: Table: DBANG Graph: dB Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[E12]	gain from port 1 to port 2	Real
DBANG[E21]	db and angle in range of -180 to 180 for gain from port 2 to 1	Real

{Other Operators: MAG[], ANG[], ANG360[], RE[], IM[] , MAGANG360[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
E12	DB[E12]	DBANG[E12]
E	---	Shows db/angle for all E_{ij}

* Not available on Smith Chart

Noise Measure (NMEAS)

The "Noise Measure" measurement is a real function of frequency and is available for 2-port networks only.

The noise measure is defined in terms of the noise figure (NF) and maximum available gain (GMAX) as:

NMEAS = [NF - 1] / [1 - (1 / GMAX)]

The noise measure represents the noise figure for an infinite number of networks in cascade.

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: MAG[] Graph: MAG[] Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[NMEAS]	noise measure in dB	Real
MAG[NMEAS]	magnitude of the noise measure	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
NMEAS	MAG[NF]	MAG[NF]
DB[NMEAS]	magnitude of the minimum noise measure	magnitude of the minimum noise measure

* Not available on Smith Chart

Noise Figure (NF) / Minimum Noise Figure (NFMIN)

The "Noise Figure" measurements are real functions of frequency and are available for 2-port networks only.

The noise figure is defined as the ratio of input signal-to-noise power ratio (SNR_{IN}) to the output signal-to-noise ratio (SNR_{OUT}):

$$NF = SNR_{IN} / SNR_{OUT}$$

The noise figure is related to the minimum noise figure (NFMIN) by the expression:

$$NF = NFMIN + \{R_N / G_S\} * | Y_S - Y_{OPT} |^2$$

where

$Ys = Gs + j\ Bs$ = Source Admittance
R_N = Normalized Noise Resistance

The minimum noise figure represents the noise figure with ideal match of source impedance (i.e. $Y_S = Y_{OPT}$)

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: dB Graph: dB Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[NF]	noise figure in dB	Real
MAG[NF]	magnitude of the noise figure	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
NF	DB[NF]	DB[NF]
MAG[NFMIN]	magnitude of the minimum noise figure	magnitude of the minimum noise figure

* Not available on Smith Chart

Constant Noise Circles (NCI)

A noise circle is a locus of load impedances for a given noise figure as a function of frequency. This locus is plotted on a Smith chart, with noise figure degradations of 0.25, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0 and 6.0 dB from the optimal noise figure.

Note: See the section on S-Parameters for a detailed discussion of noise circles.

Values: Complex values versus frequency.

Simulations: Linear

Default Format: Table: center (MAG[], ANG[]), radius (Linear) Graph: (none) Smith Chart: Circles (6)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
NCI	noise circle	locus of load impedances for optimal noise figure, for each circle: center :MAG[], ANG[] radius: Linear

* Available on Smith Chart and Table only.

Noise Correlation Matrix Parameters

The noise correlation matrix elements are complex functions of frequency. The frequency range and intervals are as specified in the Linear Simulation dialog box. For a "n" noise sources, the elements are of the form:

N_{ij} for i, j equal 1, 2

Note: See References [5,6] for a complete discussion of noise correlation matrix properties.

Values: Complex matrix versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
RECT[N11]	real/imaginary parts	Real
RE[N22]	real part	Real
MAGANG[N21]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[], MAGANG360[] }

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
N22	RE[N22]	real part of N22
RECT[N]	---	Shows real/imaginary parts of all N Parameters
MAG[N21]	Linear Magnitude of H21	Linear Magnitude of N21
N	---	Shows real/imaginary parts of all N Parameters

* Not available on Smith Chart

Simultaneous Match Gamma at Port i (GMi)

The "Simultaneous Match Gamma" is a complex function of frequency and is available for 2-port networks only. Computes the reflection coefficient that must be seen by the input port *i* to achieve a simultaneous conjugate match at both the input and output.

Values: Complex value versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: GM*i*

Commonly Used Operators:

Operator	Description	Result Type
RECT[GM1]	real/imaginary parts	Real
RE[GM1]	real part	Real
MAGANG[GM2]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[] , MAGANG360[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
GM	---	real / imaginary parts of gamma for all ports
GM1	RE[GM1]	RECT[GM1]

Simultaneous Match Admittance / Impedance at Port i (ZMi, YMi)

The "Simultaneous Match Admittance" is a complex function of frequency and is available for 2-port networks only.

This is the value of admittance which must be seen at port *i* to achieve a simultaneous match at both input and output.

Values: Complex value versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: GM*i*

Commonly Used Operators:

Operator	Description	Result Type
RECT[YM1]	real/imaginary parts	Real
RE[YM1]	real part	Real
MAGANG[ZM2]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[] , MAGANG360[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
YM	---	real / imaginary parts of admittance for all ports
ZM1	RE[ZM1]	RECT[ZM1]

Maximum Available Gain (GMAX)

The "Maximum Available Gain" measurement is a real function of frequency and is available for 2-port networks only.

For conditions where the stability factor (K) is greater than zero, i.e. the system is unconditionally stable, then:

$$GMAX = (|S_{21}| / |S_{12}|) * (K - sqrt(K^2 - 1))$$

If K < 1, then GMAX is set to the maximum stable gain, therefore:

$$GMAX = |S_{21}| / |S_{12}|$$

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: dB Graph: dB Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[GMAX]	maximum available gain in dB	Real
MAG[GMAX]	magnitude of the maximum available gain	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
GMAX	DB[GMAX]	DB[GMAX]
MAG[GMAX]	magnitude of the maximum available gain	magnitude of the maximum available gain

* Not available on Smith Chart

Available Gain & Power Gain Circles (GA, GP)

An available gain input network circle is a locus of source impedances for a given gain below the optimum gain. This locus is plotted on a Smith chart, and is only available for 2-port networks. The center of the circle is the point of maximum gain. Circles are displayed for gains of 0, 1, 2, 3, 4, 5, and 6 dB less than the optimal gain. Similarly, the power gain output network circle is a locus of load impedances for a given gain below the optimum gain. If the stability factor K is less than unity, then the 0 dB circle is at GMAX, and the inside of this circle is shaded as an unstable region. The available power gain (G_a) and power gain (G_p) are defined as:

G_a = (available from network) / (power available from source)

G_p = (power deliver to load) / (power input to network)

Note: See the section on S-Parameters for a detailed discussion of Gain Circles.

Values: Complex values versus frequency.

Simulations: Linear

Default Format: Table: center (MAG[], ANG[]), radius (Linear) Graph: None Smith Chart: Circle

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
GA	available gain circles	center :MAG[], ANG[] radius:Linear
GP	power gain circles	center :MAG[], ANG[] radius:Linear

* Available on Smith Chart and Table only.

Unilateral Gain Circles at Port i (GU1, GU2)

A unilateral gain circle at port 1 is a locus of source impedances for a given transducer power gain below the optimum gain. This locus is plotted on a Smith chart, and is only available for 2-port networks. The center of the circle is the point of maximum gain. Circles are displayed for gains of 0, 1, 2, 3, 4, 5, and 6 dB less than the optimal gain. Similarly, the unilateral gain circle at port 2 is a locus of load impedances for a given transducer power gain below the optimum gain. The transducer power gain (G_t) is defined as:

G_t = (power deliver to load) / (power available from source)

For the "unilateral" transducer gain, S_{12} is set to zero.

Note: See the section on S-Parameters for a detailed discussion of Gain Circles.

Values: Complex values versus frequency.

Simulations: Linear

Default Format: Table: center (MAG[], ANG[]), radius (Linear) Graph: (none) Smith Chart: Circles

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
GU1	unilateral gain circle at port 1	center :MAG[], ANG[] radius:Linear
GU2	unilateral gain circle at port 2	center :MAG[], ANG[] radius:Linear

* Available on Smith Chart and Table only.

Stability Factor (K), Stability Measure (B1)

The "Stability Factor and Measure" parameters are real functions of frequency and are available for 2-port networks only. These parameters aid in determining the stability of the 2-port network. If S_{12} of a device is not zero, a signal path will exist from the output to the input. This feedback path creates an opportunity for oscillation. The stability factor, K, is:

$$K = (1 - |S_{11}|^2 - |S_{22}|^2 + |D|^2) / (2 |S_{12}| |S_{21}|)$$

where $D = S_{11}S_{22} - S_{12}S_{21}$

From a practical standpoint when $K>1$, $S_{11}<1$, and $S_{22}<1$, the two-port is unconditionally stable. These are often stated as sufficient to insure stability. Theoretically, $K>1$ by itself is insufficient to insure stability, and an additional condition should be satisfied. One such parameter is the stability measure,B1, which should be greater than zero.

$$B1 = 1 + |S_{11}|^2 - |S_{22}|^2 - |D|^2 > 0$$

Note: See the section on S-Parameters for a detailed discussion of stability analysis.

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
K	stability factor	stability factor
B1	stability measure	stability measure

* Not available on Smith Chart

Input / Output Plane Stability Circles (SB1, SB2)

A output stability circle is a locus of load impedances for which the input reflection coefficient (S_{11}) is unity. This locus is plotted on a Smith chart, and is only available for 2-port networks. This locus is a circle with radius R_{out} about a point C_{out}, where:

$$R_{out} = | S_{12}S_{21} / (|S_{22}|^2 - |D|^2) | \qquad C_{out} = (S_{22} - DS_{11}) / (|S_{22}|^2 - |D|^2)$$

The region inside or outside the circle may be the stable region. The filled areas of the graphs are the unstable regions. The input plane stability circle equations are the same as the output plane equations, with 1 and 2 in the subscripts interchanged.

If SB1 and SB2 are placed on a table, you can see the PAR value. If it is zero, then the region is outside the circle is stable. If it is 180, then the region inside the circle is stable.

Note: See the section on S-Parameters for a detailed discussion of stability analysis.

Values: Complex values versus frequency.

Simulations: Linear

Default Format: Table: center (MAG[], ANG[]), radius (Linear), par Graph: (none) Smith Chart: Circle

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
SB1	input stability circle	center :MAG[], ANG[] radius:Linear "par"**
SB2	output stability circle	center :MAG[], ANG[] radius:Linear "par"**

* Available on Smith Chart and Table only.

** Parameter indicating the unstable region.

Optimal Gamma for Noise (GOPT)

The "Optimal Gamma for Noise" is a real function of frequency and is available for 2-port networks only.

The optimal gamma is defined in terms of the reference admittance (Y_o) and the optimal value of admittance (Y_{OPT}) as:

$$GOPT = [Y_o - Y_{OPT}] / [Y_o + Y_{OPT}]$$

Notice that gamma goes to zero if the reference admittance is optimal.

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: Linear Graph: Linear Smith Chart: GOPT

Commonly Used Operators: none

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
GOPT	gamma coefficient	gamma coefficient

Optimal Admittance / Impedance for Noise (YOPT, ZOPT)

The "Optimal Admittance for Noise" is a complex function of frequency and is available for 2-port networks only.

The optimal admittance is the value of the input admittance which minimized the noise figure of the network. The optimal admittance is defined in terms of the source admittance (Y_S) and the noise resistance (R_N) and the noise figures (NF, NFMIN) as:

$$NF = NFMIN + \{R_N / Re[Y_S]\} \mid Y_S - Y_{OPT} \mid$$

The optimal impedance is the inverse of the optimal admittance, i.e. $Z_{OPT} = 1 / Y_{OPT}$

Values: Complex value versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: GOPT

Commonly Used Operators:

Operator	Description	Result Type
RECT[YOPT]	real/imaginary parts	Real
RE[YOPT]	real part	Real
MAGANG[YOPT]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[] , MAGANG360[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
YOPT	real part of optimal admittance	real / imaginary parts of admittance

Effective Noise Input Temperature (NFT)

The "Effective Noise Input Temperature" is a real function of frequency and is available for 2-port networks only.

The effective noise temperature is defined in terms of the noise figure (NF) and a standard temperature (T_o) in degrees Kelvin as:

$$NFT = T_o * [NF - 1] \qquad \text{where } T_o = 300 \text{ degrees Kelvin}$$

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: none

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
NFT	noise temperature in degrees Kelvin	noise temperature in degrees Kelvin

* Not available on Smith Chart

Normalized Noise Resistance (RN)

The "Normalized Noise Resistance" measurement is a real function of frequency and is available for 2-port networks only.

The noise resistance is normalized with respect to the input impedance of the network (Zo). See the definition of Nosie Figure (NF) for a discussion of R_N.

Values: Real value versus frequency.

Simulations: Linear

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: none

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
RN	noise resistance	noise resistance

* Not available on Smith Chart

Reference Impedance (ZPORTi)

The reference impedance measurements are complex functions of frequency. The measurements are associated with the network terminations. The frequency range and intervals are as specified in the Linear Simulation dialog box. A port number "*i*" is used to identify the port:

ZPORT*i* is the reference impedance for port *i*.

Values: Complex value versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
RECT[ZPORT1]	real/imaginary parts	Real
RE[ZPORT2]	real part	Real
MAGANG[ZPORT3]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[], MAGANG360[] }

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
ZPORT2	RE[ZPORT2]	RECT [ZPORT2]
RECT[ZPORT]	- - -	Shows real/imaginary parts for all ports
MAG[ZPORT1]	Linear Magnitude of ZPORT2	Linear Magnitude of ZPORT1
ZPORT	- - -	Shows real/imaginary parts of all ports

* Not available on Smith Chart

Chapter 15 Measurements: Nonlinear

Port Power (Pport)

This power measurement is the RMS power delivered at the port. The port is identified by a port designator number.

Values: Real value in specified units.

Simulations: Nonlinear (dc analysis).

Default Format: Table: DBM Graph: DBM Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[P1]	RMS power at port 1	Real

{ Other Operators: DB[], MAG[], ANG[], ANG360[], RE[], IM[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
P1	DBM[P1] = RMS power delivered to port 1	DBM[P1]

* Not available on Smith Chart

Probe Current (Iprobe)

This current measurement is the peak current through the specified current probe. The probe is identified by a probe designator name.

Values: Real value in specified units.

Simulations: Nonlinear (dc analysis).

Default Format: Table: MAG Graph: MAG Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
MAG[I1]	linear magnitude of voltage at probe 1	Real

{ Other Operators: DB[], ANG[], ANG360[], RE[], IM[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
ICP1	MAG[ICP1] = current through current probe 1	MAG[ICP1]

* Not available on Smith Chart

Node Voltage (Vnode)

This voltage measurement is the peak voltage at the specified node. The node is the node number or the name of the node as specified by the voltage test point designator name.

Values: Real value in specified units.

Simulations: Nonlinear (dc analysis).

Default Format: Table: MAG Graph: MAG Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
MAG[V1]	linear magnitude of voltage at node 1	Real

{ Other Operators: DB[], DBM[], ANG[], ANG360[], RE[], IM[]}

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
VTP2	MAG[VTP2] = voltage at test point TP2	MAG[VTP2]

* Not available on Smith Chart

Reference Impedance (ZPORTi)

The reference impedance measurements are complex functions of frequency. The measurements are associated with the network terminations. The frequency range and intervals are as specified in the Linear Simulation dialog box. This measurement is the same as the linear measurement of the same name. A port number "*i*" is used to identify the port:

ZPORT*i* is the reference impedance for port *i*.

Values: Complex value versus frequency.

Simulations: Linear

Default Format: Table: RECT Graph: RE Smith Chart: (None)

Commonly Used Operators:

Operator	Description	Result Type
RECT[ZPORT1]	real/imaginary parts	Real
RE[ZPORT2]	real part	Real
MAGANG[ZPORT3]	Linear magnitude and angle in range of -180 to 180	Real

{Other Operators: MAG[], ANG[], ANG360[], IM[], MAGANG360[] }

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table

ZPORT2	RE[ZPORT2]	RECT [ZPORT2]
RECT[ZPORT]	- - -	Shows real/imaginary parts for all ports
MAG[ZPORT1]	Linear Magnitude of ZPORT2	Linear Magnitude of ZPORT1
ZPORT	- - -	Shows real/imaginary parts of all ports

* Not available on Smith Chart

Large Signal S Parameters

Large signal S-parameters can be measured in a HarBEC simulation. Unlike linear S-parameters, nonlinear large-signal S-parameters depend upon the signal magnitude and must take into account the harmonic content of the input and output signals since energy can be transferred to other frequencies in a nonlinear device. Large signal S-parameters are very useful in characterizing devices operating in the nonlinear range.

Note: You must have a signal at the input port to use these measurements.

Measurements

There are three large signal S-parameters measurements. They are described as follows:

=LargeSdb(output port , input port) - Magnitude of Large Signal S-Parameters in dB (i.e. '=LargeSdb(2,1)')

=LargeSAng(output port , input port) - Angle of Large Signal S-Parameters in degrees (i.e. '=LargeSAng(2,1)')

=LargeS(output port , input port) - Rectangular value of Large Signal S-Parameters (i.e. '=LargeS(2,1)')

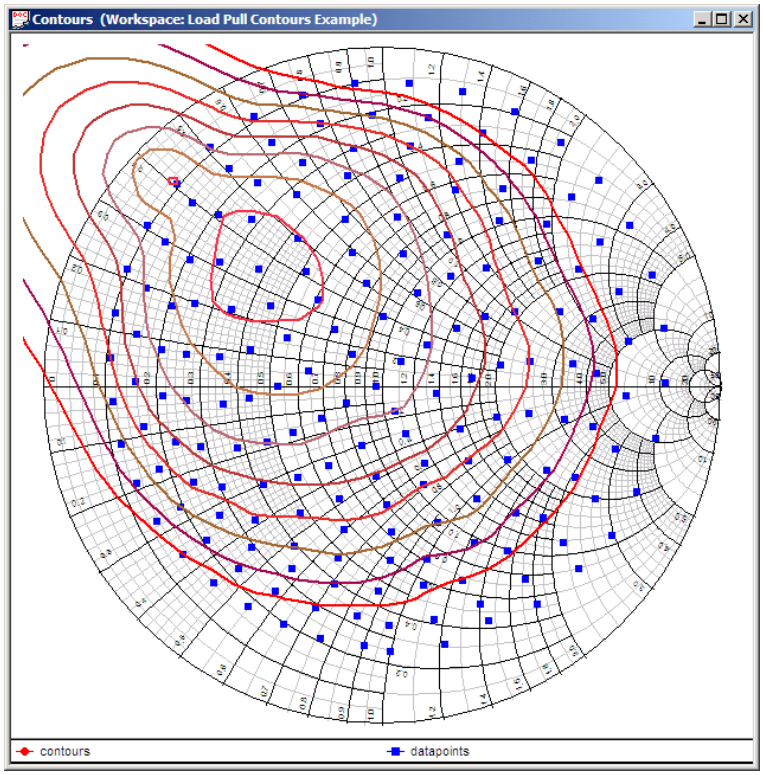
Steps for Large signal S-Parameter Analysis

1. Create a schematic with a PAC (AC Power) input at the input port. Make sure source frequencies and power levels have been specified.
2. Create variables that are intended to be swept (i.e. frequency, power, etc.) in the equations window.
3. Add a HarBEC simulation.
4. Add parameters sweeps of the desired variables, such as frequency or power. These sweeps are added under the 'Simulation/Data' workspace folder.
5. Add a graph. Graphs are added under the 'Outputs' workspace folder.
6. In the graph, select the correct sweep to use, then type the measurement, such as **=LargeS(2,1)** for S21. Currently, the measurement wizard cannot be used to add large signal S-Parameters.

For an example of Large Signal S-Parameters, choose File/Open Example, then load "Amplifiers/Large Signal S Parameters".

Load Pull Contours

GENESYS can draw load pull data contours (such as the one shown below) from data contained in Focus and Maury Microwave load pull data files.



This example (File/Open Example/Load Pull Contours Example.wsp) loads a Focus Microwaves data file. The file contains several columns of amplifier measured data. In the plot, the Gain column from the data file is used to plot load/pull contours from the two Equations shown below:

contours = CONTOUR(.GAIN,20,30,1,0,-2,2,-2,2)

datapoints = PLOTPOINTS(.GAIN)

The first equation, CONTOUR, generates the contours based on the parameters passed to the function. See Built In Functions for a description of these functions and their parameters.

GENESYS supports both Maury Microwave and Focus Microwaves data files.

To create a new file using load pull contours:

1. From a new GENESYS file, right-click Simulations on the tree and select Link to Data file.
2. After choosing a name, select the appropriate file type (Maury or Focus) and browse for or enter the file name. Click OK to close this box.
3. Add a Smith Chart by right-clicking Outputs.
4. After choosing a name, click "measurement wizard".
5. Select the load pull data simulation from the first dialog.
6. From the second measurement wizard box, select "Contours" or "Plotpoints" from the first column. Select the data to plot from the second column. Click OK to close the measurement wizard box and click OK to close the Smith Chart properties box.

You should then see contours like shown above.

Adjacent Channel Power (ACP[U or L][n])

This measurement is the integrated power of the specified adjacent channel. All adjacent channels are relative to the main channel (identified by the 'Channel Frequency' and 'Channel Measurement Bandwidth'). Consequently, channels exist above and below the main reference channel frequency. The user can specify which side of the main channel the adjacent channel is located on along with the channel number. The channel number is relative to the main channel. Therefore, channel 1 would be the first adjacent channel, channel 2 would be the second adjacent channel, and so on.

U - Upper Side

L - Lower Side

n - Channel Number (any integer > 0)

For example, ACPL2 is the power of the second adjacent channel below that specified by the channel frequency. If CF was 100 MHz and the channel bandwidth was 1 MHz then the main channel would be 99.5 to 100.5 MHz. Consequently, then ACPL2 would then be the integrated channel power between 97.5 and 98.5 MHz and ACPL1 would be the integrated channel power between 98.5 and 99.5 MHz.

Note: Only the first 2 adjacent channels on either side of the reference channel are listed in the 'Measurement Wizard". However, there is no restriction on the Adjacent Channel Number other than it must be non-negative and greater than or equal to 1.

See the 'Channel Power' measurement to determine which types of signals are included or ignored in this measurement.

This measurement is simply a 'Channel Power' measurement at the 'Adjacent Channel Frequency'.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[ACPU2]	2nd upper adjacent channel power in dBm	Real
MAG[ACPU2]	magnitude of the 2nd upper adjacent channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[ACPU2]	DBM[ACPU2]	DBM[ACPU2]
MAG[ACPU2]	MAG[ACPU2]	MAG[ACPU2]

* Not available on Smith Chart

Adjacent Channel Frequency (ACF[U or L][n])

This measurement is the frequency of the specified adjacent channel. All adjacent channel frequencies are relative to the main 'Channel Frequency'. Consequently, channels exist above and below the main reference channel frequency. The user can specify which side of the main or reference channel that the adjacent channel is located on and also the channel number. The channel number is relative to the main or reference channel. Therefore, channel 1 would be the first adjacent channel, channel 2 would be the second adjacent channel, and so on.

U - Upper Side

L - Lower Side

n - Channel Number (any integer > 0)

For example, ACFU1 if the first adjacent channel above that specified by the 'Channel Frequency'. If CF was 100 MHz and the channel bandwidth was 1 MHz then the main channel would be 99.5 to 100.5 MHz. Consequently, then ACFU1 would then be the channel 100.5 to 101.5 MHz and ACFL1 would be 98.5 to 99.5 MHz.

NOTE: Only the first 2 adjacent channels on either side of the reference channel is listed in the 'Measurement Wizard". However, there is no restriction on the Adjacent Channel Number.

Values: Real value in MHz.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
ACFL1	ACFL1	ACFL1

* Not available on Smith Chart

Added Noise (AN)

This measurement is the noise contribution of each individual stage in the main channel along the specified path as shown by:

AN[n] = CNF[n] - CNF[n-1] (dB), where AN[0] = 0 dB, n = stage number

This measurement is simply the difference in the 'Cascaded Noise Figure' measurement between the current node and the previous node. This measurement is very useful and will help the user identify the contribution to the noise figure by each stage along the path.

> See the 'Cascaded Noise Figure' measurement to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[AN]	stage noise figure in dB	Real
MAG[AN]	numeric value of the stage noise figure	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[AN]	DB[AN]	DB[AN]
MAG[AN]	MAG[AN]	MAG[AN]

* Not available on Smith Chart

Cascaded Gain (CGAIN)

This measurement is the cascaded gain of the main channel along the specified path. The 'Cascaded Gain' is the difference between the 'Desired Channel Power' measurement at the nth stage minus the 'Desired Channel Power' measurement at the input as shown by:

CGAIN[n] = DCP[n] - DCP[0] (dB), where n = stage number

The main channel is defined by the 'Channel Frequency' for the selected path and the system analysis 'Channel Measurement Bandwidth'.

> See the 'Desired Channel Power' measurement to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[CGAIN]	cascaded gain in dB	Real

MAG[CGAIN]	numeric value of the cascaded gain	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[CGAIN]	DB[CGAIN]	DB[CGAIN]
MAG[CGAIN]	MAG[CGAIN]	MAG[CGAIN]

* Not available on Smith Chart

Cascaded Gain [Third Order Intermod Analysis] (CGAINIM3)

This measurement is the cascaded gain of the main channel during the IM3 analysis. The 'Cascaded Third Order Intermod Gain' is the difference between the 'Desired Third Order Intermod Channel Power' at the nth stage and the 'Desired Third Order Intermod Channel Power' at the input as shown by:

CGAINIM3[n] = DCPIM3[n] - DCPIM3[0] (dB), where n = stage number

NOTE: This measurement is used by the IIP3, OIP3, and SFDR measurements.

The 'Calculate Intermods Along Path' checkbox must be checked and properly configured in order to make this measurement. See the 'Calculate Intermods Along Path section for information on how to configure these tests.

See the 'Desired Channel Power [Third Order Intermod Analysis]' measurement to determine which types of signals are included or ignored in this measurement.

The only difference between this measurement and the 'Cascaded Gain' (CGAIN) measurement is that this measurement applies to the IM3 analysis pass only. Consequently, this will be the same measurement as CGAIN in the Calculate Intermods Along Path Manual Mode since a dedicated IM3 analysis is not created and the normal analysis is also the IM3 analysis pass.

Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[CGAINIM3]	cascaded third order intermod gain in dB	Real

MAG[CGAINIM3]	numeric value of the cascaded third order intermod gain	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[CGAINIM3]	DB[CGAINIM3]	DB[CGAINIM3]
MAG[CGAINIM3]	MAG[CGAINIM3]	MAG[CGAINIM3]

* Not available on Smith Chart

Cascaded Gain - All Signals (CGainAll)

This measurement is the cascaded gain of the main channel along the specified path. The 'Cascaded Gain' is the difference between the 'Channel Power' measurement at the nth stage minus the 'Channel Power' measurement at the input as shown by:

CGAIN[n] = CP[n] - CP[0] (dB), where n = stage number

The main channel is defined by the 'Channel Frequency' for the selected path and the system analysis 'Channel Measurement Bandwidth'.

See the 'Channel Power' measurement to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[CGAINALL]	cascaded gain in dB	Real
MAG[CGAINALL]	numeric value of the cascaded gain	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[CGAINALL]	DB[CGAINALL]	DB[CGAINALL]
MAG[CGAINALL]	MAG[CGAINALL]	MAG[CGAINALL]

* Not available on Smith Chart

Carrier to Noise Ratio (CNR)

This measurement is the ratio of the 'Desired Channel Power' to 'Channel Noise Power' along the specified path as shown by:

CNR[n] = DCP[n] - CNP[n] (dB), where n = stage number

Both the 'Desired Channel Power' and 'Channel Noise Power' measurements use the main channel.

Note: See the 'Desired Channel Power' and 'Channel Noise Power' measurements to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[CNR]	carrier to noise ratio in dB	Real
MAG[CNR]	numeric value of the carrier to noise ratio	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[CNR]	DB[CNR]	DB[CNR]
MAG[CNR]	MAG[CNR]	MAG[CNR]

* Not available on Smith Chart

Cascaded Noise Figure (CNF)

This measurement is the cascaded noise figure in the main channel along the specified path. The 'Cascaded Noise Figure'is equal to the 'Channel Noise Power' measurement at the output of stage n minus the 'Channel Noise Power' measurement at the path input and the 'Cascaded Gain' measurement at stage n as shown by:

CNF[n] = CNP[n] - CNP[0] - CGAIN[n] (dB), where n = stage number

Note: See the 'Channel Noise Power' and 'Cascaded Gain' measurements to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[CNF]	cascaded noise figure in dB	Real
MAG[CNF]	numeric value of the cascaded noise figure	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[CNF]	DB[CNF]	DB[CNF]
MAG[CNF]	MAG[CNF]	MAG[CNF]

* Not available on Smith Chart

Channel (or Path) Frequency (CF)

Since each spectrum can contain a large number of spectral components and frequencies SPECTRASYS must be able to determine the area of the spectrum to integrate for various measurements. This integration area is defined by a 'Channel Frequency' and a 'Channel Measurement Bandwidth' which become the **main channel** for the specified path. SPECTRASYS can automatically identify the desired 'Channel Frequency' in an unambiguous case where only one frequency is on the 'from node' of the designated path. An error will appear if more than one frequency is available. For this particular case the user must specify the intended frequency for this path in the 'System Simulation Dialog Box'.

A 'Channel Frequency' exists for each node along the specified path. Consequently, each node along the path will have the same 'Channel Frequency' until a frequency translation element such as a mixer or frequency multiplier is encountered. SPECTRASYS automatically deals with frequency translation through these elements. The individual mixer parameters of 'Desired Output (Sum or Difference)' and 'LO Injection (High of Low)' are used to determine the desired frequency at the output of the mixer.

The 'Channel Frequency' is a critical parameter for SPECTRASYS since most of the measurements are based on this parameter. If this frequency is incorrectly specified then the user may get unexpected results since many measurements are based on this frequency.

The easiest way to verify the 'Channel Frequency' that SPECTRASYS is using is to look at this measurement in a Table.

Values: Real value in MHz.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
CF	CF	CF

* Not available on Smith Chart

Offset Channel Frequency (OCF)

The 'Offset Channel Frequency' and 'Offset Channel Power' are very useful measurements in SPECTRASYS. These measurements give the user the ability to create a user defined channel relative the the main channel. The user specifies both the 'Offset Frequency' relative to the main 'Channel Frequency' and the 'Offset Channel Bandwidth'. As with the 'Channel Frequency' measurement SPECTRASYS automatically deals with the frequency translations of the 'Offset Channel Frequency' through frequency translations elements such as mixers and frequency multipliers. Both the 'Offset Frequency' and the 'Offset Channel Bandwidth' can be tuned by simply placing a question mark in front of the value to be tuned. This measurement simply returns the 'Offset Channel Frequency' for every node along the specified path.

Values: Real value in MHz.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
OCF	OCF	OCF

* Not available on Smith Chart

Tone Channel Frequency (TCF)

This measurement is the frequency of the tone channel used for intermod measurements such as: IIP3, OIP3, SFDR, etc.. The 'Tone Channel Frequency' is determined by adding the offset specified in the 'Signal used for IIP3/OIP3' system analysis parameter to the 'Channel Frequency' for the given path.

The 'Calculate Intermods Along Path' option must be enabled to make this measurement.

As with other frequency measurements SPECTRASYS is able to deal with frequency translation through mixers, frequency multipliers, etc.

Values: Real value in MHz.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
TCF	TCF	TCF

* Not available on Smith Chart

Channel Noise Power (CNP)

This measurement is the integrated noise power in the main channel along the specified path.

For example, if the 'Channel Measurement Bandwidth' was specified to .1 MHz and the 'Channel Frequency' was 2000 MHz then the CNP is the integrated noise power from 1999.95 to 2000.05 MHz.

> This measurement includes ONLY NOISE traveling in FORWARD path direction through the node that fall within the main channel.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[CNP]	channel noise power in dBm	Real
MAG[CNP]	magnitude of the channel noise power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[CNP]	DBM[CNP]	DBM[CNP]
MAG[CNP]	MAG[CNP]	MAG[CNP]

* Not available on Smith Chart

Channel Power (CP)

This measurement is the total integrated power in the main channel (identified by the 'Channel Frequency' and the 'Channel Measurement Bandwidth') of the specified path.

> This measurement includes ALL SIGNALS, INTERMODS, HARMONICS, and NOISE traveling in ALL directions through the node that fall within the main channel.

For example, if the 'Channel Measurement Bandwidth' was specified to .03 MHz and the 'Channel Frequency' was 220 MHz then the CP is the integrated power from 219.985 to 220.015 MHz.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[CP]	channel power in dBm	Real
MAG[CP]	magnitude of the channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[CP]	DBM[CP]	DBM[CP]
MAG[CP]	MAG[CP]	MAG[CP]

* Not available on Smith Chart

Desired Channel Power (DCP)

This measurement is the total integrated power in the main channel (identified by the 'Channel Frequency' and the 'Channel Measurement Bandwidth') of the specified path.

> This measurement includes ONLY SIGNALS ORIGINATING on the beginning node of the path, traveling in FORWARD path direction through the node that fall within the main channel. All other intermods, harmonics, and noise is ignored in the forward path direction. All signals, intermods, harmonics, and noise are ignored for the reverse path direction.

For example, if the 'Channel Measurement Bandwidth' was specified to .03 MHz and the 'Channel Frequency' was 220 MHz then the DCP is the integrated power from 219.985 to 220.015 MHz. This power measurement will not even be affect by another 220 MHz signal traveling in the reverse direction even if it is much larger in amplitude.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[DCP]	desired channel power in dBm	Real
MAG[DCP]	magnitude of the desired channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[DCP]	DBM[DCP]	DBM[DCP]
MAG[DCP]	MAG[DCP]	MAG[DCP]

* Not available on Smith Chart

Desired Channel Power [Third Order Intermod Analysis] (DCPIM3)

This measurement is the desired channel power of the main channel during the IM3 analysis pass.

> **Note:** The 'Calculate Intermods Along Path)' checkbox must be checked and properly configured in order to make this measurement. See the 'Calculate Intermods Along Path' section for information on how to configure these tests.

See the 'Desired Channel Power' measurement to determine which types of signals are included or ignored in this measurement. The only difference between this measurement and the 'Desired Channel Power' (DCP) measurement is that this measurement applies to the IM3 analysis pass only. Consequently, this will be the same measurement as DCP in the Calculate Intermods Along Path Manual Mode since a dedicated IM3 analysis is not created and the normal analysis is also the IM3 analysis pass.

Remember intermod bandwidth is function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[DCPIM3]	desired third order intermod channel power in dBm	Real
MAG[DCPIM3]	magnitude of the desired third order intermod channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[DCPIM3]	DBM[DCPIM3]	DBM[DCPIM3]
MAG[DCPIM3]	MAG[DCPIM3]	MAG[DCPIM3]

* Not available on Smith Chart

Offset Channel Power (OCP)

The 'Offset Channel Frequency' and 'Offset Channel Power' are very useful measurements in SPECTRASYS. These measurements give the user the ability to create a user defined

channel relative the the main channel. The user specifies both the 'Offset Frequency' relative to the main 'Channel Frequency' and the 'Offset Channel Bandwidth'. As with the 'Channel Frequency' measurement SPECTRASYS automatically deals with the frequency translations of the 'Offset Channel Frequency' through mixers. Both the 'Offset Frequency' and the 'Offset Channel Bandwidth' can be tuned by simply placing a question mark in front of the value to be tuned. This measurement returns the integrated 'Offset Channel Power' for every node along the specified path.

For example, if the 'Channel Frequency' was 2140 MHz, 'Offset Channel Frequency' was 10 MHz, and the 'Offset Channel Bandwidth" was 1 MHz then the OCP is the integrated power from 2149.5 to 2150.5 MHz.

Tip: This measurement is simply a 'Channel Power' measurement at the 'Offset Channel Frequency' using the 'Offset Channel Bandwidth'.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[OCP]	offset channel power in dBm	Real
MAG[OCP]	magnitude of the offset channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[OCP]	DBM[OCP]	DBM[OCP]
MAG[OCP]	MAG[OCP]	MAG[OCP]

* Not available on Smith Chart

Tone Channel Power (TCP)

This measurement is the total integrated power in the tone channel. The tone channel is the 'Tone Channel Frequency' with the channel measurement bandwidth. This power is used for intermod measurements such as: IIP3, OIP3, SFDR, etc..

The 'Calculate Intermods Along Path' option must be enabled to make this measurement.

This measurement is simply a 'Channel Power' measurement at the 'Tone Channel Frequency'.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[TCP]	tone channel power in dBm	Real
MAG[TCP]	magnitude of the tone channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[TCP]	DBM[TCP]	DBM[TCP]
MAG[TCP]	MAG[TCP]	MAG[TCP]

* Not available on Smith Chart

Gain (GAIN)

This measurement is the individual stage gain of the main channel along the specified path. The 'Gain' is the difference between the 'Desired Channel Power' output of the current stage minus the 'Desired Channel Power' output of the prior stage as shown by:

GAIN[n] = DCP[n] - DCP[n-1] (dB), where GAIN[0] = 0 dB, n = stage number

See the 'Desired Channel Power' measurement to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[GAIN]	gain in dB	Real
MAG[GAIN]	numeric value of the gain	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[GAIN]	DB[GAIN]	DB[GAIN]
MAG[GAIN]	MAG[GAIN]	MAG[GAIN]

* Not available on Smith Chart

Gain [Third Order Intermod Analysis] (GAINIM3)

This measurement is the individual stage gain of the main channel along the specified path during the IM3 analysis pass. The 'Gain' is the difference between the 'Desired Channel

Power [Third Order Intermod Analysis]' output of the current stage minus the 'Desired Channel Power' output of the prior stage as shown by:

GAINIM3[n] = DCPIM3[n] - DCPIM3[n-1] (dB), where GAINIM3[0] = 0 dB, n = stage number

See the 'Desired Channel Power [Third Order Intermod Analysis]' measurement to determine which types of signals are included or ignored in this measurement.

The only difference between this measurement and the 'Gain' (GAIN) measurement is that this measurement applies to the IM3 analysis pass only. Consequently, this will be the same measurement as GAIN, when the 'manual' mode is used for calculating intermods along a path, since a dedicated IM3 analysis is not created and the normal analysis is also the IM3 analysis pass.

Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[GAIN]	gain in dB	Real
MAG[GAIN]	numeric value of the gain	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[GAIN]	DB[GAIN]	DB[GAIN]
MAG[GAIN]	MAG[GAIN]	MAG[GAIN]

* Not available on Smith Chart

Gain - All Signals (GAINALL)

This measurement is the individual stage gain of the main channel along the specified path. The 'Gain' is the difference between the 'Channel Power' output of the current stage minus the 'Channel Power' output of the prior stage as shown by:

GAIN[n] = CP[n] - CP[n-1] (dB), where GAIN[0] = 0 dB, n = stage number

---

See the 'Channel Power' measurement to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[GAINALL]	gain in dB	Real
MAG[GAINALL]	numeric value of the gain	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[GAINALL]	DB[GAINALL]	DB[GAINALL]
MAG[GAINALL]	MAG[GAINALL]	MAG[GAINALL]

* Not available on Smith Chart

Image Frequency (IMGF)

This measurement is the image frequency from the input to the first mixer. Any energy at the image frequency can seriously degrade the performance of a receiver. Even unfiltered noise at the image frequency will be converted into the IF band and degrade the sensitivity by as much as 3 dB. The image frequency measurements are provided to help the designer understand the impact of the image frequency on the performance of the receiver.

Since SPECTRASYS knows the 'Channel Frequency' of the specified path it also has the ability to figure out what the image frequency is up to the 1st mixer. After the 1st mixer the 'Image Frequency' measurement will show the main channel frequency. This measurement will show what that frequency is. This image frequency is used to determine the area of the spectrum that will be integrated by the 'Mixer Image Channel Power' measurement to calculate the image power.

For example if we designed a 2 GHz receiver that had an IF frequency of 150 MHz using low LO side injection then the LO frequency would be 1850 MHz and image frequency for all stages from the input to the first mixer would be 1700 MHz. All noise and interference must be rejected at this frequency to maintain the sensitivity and performance of the receiver.

Values: Real value in MHz.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators: None

Examples:

Measurement	Result in graph, Smith chart*,	Result on table

	optimization, or yield	
IMGF	IMGF	IMGF

* Not available on Smith Chart

Image Channel Noise Power (IMGNP)

This measurement is the integrated noise power of the image channel from the path input to the first mixer. After the first mixer the 'Mixer Image Channel Power' measurement will show the same noise power and the main channel noise power. This measurement is very useful in determining the amount of image noise rejection that the selected path provides.

Any energy at the image frequency can seriously degrade the performance of a receiver. Even unfiltered noise at the image frequency will be converted into the IF band and degrade the sensitivity by as much as 3 dB. The image frequency measurements are provided to help the designer understand the impact of the image frequency on the performance of the receiver.

Since SPECTRASYS knows the 'Channel Frequency' of the specified path it also has the ability to figure out what the image frequency is up to the 1st mixer. The 'Mixer Image Frequency' measurement will show what that frequency is. This image frequency is used to determine the area of the spectrum that will be integrated by the this measurement to calculate the image power. The 'Channel Measurement Bandwidth' located in the 'System Simulation Dialog Box' is used as the bandwidth for the this measurement.

For example if we designed a 2 GHz receiver that had an IF frequency of 150 MHz using low LO side injection then the LO frequency would be 1850 MHz and image frequency for all stages from the input to the first mixer would be 1700 MHz. If the receiver bandwidth was 5 MHz then the image channel would be from 1697.5 to 1702.5 MHz. All noise and interference must be rejected in this channel to maintain the sensitivity and performance of the receiver.

This measurement is simply a 'Channel Noise Power' measurement at the 'Image Frequency'.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[IMGNP]	mixer image channel power in dBm	Real
MAG[IMGNP]	magnitude of the mixer image channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*,	Result on table

	optimization, or yield	
DBM[IMGNP]	DBM[IMGNP]	DBM[IMGNP]
MAG[IMGNP]	MAG[IMGNP]	MAG[IMGNP]

* Not available on Smith Chart

Image Noise Rejection Ratio (IMGNR)

This measurement is the ratio of the 'Channel Noise Power' to 'Image Channel Noise Power' along the specified path as shown by:

IMGR[n] = CNP[n] - IMGNP[n] (dB), where n = stage number

This measurement is very useful in determining the amount of image noise rejection that the selected path provides.

For this particular measurement basically two channels exist both with the same 'Channel Measurement Bandwidth' 1) main channel and 2) 1st mixer image channel. The only difference is between these two channels are their frequencies, one is at the 'Channel Frequency' and the other is at the 'Mixer Image Frequency'.

See the 'Channel Noise Power' and 'Image Channel Noise Power' measurements to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[IMGNR]	mixer image rejection ratio in dB	Real
MAG[IMGNR]	numeric value of the mixer image rejection ratio	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[IMGNR]	DB[IMGNR]	DB[IMGNR]
MAG[IMGNR]	MAG[IMGNR]	MAG[IMGNR]

*** Not available on Smith Chart**

Minimum Detectable Signal (MDS)

This measurement is the minimum detectable (discernable) signal referred to the input and is equivalent to the input channel noise power plus the cascaded noise figure of the specified chain as shown by:

MDS[n] = CNP[0] + CNF[n] in dBm, where n = stage number

The MDS value at stage n represents the MDS of the entire system up to and including stage n. Consequently, the MDS of the entire system is the value indicated at the last stage in the path or chain. The minimum detectable signal is the equivalent noise power present on the input to a receiver that sets the limit on the smallest signal the receiver can detect.

For example, if the thermal noise power input to a receiver is -174 dBm/Hz and the channel bandwidth is 1 MHz (10 Log (1 MHz) = 60 dB) then the input channel power would be -114 dBm. For a cascaded noise figure of 5 dB the minimum detectable signal would be -109 dBm.

> See the 'Channel Noise Power' measurement to determine which types of signals are included or ignored in this measurement.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[MDS]	minimum detectable signal in dBm	Real
MAG[MDS]	magnitude of the minimum detectable signal in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[MDS]	DB[MDS]	DB[MDS]
MAG[MDS]	MAG[MDS]	MAG[MDS]

* Not available on Smith Chart

Image Channel Power (IMGP)

This measurement is the integrated power of the image channel from the path input to the first mixer. After the first mixer the 'Mixer Image Channel Power' measurement will show the same power and the main channel power. Any energy at the image frequency can seriously degrade the performance of a receiver. Even unfiltered noise at the image frequency will be converted into the IF band and degrade the sensitivity by as much as 3 dB. The image frequency measurements are provided to help the designer understand the impact of the image frequency on the performance of the receiver.

Since SPECTRASYS knows the 'Channel Frequency' of the specified path it also has the ability to figure out what the image frequency is up to the 1st mixer. The 'Mixer Image Frequency' measurement will show what that frequency is. This image frequency is used to determine the area of the spectrum that will be integrated by the this measurement to calculate the image power. The 'Channel Measurement Bandwidth' located in the 'System Simulation Dialog Box' is used as the bandwidth for the this measurement.

For example if we designed a 2 GHz receiver that had an IF frequency of 150 MHz using low LO side injection then the LO frequency would be 1850 MHz and image frequency for all stages from the input to the first mixer would be 1700 MHz. If the receiver bandwidth was 5 MHz then the image channel would be from 1697.5 to 1702.5 MHz. All noise and interference must be rejected in this channel to maintain the sensitivity and performance of the receiver.

This measurement is simply a 'Channel Power' measurement at the 'Image Frequency'.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[IMGP]	mixer image channel power in dBm	Real
MAG[IMGP]	magnitude of the mixer image channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[IMGP]	DBM[IMGP]	DBM[IMGP]
MAG[IMGP]	MAG[IMGP]	MAG[IMGP]

* Not available on Smith Chart

Image Rejection Ratio (IMGR)

This measurement is the ratio of the 'Channel Power' to 'Image Channel Power' along the specified path as shown by:

IMGR[n] = DCP[n] - IMGP[n] (dB), where n = stage number

For this particular measurement basically two channels exist both with the same 'Channel Measurement Bandwidth' 1) main channel and 2) 1st mixer image channel. The only difference is between these two channels are their frequencies, one is at the 'Channel Frequency' and the other is at the 'Mixer Image Frequency'.

See the 'Desired Channel Power' and 'Image Channel Power' measurements to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[IMGR]	mixer image rejection ratio in dB	Real
MAG[IMGR]	numeric value of the mixer image rejection ratio	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[IMGR]	DB[IMGR]	DB[IMGR]
MAG[IMGR]	MAG[IMGR]	MAG[IMGR]

* Not available on Smith Chart

Percent Noise Figure (PRNF)

This routine calculates the Percent Noise Figure Contribution by each stage to the final Cascaded Noise Figure of the path.

PRNF - Percent Noise Figure

AN - Added Noise

CNF - Cascaded Noise Figure

PRNF[n] = AN[n] / CNF[nLastStage] (this is a ratio in dB), Where PRNF[0] = 0, n is the current stage, and nLastStage is the last stage along the designated path.

This measurement will help the user pinpoint all stages and their respective contribution to the total cascaded noise figure of the selected path.

This measurement in unit-less since the measurement is a percentage. There can be a few cases where the percentage sum of all the stages in the path does not equal 100%. For instance, if the architecture contains parallel paths then each path would contribute to the total cascaded noise figure but only a single path is considered in this measurement. Another case would be where there are sufficient VSWR interactions between stages that effect the noise. Reducing the architecture to the spreadsheet case will always yield the expected spreadsheet answers with respect to percentages.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
PRNF	Stage Generated Noise Figure as a Percent	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
PRNF	PRNF	PRNF

* Not available on Smith Chart

Percent Third Order Intermod (PRIM3)

This routine calculates the Percent Third Order Intermod Contribution by each stage to the final Total Third Order Intermod Power of the path.

PRIM3 - Percent 3rd Order Intermod Power

GIM3P - Generated 3rd Order Intermod Power

CGAINIM3 - Cascade Gain for the Intermod Pass

TIM3P - Total 3rd Order Intermod Power

IM3REF - Equivalent 3rd Order Intermod Power Referenced to the Output

IM3REF = GIM3P[n] + (CGAINIM3[nLastStage] - CGAINIM3[n])

PRIM3[n] = IM3REF[n] / TIM3P[iLastStage](this is a ratio in Watts), Where PRIM3[0] = 0, n is the current stage, and nLastStage is the last stage along the designated path

This measurement will help the user pinpoint all stages and their respective contribution to the total third order intermod power of the selected path.

This measurement in unit-less since the measurement is a percentage. There can be a few cases where the percentage sum of all the stages in the path does not equal 100%. For instance, if the architecture contains parallel paths then each path would contribute to the total third order intermod power but only a single path is considered in this measurement. Another case would be where there are sufficient VSWR interactions between stages that effect the intermod levels. Reducing the architecture to the spreadsheet case will always yield the expected spreadsheet answers with respect to percentages. Sometimes this measurement can be greater than 100% if the equivalent 3rd order intermod power referenced to the output is greater than the actual total 3rd order intermod power. A good example of this would be an amplifier where intermods are cancelled at the amplifier output. In this case the generated intermod power alone may be much higher that the total intermod output power.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
PRIM3	Stage Generated IM3 as a Percent	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
PRIM3	PRIM3	PRIM3

* Not available on Smith Chart

Spurious Free Dynamic Range (SFDR)

This measurement is the spurious free dynamic range along the specified path as shown by:

SFDR[n] = 2/3 [IIP3[n] - MDS[n]] (dB), where n = stage number

The 'Spurious Free Dyanmic Range' is the range between the Minimum Detectable (Discernable) Signal (MDS) and the input power which would cause the third order intermods to be equal to the MDS. The MDS is the smallest signal that can be detected and will be equivalent to the receiver noise floor with a signal to noise ratio of 0 dB. In other words the MDS = -174 dBm/Hz + System Noise Figure + 10 Log(Channel Bandwidth).

See the 'Input Third Order Intercept' and 'Channel Noise Power' measurements to determine which types of signals are included or ignored in this measurement.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[SFDR]	spurious free dynamic range in dB	Real
MAG[SFDR]	magnitude of the spurious free dynamic range in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[SFDR]	DB[SFDR]	DB[SFDR]
MAG[SFDR]	MAG[SFDR]	MAG[SFDR]

* Not available on Smith Chart

Stage Dynamic Range (SDR)

This measurement along the specified path as shown by:

SDR[n] = SOP1DB[n] - TNP[n] (dB), where n = stage number

This simple measurement shows the difference between the 1 dB compression point of the stage and the 'Total Node Power' at the stage output. This measurement is extremely useful when trying to optimize each stage dynamic range and determine which stage that will go into compression first.

See the 'Stage Output 1 dB Compression Point' and 'Total Node Power' measurements to determine which types of signals are included or ignored in this measurement

---

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[SDR]	stage dynamic range in dB	Real
MAG[SDR]	numeric value of the stage dynamic range	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[SDR]	DB[SDR]	DB[SDR]
MAG[SDR]	MAG[SDR]	MAG[SDR]

* Not available on Smith Chart

Stage Noise Figure (SNF)

This measurement is the noise figure of each individual stage along the specified path as shown by:

SNF[n] = [CNP[n] - CNP[n-1]] - GAIN[n] (dB), where n = stage number (Passive Stages)

OR

SNF[n] = Noise Figure of the Active Stage (dB), where n = stage number (Amplifier and Mixer Stages)

The 'Stage Noise Figure' is the noise figure of each individual stage. For all passive devices this noise figure is based on the channel power and stage gain. However, for amplifier and mixer stages this noise figure will be the noise figure entered in the

parameters for these devices. This measurement is used to aid the user in determining the added noise by each stage in the cascade.

> See the 'Gain' and 'Channel Noise Power' measurements to determine which types of signals are included or ignored in this measurement.

Values: Real value (numeric).

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DB[SNF]	stage noise figure in dB	Real
MAG[SNF]	numeric value of the stage noise figure	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DB[SNF]	DB[SNF]	DB[SNF]
MAG[SNF]	MAG[SNF]	MAG[SNF]

* Not available on Smith Chart

Stage Output 1 dB Compression Point (SOP1DB)

This measurement is the output 1 dB compression point specified in the element parameters for the particular stage. This parameter is currently only available for the SPECTRASYS non-linear behavioral models such as amplifiers and mixers. For all stages where this parameter is not specified a large default value of +100 dBm is used.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[SOP1DB]	stage output 1 dB compression point in dBm	Real
MAG[SOP1DB]	numeric value of the stage output 1 dB compression point	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[SOP1DB]	DBM[SOP1DB]	DBM[SOP1DB]
MAG[SOP1DB]	MAG[SOP1DB]	MAG[SOP1DB]

* Not available on Smith Chart

Stage Output Second Order Intercept (SOIP2)

This measurement is the output second order intercept specified in the element parameters for the particular stage. This parameter is currently only available for the SPECTRASYS non-linear behavioral models such as amplifiers and mixers. For all stages where this parameter is not specified a large default value of +100 dBm is used.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[SOIP3]	stage output third order intercept in dBm	Real
MAG[SOIP3]	numeric value of the stage output third order intercept	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[SOIP3]	DBM[SOIP3]	DBM[SOIP3]
MAG[SOIP3]	MAG[SOIP3]	MAG[SOIP3]

* Not available on Smith Chart

Stage Output Third Order Intercept (SOIP3)

This measurement is the output third order intercept specified in the element parameters for the particular stage. This parameter is currently only available for the SPECTRASYS non-linear behavioral models such as amplifiers and mixers. For all stages where this parameter is not specified a large default value of +100 dBm is used.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[SOIP3]	stage output third order intercept in dBm	Real
MAG[SOIP3]	numeric value of the stage output third order intercept	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[SOIP3]	DBM[SOIP3]	DBM[SOIP3]

MAG[SOIP3]	MAG[SOIP3]	MAG[SOIP3]

* Not available on Smith Chart

Stage Output Saturation Power (SOPSAT)

This measurement is the output saturation power specified in the element parameters for the particular stage. This parameter is currently only available for the SPECTRASYS non-linear behavioral models such as amplifiers and mixers. For all stages where this parameter is not specified a large default value of +100 dBm is used.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[SOPSAT]	stage output saturation power in dBm	Real
MAG[SOPSAT]	numeric value of the stage output saturation power	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[SOPSAT]	DBM[SOPSAT]	DBM[SOPSAT]
MAG[SOPSAT]	MAG[SOPSAT]	MAG[SOPSAT]

* Not available on Smith Chart

Input Third Order Intercept (IIP3)

This measurement is the third order intercept point referenced to the path input along the specified path as shown by:

IIP3[n] = OIP3[n] - CGAINIM3[n] (dBm), where n = stage number

This measurement simple takes the computed 'Output Third Order Intercept' and references it to the input by subtracting the cascaded gain of the intermod path from the input to the current stage. The last IIP3 value for a cascaded chain will always be the actual input third order intercept for the entire chain.

This measurement is only available during the IM3 analysis pass.

The 'Calculate Intermods Along Path' checkbox must be checked and properly configured in order to make this measurement. See the 'Calculate Intermods Along Path' section for information on how to configure these tests.

See the 'Output Third Order Intercept' and 'Cascaded Third Order Intermod Gain' measurements to determine which types of signals are included or ignored in this measurement.

Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[IIP3]	input third order intercept in dBm	Real
MAG[IIP3]	magnitude of the input third order intercept in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[IIP3]	DBM[IIP3]	DBM[IIP3]
MAG[IIP3]	MAG[IIP3]	MAG[IIP3]

* Not available on Smith Chart

Output Third Order Intercept (OIP3)

This measurement is the third order intercept point referenced to the output along the specified path as shown by:

OIP3[n] = Virtual Tone Power[n] + Delta[n] / 2 (dBm), where n = stage number

Virtual Tone Power[n] = TCP[0] + CGAINIM3[n]

Delta[n] = Virtual Tone Power[n] - TIM3P[n]

Delta is the difference in dB between the 'Total Third Order Intermod Power' in the main channel and the interfering signal present in the 'Tone Channel'. The intercept point is equivalent to 1/2 Delta added to the interfering tone power level. Cascaded intermod equations are not used in SPECTRASYS. In order to correctly calculate OIP3 due to out-of-band interferers a Virtual Tone is created whose virtual power is that of an un-attenuated in-band tone. This power level is simply the 'Tone Channel Power' at the input plus the 'Cascaded Third Order Intermod Gain' at the current stage.

This Virtual Tone Power is different than the 'Tone Channel Power' measurement because the Virtual Tone Power is not attenuated by out-of-band rejection whereas the 'Tone Channel Power' is. For in-band interferers the Virtual Tone Power and the 'Tone Channel Power' measurement will be identical.

In order to make this measurement a minimum of three signals (tones) must actually be present at the input port, 1) main channel signal, 2) first interfering signal (tone), and 3)

second interfering signal (tone). Furthermore, the spacing of the two interfering tones needs to be such that intermods will actually fall into the main or primary channel. If these conditions are not met then no intermod power will be measured in the main channel.

This measurement is only available during the IM3 analysis pass.

> **Note:** The 'Calculate Intermods Along Path' checkbox must be checked and properly configured in order to make this measurement. See the 'Calculate Intermods Along Path' section for information on how to configure these tests.

Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[OIP3]	output third order intercept in dBm	Real
MAG[OIP3]	magnitude of the output third order intercept in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[OIP3]	DBM[OIP3]	DBM[OIP3]
MAG[OIP3]	MAG[OIP3]	MAG[OIP3]

* Not available on Smith Chart

Conducted Third Order Intermod Power (CIM3P)

This measurement is the integrated total intermod power in the main channel conducted from the prior stage during the IM3 analysis pass. Only Intermod signals are used for this measurement. All other types of signal are ignored. This measurement will include all intermods that are traveling in the forward path direction. In equation for the conducted third order intermod power is:

CIM3P[n] = TIM3P[n-1] + GAINIM3[n], where CIM3P[0] = 0 dB and n = stage number

Using this measurement in conjunction with the 'Generated Third Order Intermod Power (GIM3P)' and the 'Total Third Order Intermod Power (TIM3P)' the user can quickly

identify the weak intermod link in the cascaded chain and will guide the user in maximizing the Spurious Free Dynamic Range (SFDR).

> **Note:** The 'Calculate Intermods Along Path' checkbox must be checked and properly configured in order to make this measurement. See the 'Calculate Intermods Along Path' section for information on how to configure these tests.

See the 'Total Third Order Intermod Power' and 'GainIM3' measurements to determine which types of signals are included or ignored in this measurement.

When calculating intermods along a path in the 'Manual' mode a dedicated IM3 analysis is not created and the normal analysis is also the IM3 analysis pass.

Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[CIM3P]	conducted third order intermod power in dBm	Real
MAG[CIM3P]	magnitude of the conducted third order intermod power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[CIM3P]	DBM[CIM3P]	DBM[CIM3P]
MAG[CIM3P]	MAG[CIM3P]	MAG[CIM3P]

* Not available on Smith Chart

Generated Third Order Intermod Power (GIM3P)

This measurement is the generated intermod power in the main channel created at the output of the current stage during the IM3 analysis pass.In equation form the generated third order intermod power is:

GIM3P[n] = integration of the intermods generated at stage n across the channel bandwidth

Using this measurement in conjunction with the 'Conducted Third Order Intermod Power (CIM3P)' and the 'Total Third Order Intermod Power (TIM3P)' the user can quickly

identify the weak intermod link in the cascaded chain and will guide the user in maximizing the Spurious Free Dynamic Range (SFDR).

> **Note:** The 'Calculate Intermods Along Path' checkbox must be checked and properly configured in order to make this measurement. See the 'Calculate Intermods Along Path' section for information on how to configure these tests.

See the 'Desired Channel Power [Third Order Intermod Analysis]' measurement to determine which types of signals are included or ignored in this measurement.

When calculating intermods along a path in the 'Manual' mode a dedicated IM3 analysis is not created and the normal analysis is also the IM3 analysis pass.

Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[GIM3P]	generated third order intermod power in dBm	Real
MAG[GIM3P]	magnitude of the generated third order intermod power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[GIM3P]	DBM[GIM3P]	DBM[GIM3P]
MAG[GIM3P]	MAG[GIM3P]	MAG[GIM3P]

* Not available on Smith Chart

Total Third Order Intermod Power (TIM3P)

This measurement is the integrated total intermod power conducted from the prior stage plus the intermod power generated by the current stage during the IM3 analysis pass. All Intermod power is integrated across the main channel for the specified path. This measurement will include intermod power from all paths and all sources at the prior node as well as the current node if those intermods fall within the channel. In equation form the conducted third order intermod power is:

TIM3P[n] = integration of the total intermod spectrum at stage n across the main channel

Using this measurement in conjunction with the 'Conducted Third Order Intermod Power' and the 'Generated Third Order Intermod Power' the user can quickly identify the weak intermod link in the cascaded chain and will guide the user in maximizing the Spurious Free Dynamic Range.

> **Note:** The 'Calculate Intermods Along Path' checkbox must be checked and properly configured in order to make this measurement. See the 'Calculate Intermods Along Path' section for information on how to configure these tests.

When calculating intermods along a path in the 'Manual' mode a dedicated IM3 analysis is not created and the normal analysis is also the IM3 analysis pass.

Remember intermod bandwidth is a function of the governing intermod equation. For example, if the intermod equation is 2F1 - F2 then the intermod bandwidth would be: 2BW1 + BW2. Note: Bandwidths never subtract and will always add. The channel bandwidth must be set wide enough to include the entire bandwidth of the intermod to achieve the expected results. The 'Automatic Intermod Mode' will set the bandwidth appropriately.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[TIM3P]	total third order intermod power in dBm	Real
MAG[TIM3P]	total third order intermod power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[TIM3P]	DBM[TIM3P]	DBM[TIM3P]
MAG[TIM3P]	MAG[TIM3P]	MAG[TIM3P]

* Not available on Smith Chart

Total Node Power (TNP)

This measurement is the integrated power of the entire spectrum at the node. This is an extremely useful measurement in determining the total power present at the input of a device (i.e. amplifier or mixer LO). This measurement includes ALL SIGNALS, INTERMODS, HARMONICS, and NOISE traveling in ALL directions through the node.

Values: Real value in Watts.

Simulations: SPECTRASYS

Default Format: Table: Linear Graph: Linear Smith Chart: (none)

Commonly Used Operators:

Operator	Description	Result Type
DBM[TNP]	total node power in dBm	Real
MAG[TNP]	magnitude of the channel power in Watts	Real

Examples:

Measurement	Result in graph, Smith chart*, optimization, or yield	Result on table
DBM[TNP]	DBM[TNP]	DBM[TNP]
MAG[TNP]	MAG[TNP]	MAG[TNP]

* Not available on Smith Chart

Chapter 18 EMPOWER: Operation

Overview

An EMPOWER simulation requires a board layout description. The easiest (and recommended) method is to use the LAYOUT program to create a graphical representation of the desired layout pattern. The board can then be simulated by creating an EMPOWER Simulation.

This chapter describes how to use the LAYOUT program to construct a board layout and obtain an EMPOWER simulation. GENESYS is then used to display and compare the linear simulation with the EMPOWER data.

Features

EMPOWER incorporates many features still not present in competitive, late generation, EM simulators. Principle features include:

- Benchmarked accuracy
- Easy to use graphical circuit layout editor
- Complete integration with the GENESYS circuit simulation, synthesis and layout tools
- Multilayer simulations (with EMPOWER ML)
- Automatic incorporation of lumped elements
- Automatic detection and solution with symmetry
- Generalized S-parameter support
- Multi-mode support for ports and lines
- Tuning of EM objects in GENESYS using decomposition
- Deembedded or non-deembedded ports
- Viaholes including generated fields
- Any number of dielectric layers
- Dielectric and metal loss
- Includes box modes and package effects
- Slot-mode for slot and coplanar circuits
- Thick metal simulation (with EMPOWER ML)
- 32-bit code for Windows 95/98/NT

Examples

The examples are completely contained in the EXAMPLES manual. Examples which illustrate EMPOWER include:

- Microstrip Line.WSP
- Stripline Standard.WSP
- Spiral Inductor 2.WSP
- Box Modes.WSP
- Film Atten.WSP
- Edge Coupler.WSP
- Dual Mode.WSP
- 8 Way.WSP
- Edge Coupled.WSP
- Coupled Stepped Z.WSP
- Tuned Bandpass.WSP
- Patch Antenna Impedance.WSP

The required RAM specified in the Examples manual is the value estimated by EMPOWER. They are approximate and are determined by algorithm rather than a test of memory used. The execution times are for a 266 MHz Pentium II with 256Mbytes of RAM operating under Windows 98. In most cases execution time is for the discontinuity mode.

Creating a Layout

A board layout can be created one of two ways:

- By starting without a schematic
- By starting from an existing schematic

The first method starts in the GENESYS Environment by creating a layout without an associated schematic. The layout is created by drawing lines and placing footprints in the LAYOUT editor.

The second method begins in with a schematic and creates a board layout based on the schematic objects. This method is normally used when a linear simulation (using GENESYS) has been performed on a schematic and an EMPOWER simulation is desired, or when any lumped elements are needed in the EMPOWER Simulation. In addition to the schematic objects, any desired LAYOUT objects can be added to the board before simulation. For example, linear simulation would normally not include

ground pours, power supply rails, and lumped element pads. However, these are included in the EMPOWER run, allowing inspection of their effects.

Creating a Layout Without a Schematic

The complete file from this example is LayoutOnly.WSP

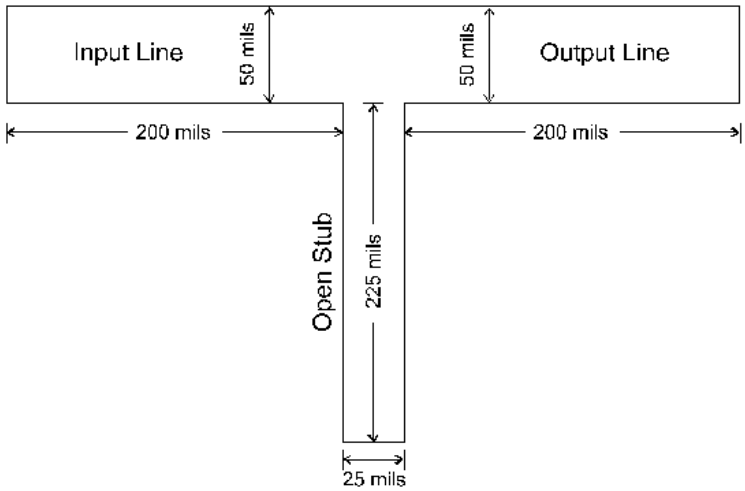
This example demonstrates the following topics:

- Creating a layout without a schematic
- Choosing grid spacings
- Choosing the box size

A microstrip stub notch filter with a transmission zero at 9.5 GHz is to be simulated. The filter has the following specifications:

- 15 mil RT/Duroid substrate (er=2.2, tan d=0.0009)
- Copper metalization
- 50 W terminations
- The stub line should be 70 W and 90° at 9.5 GHz

The series lines and the stub dimensions were calculated using T/LINE, and were rounded to the nearest 5 mil increment. The final line dimensions are shown below.



Note: Before beginning this example, you should be sure your Workspace Window is visible. Select Workspace Window from the View menu if necessary.

To begin, select New from the GENESYS File menu. Since we do not need a schematic for this circuit, we will delete the schematic: In the workspace window, Right-click on "Sch1 (Schematic)" and Select Delete This Design". Next, we will create a layout. Right-

click on “Designs” in the workspace window and select Add Layout from the LAYOUT menu. Enter “Stub” for the layout name. The “Create New Layout” dialog appears.

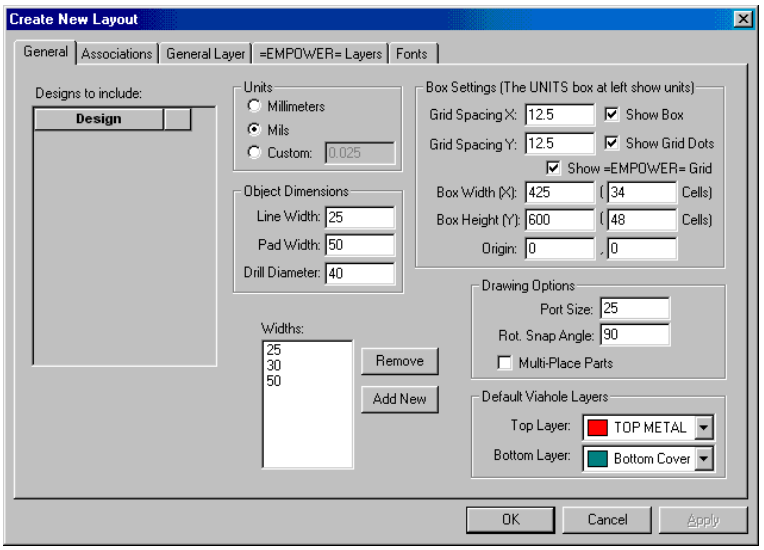
The tabs and prompts on this dialog are described in detail in the Basics section.

Note: For all dialog boxes, be sure that your screen looks exactly like the boxes shown in the figures.

Box Dimensions

Note: In EMPOWER, the layout’s box dimensions are used to define the bounding box.

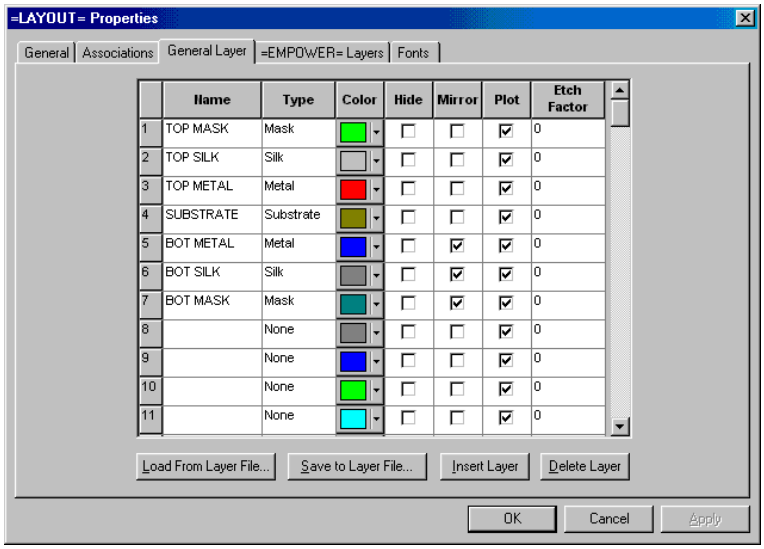
The box dimensions are shown below. Box Width was chosen as 425, the width of the filter since there are two 200 mil lines and a stub width of 25 mils. The filter height is 275 mils, including the stub length and series line width. The box height was chosen as 600 mils to give plenty of spacing on either side of the filter. This minimizes wall interference in the filter’s frequency response.



The EMPOWER grid settings for this example are shown in the upper right above. EMPOWER simulation time is greatly reduced if dimensions are chosen so that metal lies exactly on as large a grid size as possible. The grid width and height settings for this filter were chosen as 12.5, since the filter dimensions (425x275) are exactly divisible by this value.

General Layers

The general layer settings for this example are shown below.



Only three layers had to be defined for this filter:

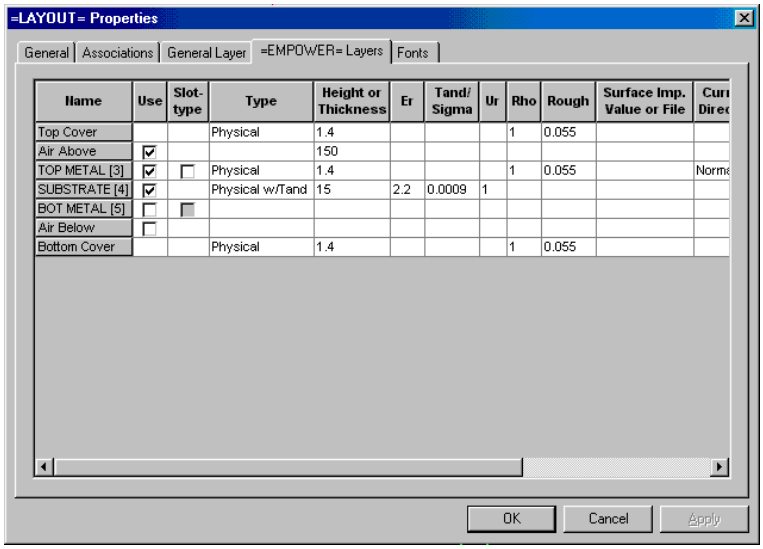
- Top Metal
- Substrate
- Bottom Metal

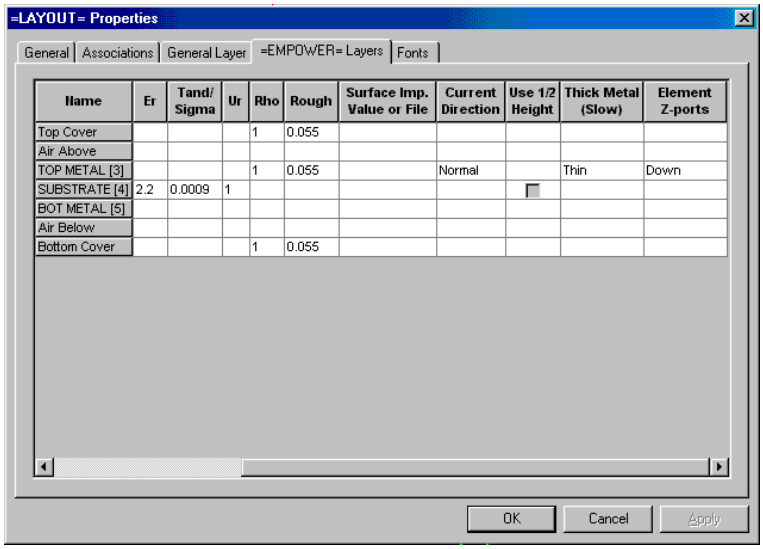
These are the only layers that are needed to simulate the microstrip filter. For a general layout, more layers are often included for purposes only. For example, defining a silk screen or mask layer would not affect simulation since none of the filter metal is placed on those layers.

> **Note:** Since the bottom of the box will be used as a ground plane, the bottom metal layer defined above may not necessary be necessary. However, since it is often necessary for manufacturing reasons, it is normally defined here.

EMPOWER Layers

The EMPOWER layers for this example are shown below. The EMPOWER layers are automatically selected from the available general layers (see the previous section). They are chosen from the available metal and substrate layers, and can be enabled or disabled for EMPOWER simulation.



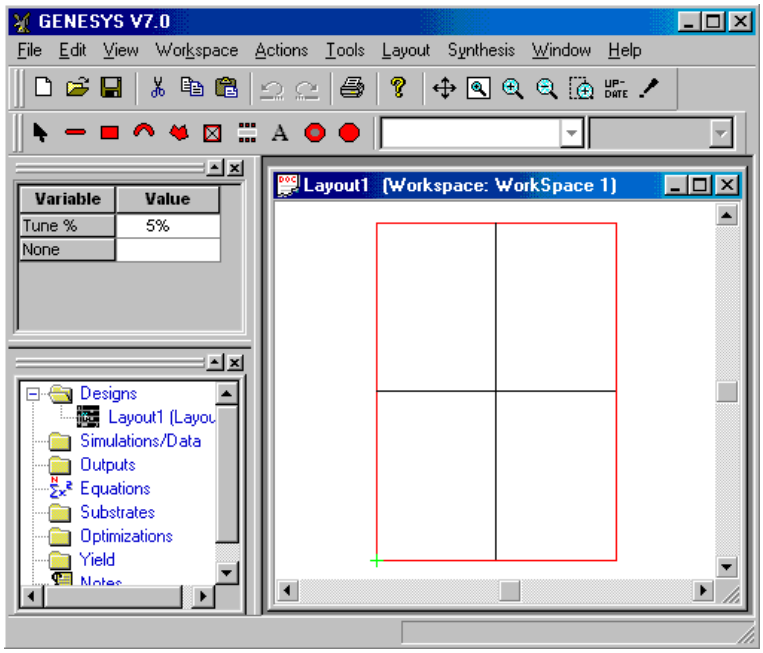


Since Air layers above and below a substrate are so common, a special option has been given here to add them. For more information on the individual layer options, see the EMPOWER Basics section.

Notice that BOT METAL and Air Below are not enabled. This places the box bottom at the lower substrate boundary so that it acts as a ground plane.

> **Note:** In almost all cases where a completely solid ground plane is used, you should use the top or bottom cover to simulate it. This is much more efficient than using an extra metal layer.

Click OK. The LAYOUT editor appears. The screen should look like similar to:

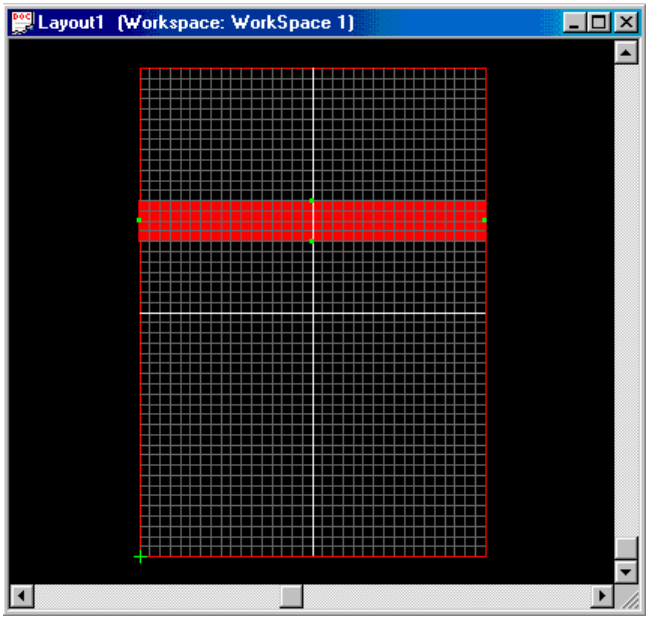


Drawing the Layout

To draw the series line:

1. Select the Rectangle button from the LAYOUT toolbar This is the third button on the bottom toolbar.
2. Click on the left edge of the page border, and drag toward the right and down until the status bar shows DX=425 and DY=50.
3. Release the mouse button.

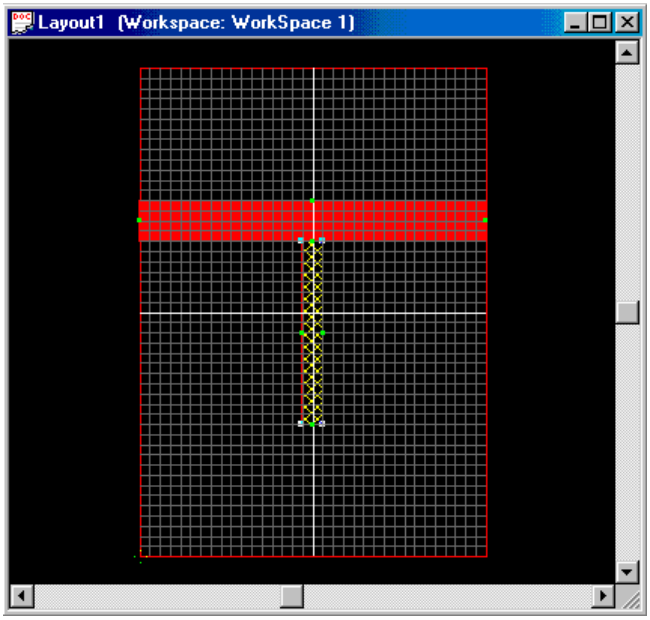
This is the series transmission line. The screen should now look as below. Don't worry if the line isn't at the exact same position on the page - the layout will be centered later.



To draw the open stub:

1. Select the Rectangle button from the toolbar.
2. Click at the bottom edge of the line just drawn, one grid cell left of the series line's center.
3. Drag to the right and down until the status bar shows DX=25 and DY=225.
4. Release the mouse button.

The screen should now look like the following. If the stub line isn't centered horizontally on the screen, select the stub by clicking on it, and drag it to the proper position.



Centering the Layout

As a general rule, EMPOWER simulation time is greatly reduced if the circuit to be simulated exhibits symmetry in any of several planes. Many circuits will exhibit some form of symmetry if they are centered in the page area.

To center the example filter:

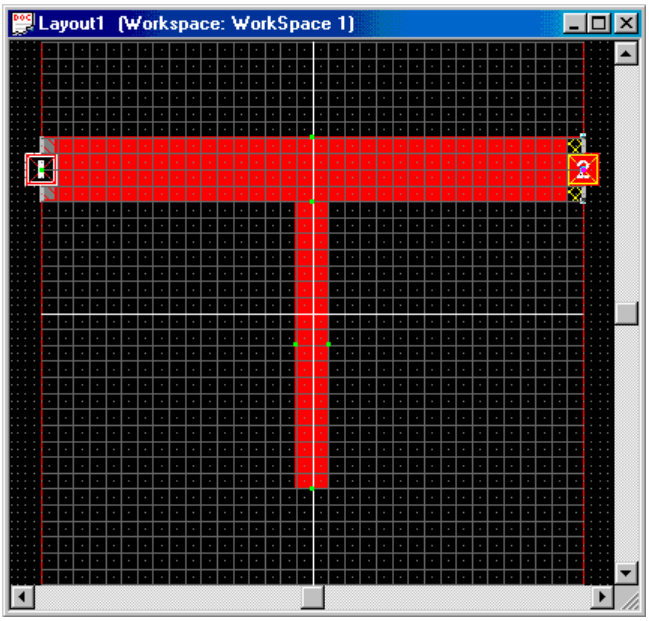
1. Choose Select All from the Edit menu.
2. Choose Center Selected On Page from the Layout menu.

Placing EMPOWER Ports

Before running EMPOWER, the filter's ports must be designated. Select the EMPort button (⊠) on the LAYOUT toolbar, and click on the center left end of the series line. The EM Port Properties dialog appears. Set the drawing size to 25. This controls how large the ports will be drawn on the LAYOUT screen. Note that the default port number is 1. Select the OK button.

Next, select the EMPort button on the toolbar again. Click on the center right end of the series line. The EM Port Properties dialog appears. Again, type 25 into the Draw Size box. Note that the default port number is 2. Select the OK button.

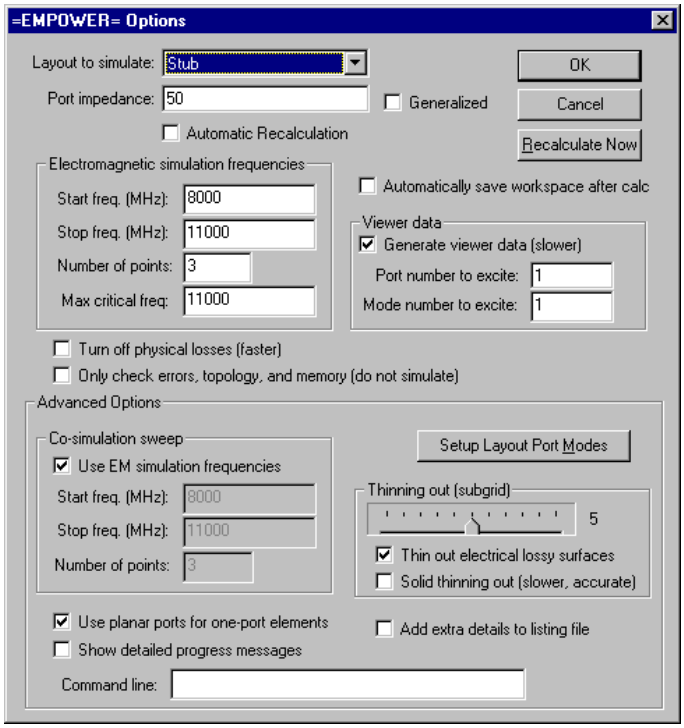
The screen should now look like:



For simulation, EMPOWER will take S-Parameters from these ports.

Simulating the Layout

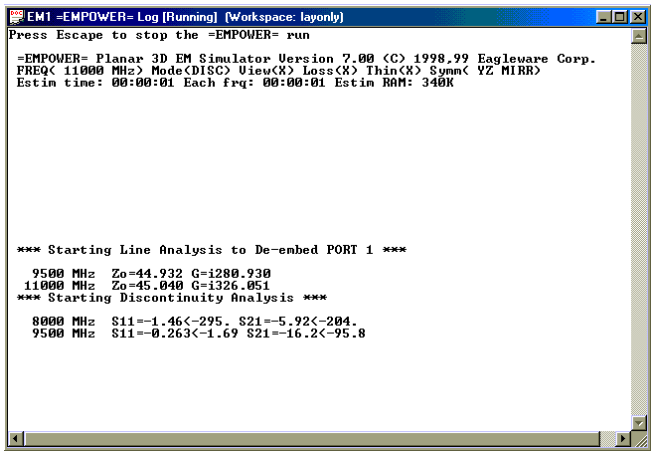
To run EMPOWER, you must create a simulation. Right-click on “Simulations/Data” in the WorkspaceWindow and choose Add Planar 3D EM Analysis from the menu. Accept “EM1" as the analysis name. This displays the EMPOWER Options dialog. This dialog is shown below. For a description of the dialog options, see the section on External Ports. For now, ***just set the prompts as shown below.***



We are starting with 3 sample points in the range 8-11 GHz. This will place 1 point at 8, 9.5 (the supposed resonance), and 11 GHz.

Click the Recalculate Now button. This launches EMPOWER to simulate the layout.

> **Note:** While EMPOWER is calculating, a window similar to the one in below will be shown. This window shows the current status throughout the calculation mode. For more details on this window, see the Basics Console section.



Viewing Results

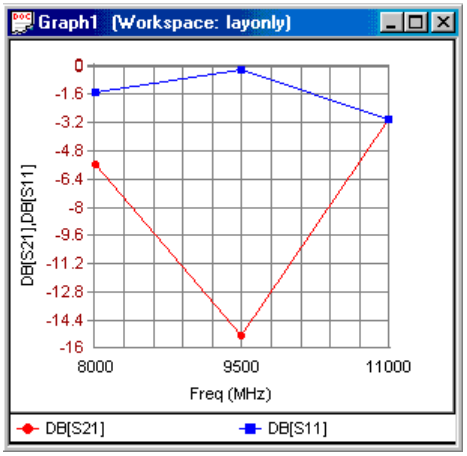
After EMPOWER simulation of the layout, the data must be displayed in GENESYS. This is done by creating a Data Output such as a Rectangular Graph.

To create a rectangular graph in this workspace:

1. Right-click on Outputs in the Workspace Window and Select Add Rectangular Graph from the menu. Accept the default name “Graph1".
2. Select EM1.Stub for “Default Simulation/Data or Equations”.
3. Enter S21 for the first measurement and S11 for the second measurement.

This instructs GENESYS to display a window containing EMPOWER data S21 and S11 will be displayed at 8000, 9500, and 11000 MHz. (For a complete description of rectangular graphs, see the GENESYS User’s Guide.)

The GENESYS display below shows the EMPOWER run with 3 sample points.



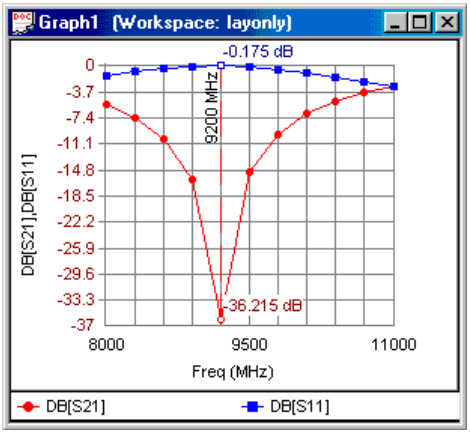
In this response, the notch frequency appears to occur exactly at 9.5 GHz. Or does it? Let’s add some frequency points to the EMPOWER simulation.

To re-simulate, adding more points:

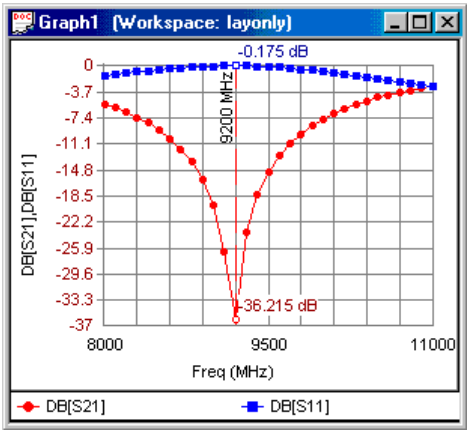
1. Double-click “EM1" under Simulations/Data in the Workspace Window.
2. Change the “Number of Points:” prompt in the “Electromagnetic Simulation Frequencies” to 11.
3. Click the Recalculate Now button.
4. Close the EMPOWER log (click on the X in the upper-right corner of the window).

This will add to the previous EMPOWER simulation so that we have 11 instead of 3 data points. EMPOWER will intelligently recalculate only the additional points.

The figure below shows the simulation with 11 EMPOWER data points. The notch frequency now appears to be at 9.2 GHz! Let's add the full 31 points to the EMPOWER simulation to ensure that we get the actual notch frequency. Repeat the previous steps to change the number of EMPOWER points to 31 and recalculate.



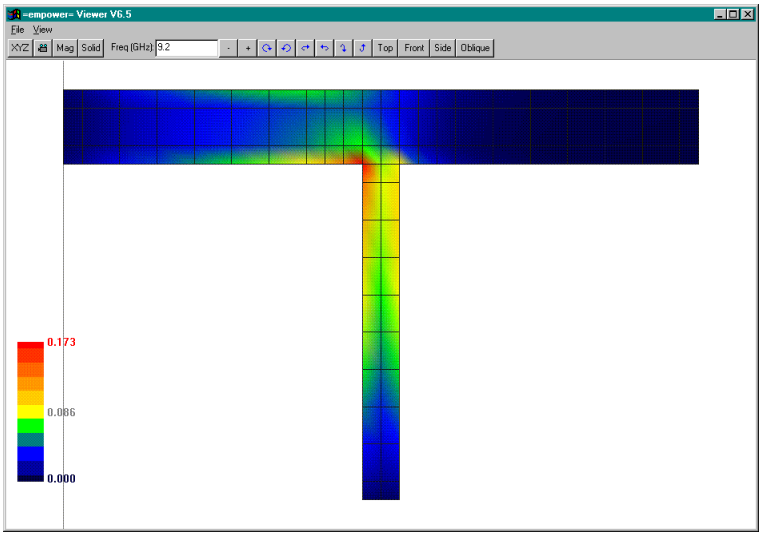
The display below is after the EMPOWER run with 31 points. The response has not changed noticeably since the 11 point simulation, so we must have found the correct notch frequency.



For the example filter, the notch occurs at 9.2 GHz instead of the desired 9.5 GHz. Much of this shift is due to rounding the line dimensions to the nearest 5 mils.

Using the Viewer

Once the EMPOWER run is completed, the viewer can be loaded if Generate Viewer Data was selected in the EMPOWER options dialog. Generating this data slows the EMPOWER simulation, so it's usually only checked during last-run simulations.



Right-Click the EMPOWER simulation in the Workspace Window and select Run Viewer. A top-down view has been selected, and the notch frequency has been specified. Port #1 is at the left of the image, and port #2 is at the right. The plot is color-coded to the scale given in the lower left of the figure. Notice that port #2 is nearly black. This indicates that very little energy is being delivered to that port at 9.2 GHz, as we'd expect.

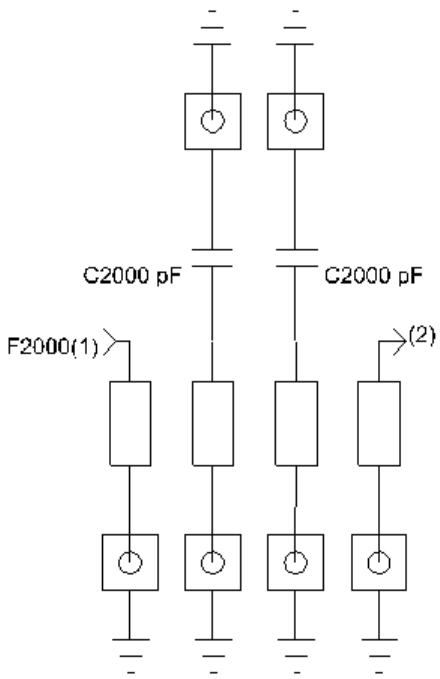
Creating a Layout From an Existing Schematic

The file used in this example is Filters\Tuned Bandpass.wsp. This example demonstrates the following topics:

- Creating a layout from an existing schematic
- "Tuning" with EMPOWER data
- Using lumped elements with EMPOWER

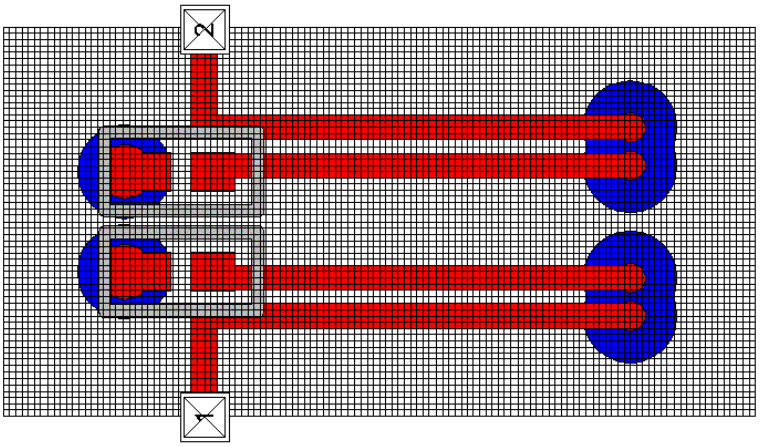
This circuit is a tunable bandpass filter. Operational theory is given in the example above.

In GENESYS, select Open Example from the File menu. Then, select "Tuned Bandpass.wsp" from the Filters directory. Double-Click F2000 under Designs in the Workspace Window to display the schematic for this filter (shown below).



This is the schematic of a 2nd order microstrip combline bandpass filter with 50 W terminations and transformer coupling on the input and output. The lumped capacitors are gang-tuned to adjust the resonant frequency of the two center lines. Tuning in this manner affects only the center frequency, and keeps the passband bandwidth constant.

Double-Click Layout1 under Designs in the Workspace Window to display the layout for this schematic. The layout for this example is shown below.

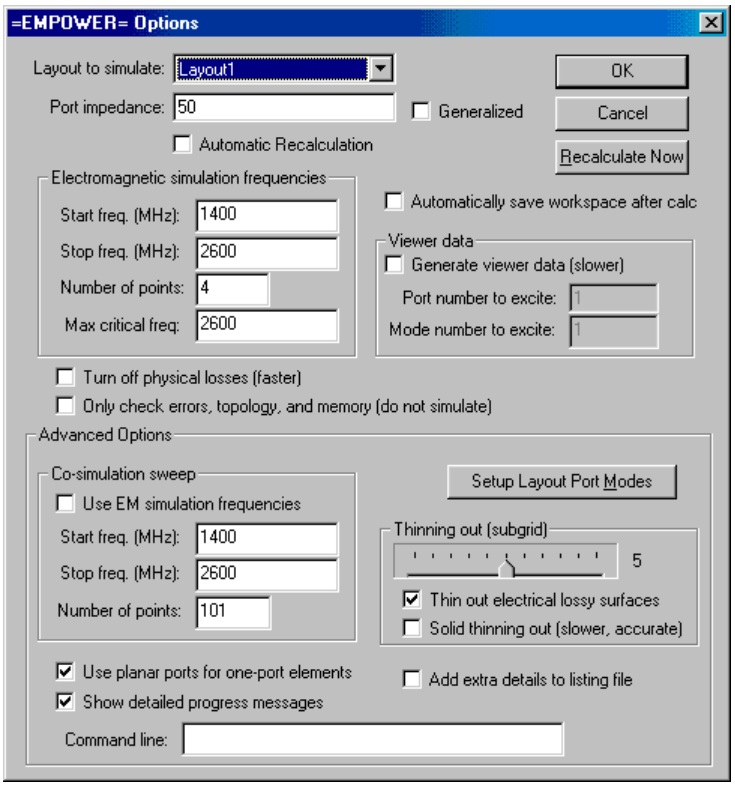


A 0402 Chip Capacitor footprint has been used for each of the lumped capacitors. Whenever a lumped element is used for an EMPOWER run, GENESYS creates an internal ports for the element. These ports are placed:

- If “Use Planar Ports for two-port elements” is checked in the EMPOWER properties box, one port is created for 2-terminal elements (like resistors or capacitors) which are aligned horizontally or vertically.
- In all other cases, an internal port is used for each terminal of the element. This port is placed at the center of the pad footprint, and EMPOWER writes data for each port created, whether internal or external.
- The “1" and ”2" ports pictured in the figure above are examples of external ports. Ports are described in the External Ports and the Internal Ports sections.
- This is a powerful technique, since real time tuning can be employed in GENESYS once the EMPOWER data for has been calculated.

Simulating the Layout

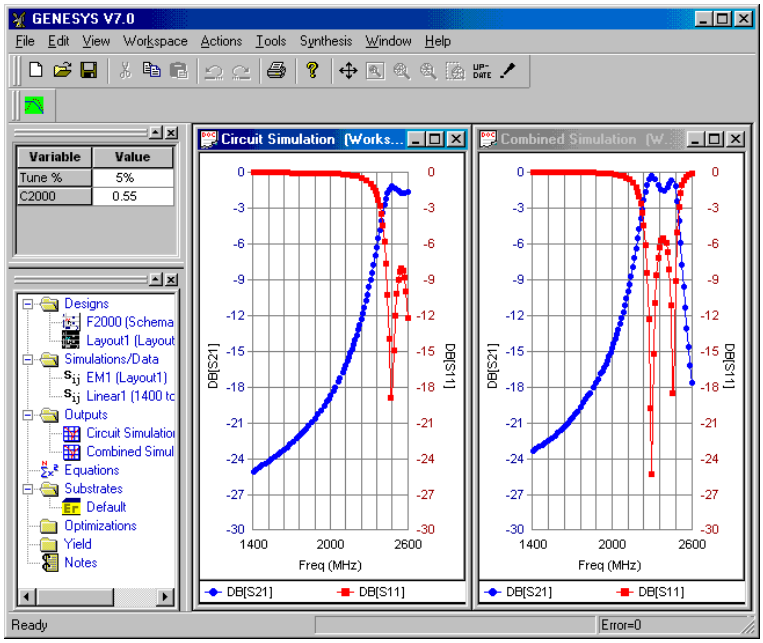
Double-click EM1 in the Workspace Window. This displays the EMPOWER Options dialog shown below.



Click the Recalculate Now button. If anything has been modified since the last EMPOWER run, this launches EMPOWER to simulate the layout.

Note: EMPOWER has been given a lot of intelligence to determine when it needs to calculate. Clicking Recalculate Now will not do anything if EMPOWER believes it is up to date. To force EMPOWER to recalculate from scratch, right-click on the electromagnetic simulation in the workspace window and select "Delete all internal files".

Once EMPOWER calculation is completed, GENESYS displays the calculated data. The graphs below show GENESYS after EMPOWER simulation. (Double-click the graph items in the workspace window to open them and select Tile Vertical from the Window menu to organize them.)



Lumped Elements

The first example in this section required several data points to find the exact notch frequency. This second example only used 4 data points, and produces data very close to the SUPERSTAR simulation. This is because the capacitors which load the coupled lines (causing resonances at the center frequency) were removed during the EMPOWER simulation. This effectively removes the resonances from the simulation range, producing a flat response from the open coupled lines. Since a flat response is well suited for linear interpolation, few data points are required in the EMPOWER simulation. In the EMPOWER options dialog, the Co-Simulation Sweep box is used to set up a simulation with more points after lumped elements are added.

When GENESYS uses the EMPOWER results, it replaces the lumped capacitances, resulting in the bandpass response shown in the previous section.

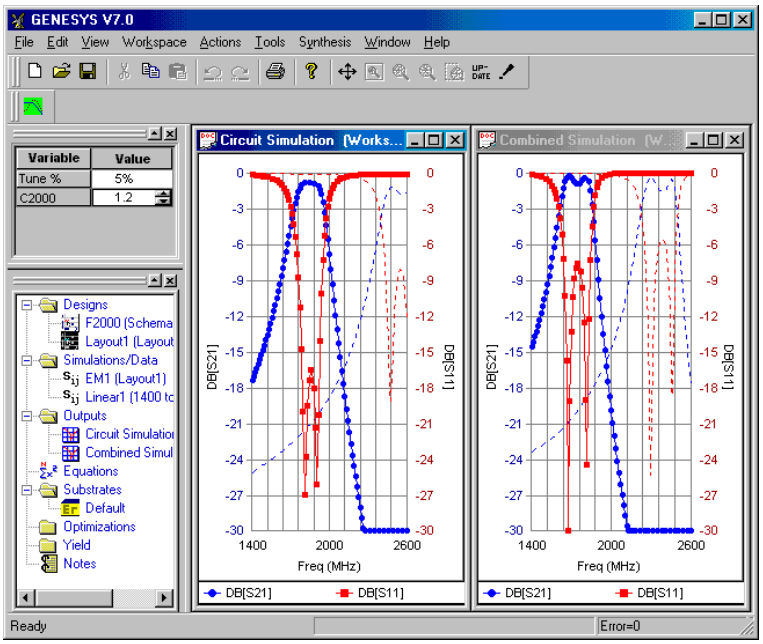
Real-Time Tuning

As stated before, GENESYS creates ports internal to a layout structure containing lumped elements before invoking EMPOWER. During calculation, EMPOWER creates s-parameter data with port data for all ports, whether internal or external. This allows GENESYS to tune the lumped elements while still using the EMPOWER data.

To see an example of tuning:

1. Click inside the C2000 prompt in the Tune Window.
2. Type a new value for the capacitor, or tune using Page Up/Page Down keys or the spin buttons.

The GENESYS screen below is shown after tuning the capacitors from 0.55 pF to 1.2 pF. The response shown on the left in this figure is the SUPERSTAR linear simulation response. The EMPOWER data is combined with the lumped elements in the rightmost response.



Chapter 19 EMPOWER: Basics

Overview

A major part of any electromagnetic simulation is to break the problem down into manageable size pieces that allow an approximation of Maxwell's equations to be solved. Electromagnetic simulators traditionally fall into three major categories: 2-D, 3-D, and 2-1/2-D.

2-D SIMULATORS

2-D simulators can only analyze problems that are infinitely continuous in one direction. Ideal transmission lines and some waveguide problems are practical problems which fall into this category. A 2-D simulator will analyze a slice of the line(s) and determine propagation, impedance, and coupling values. 2-D simulators are the fastest but most limited type of simulator available.

3-D SIMULATORS

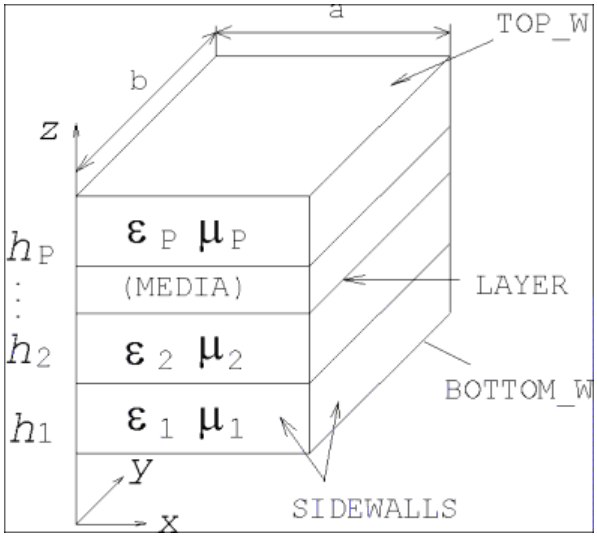
3-D simulators can analyze virtually any type of problem and are ideal for use with non-planar geometries such as a coaxial T-junction, radar target reflections, or other truly three dimensional problems. 3-D simulators have the advantage that they can analyze almost any problem, but they have the disadvantage that they are extremely slow.

2 1/2-D SIMULATORS

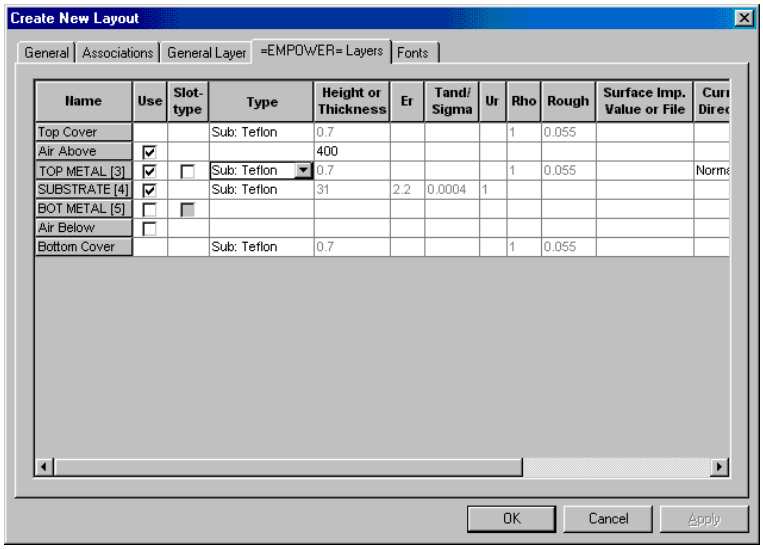
2 1/2-D Simulators are simulators designed for mainly planar (microstrip, stripline, etc.) circuits. While they have less flexibility than true 3-D simulators, they are much faster and are ideally suited for microstrip, stripline, and other similar geometries. EMPOWER is an advanced 2 1/2-D simulator. It can solve planar problems as well as problems with via holes and other z-directed currents, putting it in a class above true 2 1/2-D simulators which do not allow z-directed currents. In fact, most people would consider EMPOWER to be a 3-D simulator because it can handle z-directed currents.

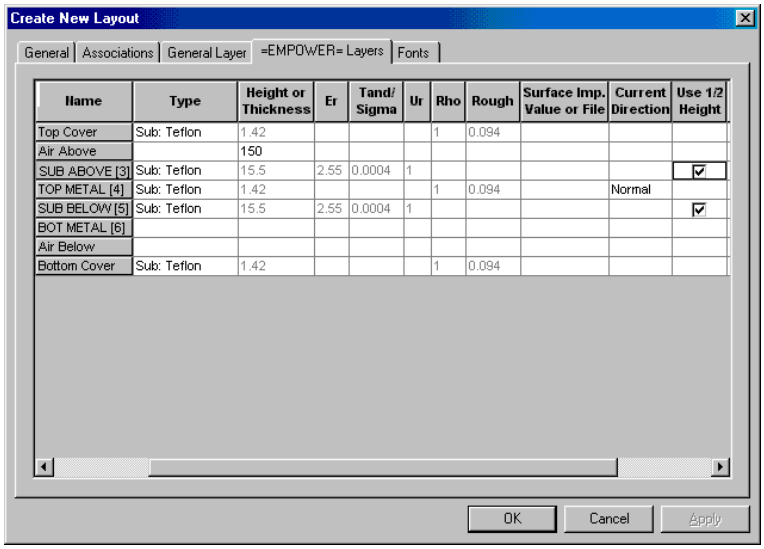
Basic Geometry

All circuits in EMPOWER exist in a rectangular box as shown below. The Media (substrate) layers each have specific dielectric and permittivity constants and loss tangents. There must be at least two media layers: One above the metalization layer and one below. For standard microstrip, there is a substrate below and air above. For suspended microstrip, there are three media layers (two air and one substrate). For buried microstrip, there are also three media layers (two substrate and one air).



The dialogs below show two typical EMPOWER Layer Tab setups: one for microstrip and one for stripline (triplate). The EMPOWER Layer Tab must be carefully checked when a new problem is created, as it is probably the most likely source of errors when setting up an EMPOWER run.





The EMPOWER Layer Tab consists of the following main entries:

Top Cover and **Bottom Cover** - Describes the top and bottom covers (ground planes) of the circuit:

- Lossless: The cover is ideal metal.
- Physical Desc: The cover is lossy. These losses are described by Rho (resistivity relative to copper), Thickness, and Surface Roughness.
- Electrical Desc: The cover is lossy and is described by an impedance or file. See the description below under metal for more information.
- Semi-Infinite Waveguide: There is no cover, and the circuit is simulated as if the box walls and uppermost substrate/air layer extend up or down forever (an infinite tube).
- Magnetic Wall: The cover is an ideal magnetic wall. This setting is only used in advanced applications.
- SCHEMAX substrates: Choosing a SCHEMAX substrate causes the cover to get the rho, thickness, and roughness parameters from that substrate definition. We recommend using this setting whenever possible so that parameters do not need to be duplicated in SCHEMAX and LAYOUT.

Air Above and **Air Below** - The presence of air at the top of the box (as in microstrip) or the bottom of the box (as in suspended microstrip) is so common that special entries have been provided for these cases. Checking the box to turn these layers on is the equivalent of adding a substrate layer with Er=1, Ur=1, and Height (in units specified in the Dimensions tab) as specified.

Caution: When setting up a new circuit, be sure to check the height of the air above, as it is often the only parameter on this tab which must be changed, and is therefore easily forgotten.

Metal Layers - In LAYOUT, multiple METAL layers (e.g., copper and resistive film) are automatically converted to one EMPOWER signal layer if no media layer is in between the metal layers.

All metal layers from the General Layer Tab are also shown in the EMPOWER Layer tab. These layers are used for metal and other conductive material such as resistive film. The following types are available:

- **Lossless**: The layer is ideal metal.
- **Physical Desc**: The layer is lossy. These losses are described by Rho (resistivity relative to copper), Thickness, and Surface Roughness.
- **Electrical Desc**: The layer is lossy and is described by an impedance or file. This type is commonly used for resistive films and superconductors. If the entry in this box is a number, it specifies the impedance of the material in ohms per square. If the entry in this box is a filename, it specifies the name of a one-port data file which contains impedance data versus frequency. This data file will be interpolated/extrapolated as necessary. See the Reference manual for a description of one-port data files.
- **SCHEMAX substrates**: Choosing a SCHEMAX substrate causes the layer to get the rho, thickness, and roughness parameters from that substrate definition. We recommend using this setting whenever possible so that parameters do not need to be duplicated in SCHEMAX and LAYOUT.

Caution: Thickness is only used for calculation of losses. It is not otherwise used, and all strips are calculated as if they are infinitely thin.

Metal layers have three additional settings available:

Slot Type - Check this box to simulate the non-lossless-metal areas (as opposed to the metal areas) in EMPOWER. Use this for ground-planes and other layers which are primarily metal. Do not use this for lossy layers. See your EMPOWER manual for details.

Current Direction - Specifies which direction the current flows in this layer. The default is along X and Y. "X Only" and "Y Only" can be used to save times on long stretches of uniform lines. "Z Up", "Z Down", "XYZ Up", and "XYZ Down" allow the creation of thick metal going up/down to the next level or cover.

Thick Metal - Checking this box forces EMPOWER to model the metal including thickness. EMPOWER does this by putting two metal layers close together, duplicating the traces on each, and connecting them with z-directed currents. If thick metal is used, then Current Direction is ignored.

Element Z-Ports - This setting specifies the default direction for automatically created element ports, either to the level above or to the level below. Generally, you should choose the electrically shortest path for this direction.

Substrate/Media Layers - All substrate layers from the General Layer Tab are also shown in the EMPOWER Layer tab. These layers are used for substrate and other continuous materials such as absorbers inside the top cover. An unlimited number of substrate/media layers can be used. The following types are available:

- Physical Desc: The layer is lossy. These losses are described by Height (in units specified in the Dimensions tab), Er (relative dielectric constant), Ur (relative permittivity constant, normally 1), and Tand (Loss Tangent).
- SCHEMAX substrates: Choosing a SCHEMAX substrate causes the cover to get the height, Er, Ur, and Tand parameters from that substrate definition. We recommend using this setting whenever possible so that parameters do not need to be duplicated in SCHEMAX and LAYOUT.

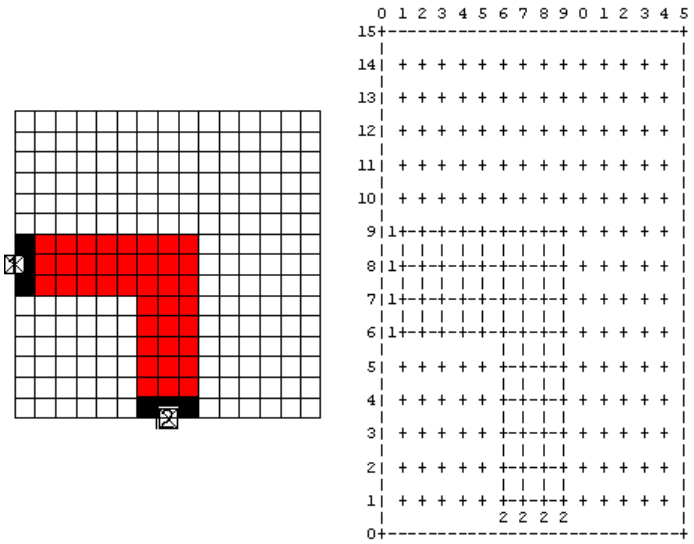
Caution: For true stripline (triplate), be sure to check the "Use 1/2 Height" checkbox if you are using a substrate from SCHEMAX. This forces EMPOWER to use 1/2 of the SCHEMAX substrate height for each substrate (above and below) so that the total height for both media layers is correct.

In addition to the metalization and substrate layers, viaholes and other z-directed currents can be used. These currents can go from the metalization layer through one media/air layer to either the top or bottom walls.

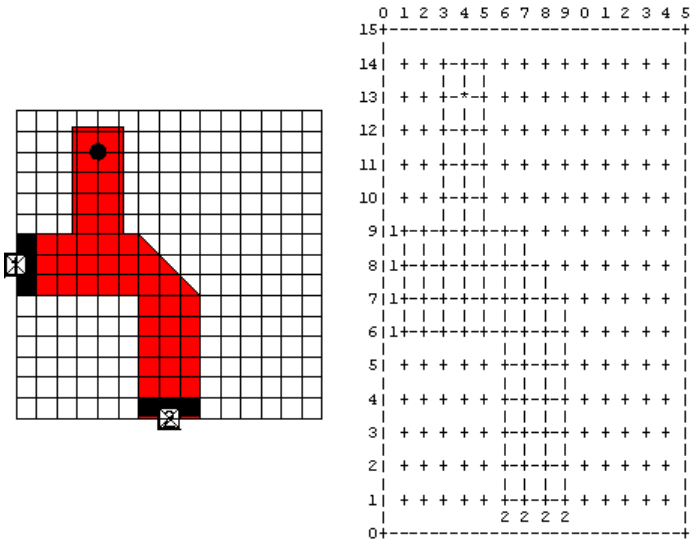
Besides conductive materials, ports are placed on the metal layers and in z-directed positions.

The Grid

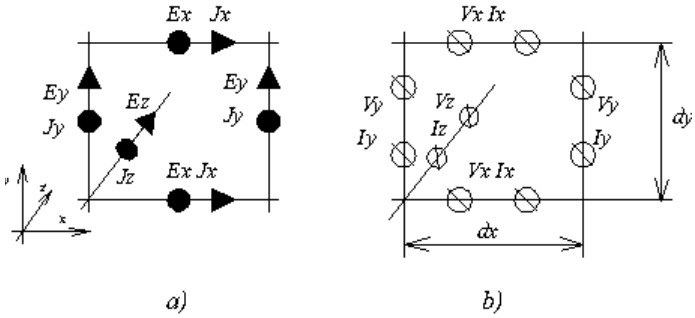
All conductive surfaces and ports must be on a grid. This grid is composed of regular rectangular cells. An example of mapping a microstrip bend to the grid is shown below. The left half of the figure shows the circuit as it appears in LAYOUT. The right half of the circuit shows a part of the EMPOWER listing file. Each of the plus signs ("+") in the listing file represents an intersection of two grid lines as shown on the layout. Lines connecting plus signs represent metal. Numbers represent port locations. Notice that the ports map onto the grid in place of metal, so the ports go between the end of the line and ground (the wall), so each port has a ground reference as would be expected.



EMPOWER will move all surfaces to the nearest grid cell before analyzing a circuit. EMPOWER maps the structure onto the borders of the cell, not onto the space inside the cell. A slightly more complex example which does not exactly fit the grid is shown below. There are three important things to notice in this figure: 1) The stub line going up is about 2 1/2 cells wide, but is approximated by EMPOWER as being 2 cells wide. 2) The chamfered corner is approximated by a "stairstep." 3) The viahole near the end of the stub is represented by an asterisk in the listing.

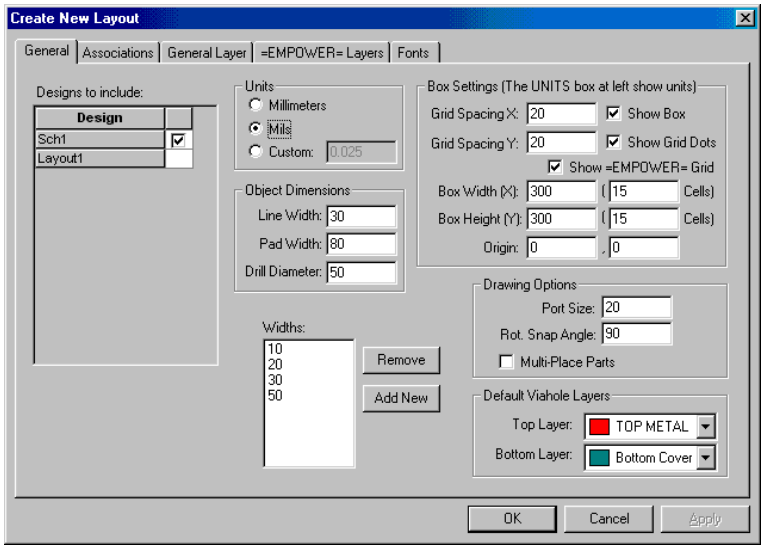


A close-up is shown below where you can see how metal and ports are mapped onto the borders of the cells. The presence of metal or conductors along the grid causes EMPOWER to close the connections along the grid. The presence of an EMPort causes the line to be opened, creating an open circuit which turns into a port in the final data file.



Note: It is possible to make a line so narrow that it maps to one border between cells (zero cells wide). This is legal, but is not normally recommended and should be used only for very high impedance lines where accuracy is not important, such as DC power lines.

The grid and the box are controlled with parameters in the Preferences box from the LAYOUT File menu. The Dimensions Tab shown below is as it was setup for the microstrip bend above.



The following entries are especially relevant to EMPOWER:

Show EMPOWER Grid - Turning on this checkbox forces LAYOUT to display the rectangular EMPOWER grid. It also allows different grid spacings in the X and Y dimensions. *It is strongly recommended to turn this checkbox on whenever you are creating a layout for EMPOWER.*

Grid Spacing X and Grid Spacing Y - These control the cell size for the EMPOWER run as well as the grid snap feature in LAYOUT. When using the "EMPOWER Grid Style," there will be LAYOUT snap points between each grid line which allow lines to be

centered between two grid points if necessary. They are often referred to as *dx* and *dy* and should be small with respect to a wavelength at the maximum frequency to be analyzed, preferably less than wavelength/20 and always less than wavelength/10.

Box Width and Box Height - These are the box size for EMPOWER simulation. They correspond directly to the SIZE statement in the TPL file. The number of cells across the box (equal to Width or Height divided by Grid Spacing X or Y) is displayed for your convenience and can be changed to adjust the page width. Note: Any metal put down completely outside the box will be ignored by EMPOWER. This can be used to your advantage to temporarily or permanently remove metal or components from the EMPOWER simulation.

Default Viahole Layers - The Start Layer and End Layer combo boxes control the default layers for the viaholes. These can be overridden individually for each viahole.

Viaholes and Z-Directed Ports

The grid in EMPOWER is a truly three dimensional grid (rectangular lattice). Z-Directed currents and ports are mapped from the intersection points to the top or bottom cover. There are two caveats: metal and ports in the z-direction are modeled as one continuous current, so the viaholes should be small in comparison with a wavelength. Also, you cannot have both a port and metal along the same grid line, so you should be extremely careful when placing a viahole directly underneath an internal port. You should check the listing file (select "Show Listing File" from the EMPOWER simulation right-click menu) carefully to see that both the port and the viahole are represented on the grid.

The physical length of a viahole in a substrate should be kept shorter than about 1/10 to 1/20 wavelength within the analysis range. Longer lengths can suffer calculation inaccuracies in EMPOWER. For example, suppose a microstrip circuit with a 10 mil substrate and a dielectric constant of 2.4 is to be used. What is the highest accurate frequency for this setup?

Note: If the substrate layer is broken down into two substrate layers (by adding an additional layer), each 1/2 the height of the original, then the viaholes will be accurate at twice the original frequency. This procedure can be repeated as necessary.

$$\varepsilon_0 = 8.8542 \times 10^{-12} \ \ F/m$$

$$\mu_0 = 1.2566 \times 10^{-6} \ \ H/m$$

$$c = \frac{1}{\sqrt{\varepsilon_r \varepsilon_0 \mu_0}} = \frac{1}{\sqrt{2.4 \varepsilon_0 \mu_0}} = 1.935 \times 10^8 \ \ m/s$$

$$\frac{\lambda}{15} = 10 \ mils = 2.54 \times 10^{-4} \ m \Rightarrow \lambda = 3.81 \times 10^{-3} \ m$$

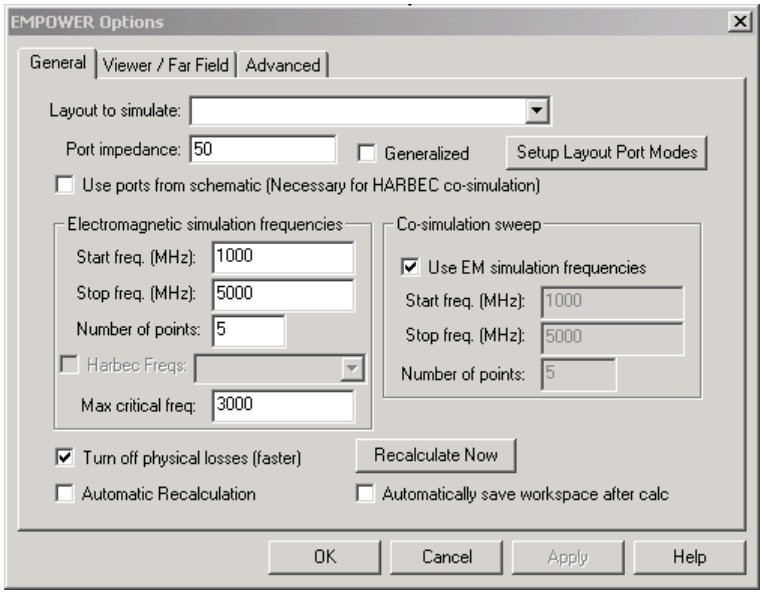
$$f_{\max} = \frac{c}{\lambda} = \frac{1.935 \times 10^8 \ m/s}{3.81 \times 10^{-3} \ m} \approx 50 \ GHz$$

EM Ports

All circuits must contain at least one EMPort to allow data to be taken from the EMPOWER simulation. The number of ports is equal to the number of ports in the EMPOWER network to be analyzed. They are placed in the layout using the EMPort button and can be Normal deembedded external ports (gray), external ports with No Deembedding (white), or internal ports (white). External Ports and Lumped Elements and Internal Ports are discussed in their respective sections.

EMPOWER Options

To open: double-click or create a Planar 3D Electromagnetic Simuation.



General Tab

Layout to Simulate - Allows you to select which layout in the current workspace to simulate. Since workspaces can have multiple layouts and multiple EMPOWER simulations, you can simulate many different layouts within the same workspace.

Port Impedance - When EMPOWER S-Parameter data is plotted on a graph, it will be normalized to this impedance. Different impedances can be used for each port by separating impedances with commas. A 1-Port Device Data File can be used in place of any impedance file to specify frequency dependent or complex port impedances.

Generalized - When this box is checked, the impedance for each line as calculated by EMPOWER are used for their terminating impedance. See your EMPOWER manual for details on Generalized S-Parameters.

Use ports from schematic - Check this box when co-simulating with HARBEC harmonic balance nonlinear simulation. This forces all sources and impedances to be considered in the simulation.

Note: Be sure to check "Use ports from schematic" if you will be using this simulation as the basis for a HARBEC Simulation, otherwise there will be no nonlinear sources available.

Electromagnetic Simulation Frequencies - Specifies the frequencies at which to run EMPOWER. If you have lumped elements in your simulation, you can often turn down the number of frequencies here and increase the number of frequencies in the Co-simulation sweep specified below.

Start Freq (MHz) - Specifies the minimum frequency to analyze.

Stop Freq (MHz) - Specifies the maximum frequency to analyze.

Number of Points - Specifies the number of frequency points to analyze. Points are distributed linearly between the low and high freq specified above.

HARBEC Freqs - Select this box to cause EMPOWER to simulate the layout at each frequency calculated by the harmonic balance simulator. Checking this box makes sure that EM results are available at all frequencies so that the data will not need to be interpolated or extrapolated for harmonic balance analysis.

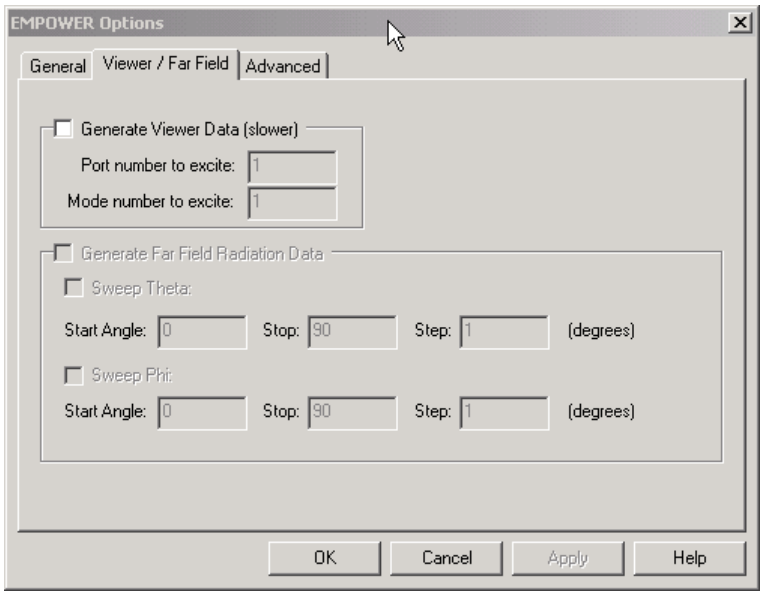
Max Critical Freq (MHz) - Specifies the highest important frequency that will be analyzed on any run of this circuit. MAXFRQ is specified in the units defined in the DIM block. (The default units are MHz.) Parameters of the solution quality, thinning out thresholds, and lengths of lines for de-embedding are based on the maximum critical frequency value. In other words, this value influences both accuracy of simulation and calculation time. Decreasing the value accelerates simulation but may increase model error, especially at frequencies above the value, On the other hand, an unnecessarily high value may slow down the solution without visible improvements in accuracy.

Note: An important reason to specify MAXFRQ: By default this value is set equal to the highest sweep frequency specified in EMFRQ. Even a small change of its value may cause the grid to change, forcing recalculation of de-embedding parameters and unnecessarily increasing simulation time as a consequence, This change will also change the answer slightly, with disastrous results if you are merging data. This will not happen if you use MAXFRQ. It is also important to remember to update it if you change the frequency range substantially.

Co-Simulation Sweep - Specifies the frequencies at which to run simulate the lumped elements + EMPOWER data combination. If you have no lumped elements in your simulation, you should normally check the "Use EM Simulation Frequencies" box. For circuits with lumped elements, you can often save much time by using fewer points in the electromagnetic simulation frequencies above, allowing the co-simulation to interpolate the EMPOWER data *before* the lumped elements are added.

Turn off physical losses (Faster) - If checked, EMPOWER will ignore any losses specified in the EMPOWER Layer tab. This option is very useful to speed up any preliminary runs.

Automatically save workspace after calc - This checkbox is handy for overnight runs to help protect against a power outage. Note that checking this box will force the *entire* workspace to be saved after each run.



Viewer / Far Field Tab

Generate Viewer Data (Slower) - Checking this box causes EMPOWER to generate a *.EMV file that can be loaded in the EMPOWER current/voltage viewer program. Selecting this box will increase the amount of time required to solve the problem. This box must also be checked in order to generate far field radiation data. See the Viewer section for more information.

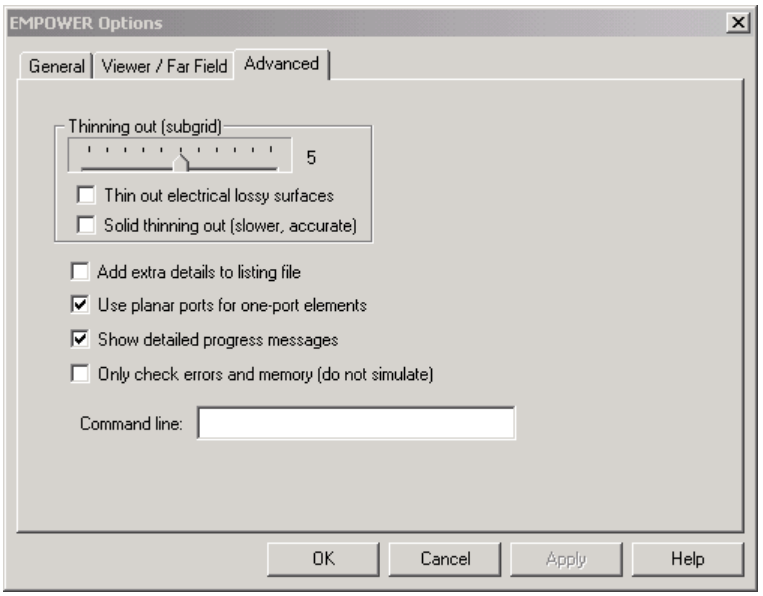
Port number to excite - This option is available if "Generate viewer data" above is checked. It specifies which EMport to excite for viewer data. By default, mode one is excited, but if the input is multi-mode, then you can add the option -Imj to excite mode j instead.

Mode number to excite - This option is available if "Generate viewer data" above is checked. It specifies which mode to excite for viewer data. Generally, mode one is excited, but if the input is multi-mode, then you can add excite any mode number up to the number of modes at that input.

Generate Far Field Radiation Data - Checking this box causes EMPOWER to generate data for the radiated electric fields of a structure in the far field region. The data generated is specified by the sweeping theta and phi coordinates of the spherical coordinate system.

Sweep Theta - This option is available if "Generate Far Field Data" above is checked. It generates data for varying theta in the spherical coordinate system. Theta is the angle formed from the z-axis to a point in 3-space. If "Sweep Theta" is unchecked, a fixed-angle will be specified, and far-field data will be produced only at this theta angle.

Sweep Phi - This option is available if "Generate Far Field Data" above is checked. It generates data for varying phi in the spherical coordinate system. Phi is the angle formed from the positive x-axis to a point projected on the xy plane in 3-space. If "Sweep Phi" is unchecked, a fixed-angle will be specified, and far-field data will be produced only at this phi angle.



Advanced Tab

Only check errors, topology, and memory (do not simulate) - Useful to make sure you have the simulation and layout setup properly before a long EMPOWER run. This option provides a very important means both for checking the grid mapping and required memory. EMPOWER just maps the problem onto the grid and calculates the required number of the grid variables for each frequency. Check the “map of terminals” in the listing file to see the grid model of the problem and check the MEMORY lines in the listing file to get some idea about problem complexity and probable simulation time.

Setup Layout Port Modes - Brings up the multi-mode setup dialog box as described in the Decomposition section of your EMPOWER manual. If this button has exclamation points on it, then multi-mode lines are active.

Thinning out (slider) - Control the amount of thinning. The default thinning out amount is 5. Setting the slider to zero turns off thinning. See your EMPOWER manual for details on thinning.

Thin out electrical lossy surfaces - If checked, lossy metal described using electrical parameters will also be thinned. Since the thinning out model assumes that most current flows on the edges of the lines, this option will be somewhat less accurate for resistive films (where current flows more evenly throughout the material). In these cases, you should probably also check the Solid thinning option shown below.

Solid Thinning out (slower) - If checked, slower solid thinning out model is used. This model restores capacitance lost due to thinning out and can be most useful for when large sections of metal have been thinned out.

Use planar ports for one-port elements - This box should almost always be checked. When not checked, EMPOWER uses z-directed ports at each terminal for all devices. When it is checked, EMPOWER uses in-line ports for elements like resistors and capacitors (two-terminal, one-port devices). The only time this can cause a problem is when you have a line running "under" an element (for example, running a line between the two terminals on a resistor, in the same metal layer as the resistor pads).

Note: EMPOWER planar ports cannot be used for ground referenced elements, such as transmission lines, even though the element might only have terminals.

Add extra details to listing file - If checked, extra information which can be used to double-check your setup is inserted into the listing file.

Show detailed progress messages - Turning this option off suppresses almost all output in the EMPOWER log. (The listing file is not affected.) Turning it off can dramatically speed up very small runs.

Command Line - Some options are available which are not shown on this dialog box. One common example is the -O*n* option which controls the size of the box for line analysis.

-NC - If this option is used, EMPOWER will allow de-embedded ports to be away from the wall. This option is especially useful for finline and slotline configurations.

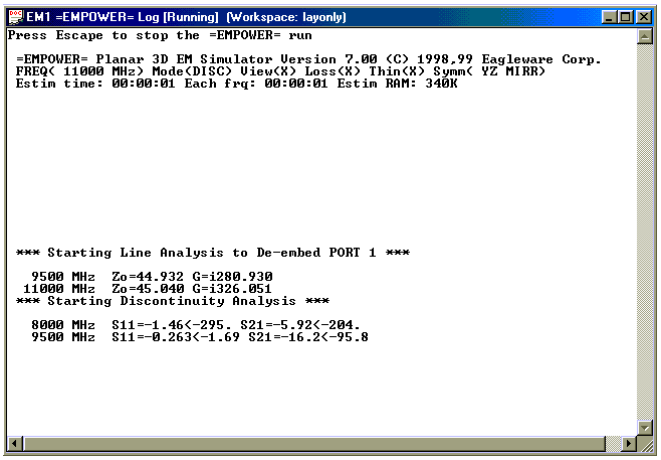
-VM - Allow virtual memory usage. To solve a complex problem, EMPOWER always limits usage of computer virtual memory (hard disk space) in a rational way. It will not use it for some numerically intensive parts of the simulation. The option VM tells EMPOWER to use virtual memory more freely. But even with this option, the program stops calculations if substantial hard disk space is involved in some parts of the simulation. Check the MEMORY lines in the listing file to have an idea how much memory your computer lacks or how to reduce the problem.

-Sg - Use an alternate method of thinning out, "Global" thinning. Can reduce memory requirements under some circumstances.

-O*n* - Use a smaller line segment (*n* times smaller) for de-embedding calculations. Can speed up line analysis.

-IT - Output viewer data file in text format (.PLX).

Console Window



The window above is shown when EMPOWER is running.

The objects on the second line are:

FREQ - The current calculation frequency.

Mode - DISC (discontinuity), LINE (line analysis), or LN+D (both).

View - Checked if viewer data is to be generated.

Loss - Checked if physical loss is being modeled.

Thin - Checked if thinning is enabled.

Symm - Displays the type of symmetry possessed by the circuit being analyzed. This option can be XZ, YZ, Mirror, 2-way mirror, or 180o rotational.

The objects on the third line are:

Estim Time - The estimated total time to complete the current calculation mode.

Each frq - The estimated calculation time per frequency in the current mode.

Estim RAM - The estimated total memory required for the current simulation.

The fourth line displays the simulation time of the current frequency and symmetry, plus the symmetry stage.

The fifth line displays the calculation stage.

The lines below the fifth line describe the calculated data for each frequency. During line analysis, the impedance (Z) and propagation constant (G) are displayed for each frequency. In the discontinuity calculation mode, the first row of the s-matrix is displayed at each frequency.

Batch Runs

Starting with GENESYS Version 7.0, multiple workspaces can be loaded simultaneous, and all EMPOWER simulations can be updated sequentially. This new capability makes the techniques given in this section much less important for most users. Simply open as many Workspace files as you need (Select Options from the Tools menu and check "Allow Multiple Open Workspaces"). Right-click on any of the EMPOWER simulations and press Recalculate Now. You will then be asked if you want to recalculate all simulations; select "Yes".

Note: You should probably check "Automatically save workspace after calc" if you are running long or overnight batches so that if there is a power outage you will not lose your results.

Chapter 20 EMPOWER: Tips

Overview

Often, electromagnetic simulation involves tradeoffs and compromises to keep simulation times and memory requirements as small as possible while making accuracy as high as possible. This section looks at several choices and clarifies the tradeoffs. Table 3-1 lists various features and gives their impact on simulation times, accuracy, and memory requirements. Each of these choices are looked at in detail below. The values are approximate and may vary.

Choice	Memory	Accuracy	Time
Reducing Cell Size by 2	x16	x2	x64
Raising Max Critical Freq	x1.5	x1.5	x2
Fixing Symmetry	x1/4 to x1/16	0	x1/4 to x1/16
Turning Off Thinning Out	x16	x1.2	x64
Increasing Wall & Cover Spacing	x1.1	x1.5	x1.1
Choosing Correct Cover	0	x4	0
Including Loss	x2	Lossy	x4
Generating Viewer Data	x1.2	0	x2-x10
Corecting Slot-Type Structure	x1/64	0	x1/256
Using Preferred Box Cell Count	0	0	
Using Thick Metal	x6	?	x16

Cell Size

Cells should be small enough so that the result is accurate, at least 10 cells per wavelength at the maximum critical frequency (see below). Additionally, the cells should be small enough that there is at least one, and preferably more, cell across every line and gap. Decreasing the cell size makes all stages of the solution take longer, so decreasing cell size can be an expensive way to get more accuracy. Conversely, increasing cell size is a great

way to do an initial run of your problem to make sure that the result is close before you start a simulation that will take hours.

See the EMPOWER Basics section for more information on cells and the problem geometry.

Maximum Critical Frequency

This parameter is set in the EMPOWER dialog box when starting a simulation. Changing this parameter has three (and only three) effects:

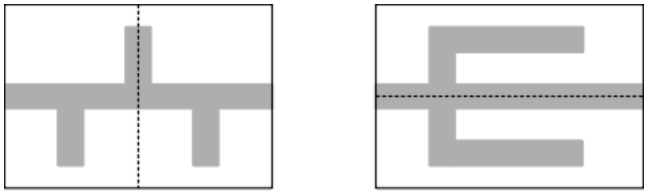
1. The maximum amount of thinning out is affected. EMPOWER will thin out until an area is 1/20th of a wavelength at this frequency in the default thinning mode.
2. The length of line analyzed for deembedding is 1/2 wavelength at this frequency in automatic mode.
3. Many parameters in the listing file are based on this frequency.

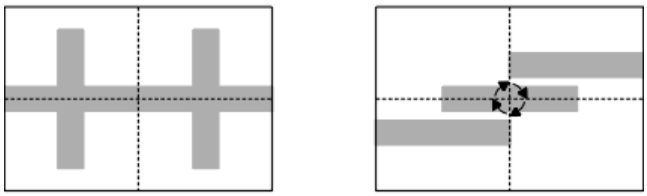
The most important thing to know about maximum critical frequency is to keep it the same between runs of the same problem, even if you are changing the frequency range which you are analyzing. If it is changed, then the thinning out is changed, and the entire problem geometry is slightly different. As an example, if you are analyzing a filter with a passband from 5.1 to 5.5 GHz with a reentrance mode (additional passband) around 15 GHz, you should probably set the maximum critical frequency to 5.5 GHz. This is because the exact characteristic of the reentrance mode probably is not important ("critical"); you just want to know approximately where the filter re-enters. On the other hand, you want to know *precisely* where the passband is, so you set the maximum critical frequency above it.

The effect of maximum critical frequency is generally secondary. Most of the other choices in the table above have a bigger effect on accuracy.

Symmetry

Making a problem exactly symmetrical is an easy way to make a problem require less memory and time without sacrificing any accuracy. There are four types of symmetry recognized by EMPOWER: YZ mirror symmetry, XZ mirror symmetry, two mirror symmetry, and 180° rotational symmetry. These types are illustrated below.





When EMPOWER is running, you should look at the information area at the top of the screen to see if symmetry is active. If it is not, recheck your problem to see if it is exactly centered on the box, and to see if it is in fact symmetrical. Two tools can help with this:

1. Using "Center Selected on Page" from the Edit menu in LAYOUT. This command makes it easy to make sure that your circuit is exactly centered on the page
2. Showing the listing file by selecting "Show Listing File" from the EMPOWER right-click menu. This file shows exactly how the problem was put on the grid, and lack of symmetry is often obvious.

Making an unsymmetrical problem symmetrical will make it run 4 times faster in most cases, and will make it 16 times faster if your problem can use two-mirror symmetry.

See EMPOWER Basics section for more information on cells and the problem geometry. See the Files section for more information on the listing file.

Thinning Out

For most examples, the default thinning out should be used. As a general rule, you will get better accuracy for a given amount of time and memory when you use thinning. Thinning out helps by removing currents which have little or no effect. This reduces the number of variables in the problem considerably with little effect on the accuracy of the solution. There are a few cases where thinning out should not be used, and they generally involve very large sections of metal which are affected too much by thinning out. The Dual-Mode Power Divider example is one of these cases.

Wall & Cover Spacing

Generally, the wall and cover spacing should match the problem which you are trying to model. This will give an accurate assessment not only of circuit performance but also of box resonances. However, this will not be possible in a few situations:

1. The structure will not be in a box.
2. You are analyzing part of a larger circuit, and the box walls would be prohibitively large to model.
3. You are designing a component (such as a spiral inductor) which will be reused in many different circuits, so the cover height is not known.

In these cases, you must use an approximation. Set the box size so that the walls are separated from the circuit by at least 3 times the substrate thickness, preferably 6 times. For microstrip, set the cover spacing (air above) to 5 to 10 times the substrate height.

See for more info, see Box Modes. See Microstrip Line for an example of the effect of wall spacing on line impedance.

Cover Type

Choosing the correct cover type is absolutely critical to getting an analysis which matches measured results. The choice is usually between whether to use an open cover or a closed cover. Choosing the correct cover type usually has no effect on analysis time, so there is no reason not to set this to the proper type. With an open cover, there will be radiation, and this can have a huge impact on circuit performance. You can choose the correct cover types in the Layers Tab when starting an EMPOWER run.

See the EMPOWER Basics and Box Modes sections for more information on covers. See the Edge-Coupled Filter example for an example of the impact that removing a cover has on circuit performance.

Lossy Analysis

If you do not need information about circuit loss, you can check the box labeled "Don't use physical loss (Faster)" when starting an EMPOWER run. Turning off losses will generally make a problem require 1/2 the memory and 1/4 the time as a lossy problem.

We recommend that you define all layers with their proper characteristics, including losses. You can then quickly change between lossy and lossless modes as described above. A common technique is to analyze a circuit first without losses, then turn on losses and run an analysis with a few points in it. This allows you to determine the amount of loss and confirm that it has no other major effect on performance while not having to wait the additional time while doing most of your analyses.

There is an additional caveat regarding loss described in the section on Slot-type structure.

See the Narrowband Interdgital example for an example of the effect of loss on an interdigital filter.

Viewer Data

Looking at currents in the viewer is a great way to get insights into circuit performance. However, generating this viewer data requires additional time, increasing the length of a run by a factor from two to ten, and sometimes requiring additional memory also. Generating viewer data has no effect whatsoever on the solution given, so you should not

have this option turned on unless you actually intend to run the viewer. You can turn this option on and off by using the checkbox labeled "Generate Viewer Data (Slower)" when starting an EMPOWER run.

You will not normally need viewer data, and when it is needed, you will not normally need viewer data at every frequency. Our recommendation:

1. Run all problems the first time without generating viewer data. If the answer is completely unexpected, check for errors in your description of the file. This can save a lot of time in the experimenting stage.
2. If you decide you want viewer data, open the EMPOWER Options dialog box. Reduce the number of frequency points to be analyzed and turn on "Generate Viewer Data (Slower)." Recalculate the EMPOWER simulation, and you will now have viewer data at some points.
3. If your problem is very large, you may want to increase the cell size or make other tradeoffs to reduce the time required for calculation. If you use this technique, save the file with a new name before you generate viewer data so that you do not corrupt your existing S-Parameter data!

See the EMPOWER Viewer section for more information.

Slot-Type Structure

In the normal mode EMPOWER solves for the currents in the metal. There is an additional mode where EMPOWER solves for the voltages in the gaps and in lossy metals. This mode must be turned on manually by checking "Slot-type structure" when starting an EMPOWER run from GENESYS, or by using the VOLTAGE keyword when describing a LAYER in a TPL file.

In general, you should check "Slot-type structure" whenever the metalization layer has more lossless metal than open space. This is often the case in a slot-type structure such as coplanar waveguide. The answer will always be identical, but you will save orders of magnitude of memory and simulation time by ensuring that this checkbox is set to the right value. Note: This setting has no effect on z-directed metal (viaholes, etc.) which is always calculated as currents.

There is a caveat when describing lossy problems with this option: All non-ideal metal must be analyzed, so if the metal in your problem is lossy, turning on "Slot-type structure" will result in both the air and the metal being analyzed, which will have a disastrous effect on memory and time requirements. Be sure that your metal layers are set to lossless if you check the slot-type structure box.

Preferred Cell Count

The first part of an EMPOWER run involves taking Fourier Transforms of the grid. These transforms will run much faster if the number of cells along the each side of the box is of the form $2^a3^b5^c7^d11^e13^f$ where e and f are either zero or one, and a,b,c, and d are arbitrary integers. In other words, a circuit with a box 512 cells by 512 cells (28 by 28) will analyze much faster than a circuit with a box 509 cells by 509 cells (509 is prime). Making

one side a preferred number will help, so a box 509 x 512 cells is better than one 509 x 509 cells. Note that only the time while EMPOWER is working on the Fourier Transform is affected, and this is normally only substantial with boxes 100x100 or larger. If you see a status with "FFT" in the message for a long time, check to see that the box width and height are a preferred number of cells across. Preferred numbers (which fit the form given above) 10000 and below are:

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 18 20 21 22 24 25 26 27 28 30 32 33 35 36 39 40 42 44 45 48 49 50 52 54 55 56 60 63 64 65 66 70 72 75 77 78 80 81 84 88 90 91 96 98 99 100 104 105 108 110 112 117 120 125 126 128 130 132 135 140 143 144 147 150 154 156 160 162 165 168 175 176 180 182 189 192 195 196 198 200 208 210 216 220 224 225 231 234 240 243 245 250 252 256 260 264 270 273 275 280 286 288 294 297 300 308 312 315 320 324 325 330 336 343 350 351 352 360 364 375 378 384 385 390 392 396 400 405 416 420 429 432 440 441 448 450 455 462 468 480 486 490 495 500 504 512 520 525 528 539 540 546 550 560 567 572 576 585 588 594 600 616 624 625 630 637 640 648 650 660 672 675 686 693 700 702 704 715 720 728 729 735 750 756 768 770 780 784 792 800 810 819 825 832 840 858 864 875 880 882 891 896 900 910 924 936 945 960 972 975 980 990 1000 1001 1008 1024 1029 1040 1050 1053 1056 1078 1080 1092 1100 1120 1125 1134 1144 1152 1155 1170 1176 1188 1200 1215 1225 1232 1248 1250 1260 1274 1280 1287 1296 1300 1320 1323 1344 1350 1365 1372 1375 1386 1400 1404 1408 1430 1440 1456 1458 1470 1485 1500 1512 1536 1540 1560 1568 1575 1584 1600 1617 1620 1625 1638 1650 1664 1680 1701 1715 1716 1728 1750 1755 1760 1764 1782 1792 1800 1820 1848 1872 1875 1890 1911 1920 1925 1944 1950 1960 1980 2000 2002 2016 2025 2048 2058 2079 2080 2100 2106 2112 2145 2156 2160 2184 2187 2200 2205 2240 2250 2268 2275 2288 2304 2310 2340 2352 2376 2400 2401 2430 2450 2457 2464 2475 2496 2500 2520 2548 2560 2574 2592 2600 2625 2640 2646 2673 2688 2695 2700 2730 2744 2750 2772 2800 2808 2816 2835 2860 2880 2912 2916 2925 2940 2970 3000 3003 3024 3072 3080 3087 3120 3125 3136 3150 3159 3168 3185 3200 3234 3240 3250 3276 3300 3328 3360 3375 3402 3430 3432 3456 3465 3500 3510 3520 3528 3564 3575 3584 3600 3640 3645 3675 3696 3744 3750 3773 3780 3822 3840 3850 3861 3888 3900 3920 3960 3969 4000 4004 4032 4050 4095 4096 4116 4125 4158 4160 4200 4212 4224 4290 4312 4320 4368 4374 4375 4400 4410 4455 4459 4480 4500 4536 4550 4576 4608 4620 4680 4704 4725 4752 4800 4802 4851 4860 4875 4900 4914 4928 4950 4992 5000 5005 5040 5096 5103 5120 5145 5148 5184 5200 5250 5265 5280 5292 5346 5376 5390 5400 5460 5488 5500 5544 5600 5616 5625 5632 5670 5720 5733 5760 5775 5824 5832 5850 5880 5940 6000 6006 6048 6075 6125 6144 6160 6174 6237 6240 6250 6272 6300 6318 6336 6370 6400 6435 6468 6480 6500 6552 6561 6600 6615 6656 6720 6750 6804 6825 6860 6864 6875 6912 6930 7000 7007 7020 7040 7056 7128 7150 7168 7200 7203 7280 7290 7350 7371 7392 7425 7488 7500 7546 7560 7644 7680 7700 7722 7776 7800 7840 7875 7920 7938 8000 8008 8019 8064 8085 8100 8125 8190 8192 8232 8250 8316 8320 8400 8424 8448 8505 8575 8580 8624 8640 8736 8748 8750 8775 8800 8820 8910 8918 8960 9000 9009 9072 9100 9152 9216 9240 9261 9360 9375 9408 9450 9477 9504 9555 9600 9604 9625 9702 9720 9750 9800 9828 9856 9900 9984 10000

Thick Metal

Using "Thick Up" or "Thick Down" metal will greatly increase the complexity of an EMPOWER run, as all metal layers must be duplicated (for the top and bottom of the thick metal) and z-directed currents must be added along the sides of all metal.

The detailed of defining metal layers is found in the EMPOWER layers dialog box, as follows:

Metal Layers - All metal layers from the General Layer Tab are also shown in the EMPOWER Layer tab. These layers are used for metal and other conductive material such as resistive film. The following types are available:

- Lossless: The layer is ideal metal.
- Physical Desc: The layer is lossy. These losses are described by Rho (resistivity relative to copper), Thickness, and Surface Roughness.
- Electrical Desc: The layer is lossy and is described by an impedance or file. This type is commonly used for resistive films and superconductors. If the entry in this box is a number, it specifies the impedance of the material in ohms per square. If the entry in this box is a filename, it specifies the name of a one-port data file which contains impedance data versus frequency. This data file will be interpolated/extrapolated as necessary. See the Device Data section for a description of one-port data files.
- Substrates: Choosing a substrate causes the layer to get the rho, thickness, and roughness parameters from that substrate definition. We recommend using this setting whenever possible so that parameters do not need to be duplicated between substrates and layouts.

Caution: Unless thick metal is selected, thickness is only used for calculation of losses. It is not otherwise used, and all strips are calculated as if they are infinitely thin.

Metal layers have three additional settings available:

Slot Type - Check this box to simulate the non-lossless-metal areas (as opposed to the metal areas) in EMPOWER. Use this for ground-planes and other layers which are primarily metal. Do not use this for lossy layers. See your EMPOWER manual for details.

Current Direction - Specifies which direction the current flows in this layer. The default is along X and Y. "X Only" and "Y Only" can be used to save times on long stretches of uniform lines. "Z Up", "Z Down", "XYZ Up", and "XYZ Down" allow the creation of thick metal going up/down to the next level or cover.

Thick Metal - Checking this box forces EMPOWER to model the metal including thickness. EMPOWER does this by putting two metal layers close together, duplicating the traces on each, and connecting them with z-directed currents. If thick metal is used, then Current Direction is ignored.

Element Z-Ports - This setting specifies the default direction for automatically created element ports, either to the level above or to the level below. Generally, you should choose the electrically shortest path for this direction.

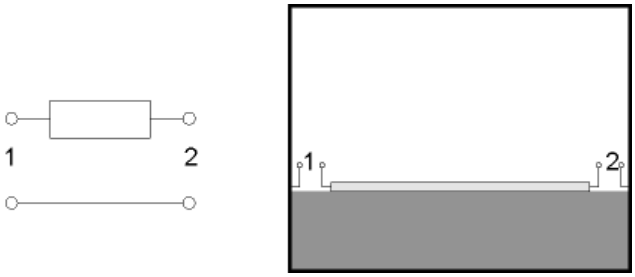
Chapter 21 EMPOWER: External Ports

Overview

Every EMPOWER circuit must contain at least one port. These ports are divided into two major categories: external ports which are at a sidewall, and internal ports which are inside the box. This section will cover only external ports; internal ports are discussed in a later section of this manual.

Placing External Ports

By now you should be familiar with the placement of external ports (EMPorts). If not, you should follow the first example To briefly review: An external port is placed in LAYOUT by selecting EMPort from the toolbar. These ports are generally placed on the edge of the box at the end of a line.

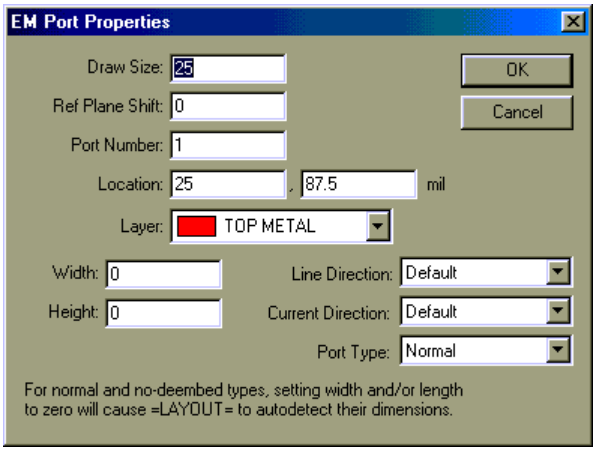


This figure shows a comparison between a port in circuit theory and a port in EMPOWER. In the circuit theory schematic on the left, there are two ports. Each port has two terminals, with the bottom terminal generally being ground.

In the EMPOWER illustration shown on the right the figure, the section of line stops before the edge of the box (generally one cell-width away) and a port begins in its place. (See the Grid discussion in the Basics section to see how this is mapped onto the grid.) As in the circuit theory schematic, there are two ports, and each port has two terminals. However, in EMPOWER, instead of the ground plane being modeled as a simple short circuit, the effect of currents traveling through the box is taken into account.

EMPort Options

When you first create a port, it is automatically configured to be an external port with the proper characteristics to be placed on the end of a transmission line. For many applications, you will want to modify these characteristics when you place the port. These characteristics are shown in the EM Port Properties dialog box which comes up automatically when the port is placed and which can be accessed later by either double-clicking on the port or by selecting the port and choosing Details from the Edit menu. A typical EM Port Properties dialog box is shown below. The following sections describe the entries in this dialog box.



Draw Size - This has no effect on the simulation. It controls the size that the port number appears on screen and on printouts.

Ref Plane Shift - This parameter is only available if "Port Type" is set to "Normal" (see below). On most complete circuits, this value can be left at zero. A positive Reference Plane shift causes the deembedding to add extra line length to the circuit; A negative value is more common and causes the reference planes to move inside the box. (See the Patch Antenna Impedance example for an example of a patch antenna simulation and the Edge-Coupled Filter example which uses a reference plane shift.) The reference plane is shown as an arrow on the layout. Additionally, when the EMPort is selected, Handles appear on the reference plane, allowing it to be moved with the mouse.

Port Number - When EMPOWER is run, the port numbers specified here correspond to the port numbers in the resulting data. These port numbers must be sequential (numbers cannot be skipped), and Normal ports must always have lower numbers than non-deembedded and internal ports. LAYOUT assigns a new port number automatically when an EMPort is placed, and the port number is displayed on the layout at the port.

Width & Length - When placing an external port on the end of a strip-type transmission line, you should normally leave these at zero so that LAYOUT sizes the port automatically. If you want to override the size, or for slot-type or internal ports, you can specify width and length here. Note: Width and length are measured relative to the line direction, so these parameters can appear to be reversed. Length is the length in the direction of propagation (along the line), and width is the width of the strip.

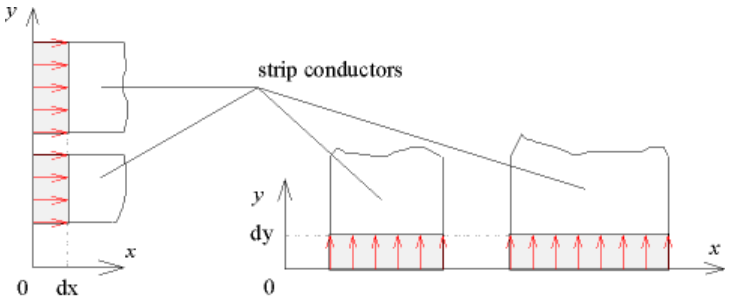
Layer - Specifies the metal layer on which the port is placed.

Location - specifies the edge of the port for external ports and the center of the port for internal ports.

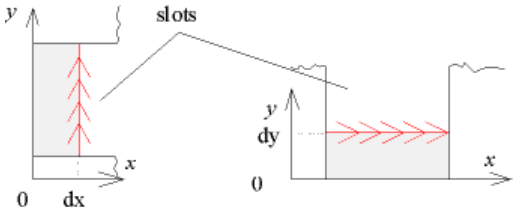
Line Direction - Gives the direction of the line at the port. In the default mode, the nearest wall determines the direction of the line. This value rarely needs to be overridden.

Current Dir - Specifies the direction of current flow within the port. The first figure below shows the default current direction for external ports on strip-type structures such as microstrip and stripline. The second figure shows the default current direction for external

ports on slot-type structures such as coplanar waveguide. For internal ports, the default current direction is "Along Z." This value also rarely needs to overridden.



A) Along X *B) Along Y*



A) Along Y *B) Along X*

Port Type - Specifies the basic type of port: Normal, No Deembed, and Internal.

- Normal ports are external ports which are deembedded and may be multi-mode. They are shown in gray on the layout.
- No Deembed ports are external ports which are not deembedded and cannot be multi-mode. They are shown in white on the layout.
- Internal ports are also not deembedded and cannot be multi-mode. They are shown in white on the layout.

For more information on dembedding and multi-mode lines, see below.

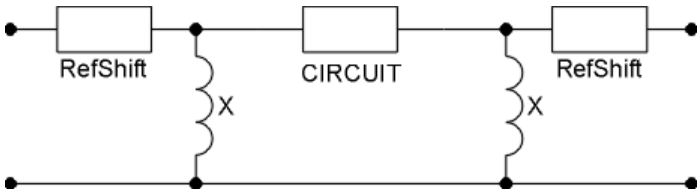
Deembedding

If you are actually building your circuit in the same style as an EMPort, that is, if your ports consist of a line which stops just short of the end wall, as is often the case with a coax-microstrip junction, then you may not need to use deembedding, because EMPOWER is simulating the circuit as you are actually going to build it.

However, you may not have this kind of construction or you may be simulating a small segment of a larger circuit. In an external port, there is capacitance at the port due to coupling from the open end of the line to the wall. Deembedding removes this extra

reactance, perfectly matching the transmission line, modeling it as though the line and box extend out to infinity.

Deembedding also allows you to define a reference plane shift. By default, the reference plane shift is zero, which means that the resulting data is measured at exactly the side wall. If the reference plane shift is negative, then the data is measured from inside the box, effectively subtracting length from the circuit. If the reference plane shift is positive, then the data is measured from outside the box, effectively adding length to the circuit.



This is the equivalent network used when deembedding is active. The center of the figure, labeled CIRCUIT contains the raw results from the EMPOWER simulation. Reactance X (shown as inductors above) cancels the capacitance caused by the end wall as well as correcting other reactances. (The value of X may be negative, and it is frequency dependent.) The RefShift lines at the outside move the reference planes to the correct location. Since the RefShift lines also help to correct for the discontinuity at the box wall, their lengths are normally not zero, even if the reference shift specified for the port is zero. The impedance of the RefShift lines is equal to the port line impedance, so only the phase is shifted by the addition of these lines. The magnitude of the reflection coefficients is not affected.

The parameters for deembedding are calculated prior to the analysis of the circuit. EMPOWER does this automatically by analyzing two different length lines at each frequency for each port used, solving for the reactance and the base RefShift value.

Note: Deembedding requires an additional line analysis mode at the start of the run, so runs using deembedding can take substantially longer.

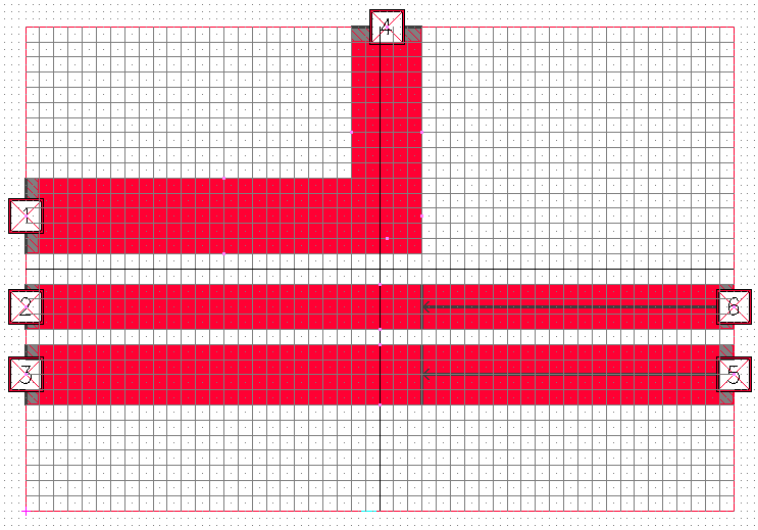
This is especially true if the lines at the ports are wide, since a wide line is simulated across the entire length of the box. However, line analysis is always symmetrical, and may be symmetrical in two planes if the port lines are placed exactly in the middle of the box. EMPOWER also caches the line analysis results, so if the box and port lines are not changed between runs, previous data will be used. The data for these lines are stored internally in the Workspace (WSP) file using internal files namedEMPOWER.R1, EMPOWER.R2, etc. EMPOWER also has the intelligence to detect when two or more ports have the same configuration (width, position, etc.), and will only run the line analysis once.

See Microstrip Line for a complete example which examines deembedding.

MultiMode Ports

Until now, all ports which we have looked at have been single mode ports. Single mode ports act just like regular nodes in SUPERSTAR, and external components can be added

directly to these ports. EMPOWER also supports external multimode ports where two EMPorts are close enough together that they are coupled. This circuit uses multimode ports, with ports 1, 2, and 3 being a 3-mode port, 4 being a normal single-mode port, and ports 5 and 6 being a 2-mode port:



Multimode ports have the following features:

- They much more accurately characterize the performance of a network with two or more lines close together on one wall.
- ***They cannot be used like normal SUPERSTAR nodes; They can only be connected to other multimode ports, including multi-mode lines and multimode EMPOWER data. Further, any multi-mode elements connected together must have the same number of modes for each port.***

Caution: Do not connect standard lumped elements to a multimode port! The results will not be correct. If you will be connecting directly to components, you should use single-mode ports. Use multi-mode ports only for connection only with other multi-mode ports and multi-mode lines.

- They can be used with decomposition to accurately analyze much larger structures than would be possible in a single EMPOWER circuit. See the Decomposition section for more details.

To create a multi-mode port, click on the "Mode Setup" Button from the EMPOWER setup dialog box when you start an EMPOWER run. You will see a box similar to the one at the end of this section. To make ports multi-mode, check the boxes between them. EMPorts 1,2, and 3 form one multi-mode port, and EMPorts 5 and 6 form another multi-mode port. EMPort 4 is a single mode port. To make a multi-mode port, you must follow these rules:

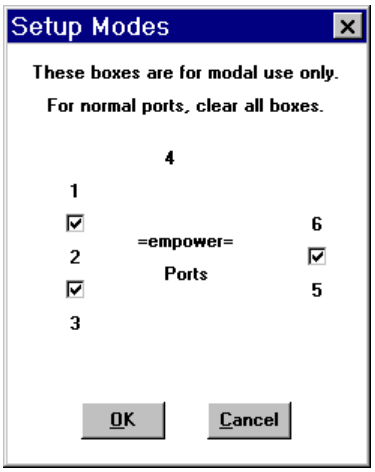
- All EMPorts for a multi-mode port must be on the same wall.
- All EMPorts must have the same length, line direction, current direction, and reference plane shift. The EMPorts may (and often do) have different widths, as above..
- All EMPorts must be “Normal” (not “No Deembed” or “Internal”).
- Port numbers must be sequential and in order. For example, if you swapped ports 1 and 2 above, you could not use a 3-mode port, because the ports would be in the order 2-1-3 along the sidewall.

Running the circuit above in EMPOWER will give 6-port data, as would be expected by glancing at the picture. However, the fourth port is the only normal, single-mode port. In the data file, the first three ports of data are in mode-space, and the last two ports of data are in mode-space. For example, in the data file:

- S41 represents the transmission of energy from mode 1 of multi-mode port (1,2,3) to port 4.
- S25 represents the transmission of energy from mode 1 of multi-mode port (5,6) to mode 2 of multi-mode port (1,2,3).
- S66 represents the reflection of energy in mode 2 of multi-mode port (5,6).

Multimode data should be carefully connected. Multimode ports should be connected only to other identical multi-mode port or line configuration (same box, line widths, spacings, etc.). Otherwise, the connection is non-physical and the results are meaningless.

See the Spiral Inductor example in the Decomposition section for more information on the use of Multimode lines.



Generalized S-Parameters

When normal circuit-theory analysis is performed, the ports are often terminated with a standard impedance such as 50 or 75 ohms. However, EMPOWER will give much more

accurate results if you use generalized S-Parameters. With generalized S-Parameters, instead of the ports being terminated with 50 or 75 ohms, the ports are terminated with the characteristic impedance of the line as calculated by EMPOWER. This is a more internally consistent representation, and the results are often far more accurate. You should use generalized S-Parameters if the following three conditions hold:

1. You are using normal, deembedded ports. Ports marked "No Deembed" or "Internal" are not appropriate for reporting generalized S-Parameters, so they are normalized to 50 ohms if generalized parameters are requested.
2. You have calculated the impedance of the lines at the ports (using T/LINE, for instance), and they are 50 (or 75) ohms.
3. You have run EMPOWER, but it calculated the port impedances to be a little different (for example, 47 instead of 50 ohms). This error is generally a result of the grid size. (A finer grid would result in less error in the impedance).

In this case, you know that your port lines should be 50 ohms, but EMPOWER reported 47 ohms. If you then request Generalized S-Parameters, GENESYS will also use 47 ohms for the terminating impedance, and a large part of the analysis error due to the grid will be cancelled. The results will be close to the results obtained if you measured the circuit in a 50 ohm network analyzer.

To get generalized S-Parameters from GENESYS, Check the "Generalized" box in the EMPOWER properties dialog box.

When EMPOWER is run, it outputs a file (in the structured storage when run from GENESYS) for each port with impedance data with extensions R1, R2, R3, etc., so, for a 2-port network in file EMPOWER analysis EM1, using "Generalized" impedance is equivalent to using an impedance of "WSP:Simulations\EM1\EMPOWER.R1, WSP:Simulations\EM1\EMPOWER.R2".

See the examples manual an example of the use of generalized S-Parameters.

Chapter 22 EMPOWER: Decomposition

Overview

In EMPOWER, it is possible to break down large circuits into smaller segments which are connected by transmission line sections. Decomposition can be tedious to implement, but its reward is that simulations can be performed accurately in much less time and with fewer frequency points. The principal benefits of decomposition are:

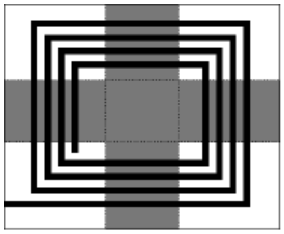
- Ability to tune single or coupled transmission line sections inside a circuit which was simulated by EMPOWER. For example, you can change the size of a meander line or adjust the tap point on an interdigital filter without rerunning the EMPOWER simulation.
- Most circuits require far fewer frequency points for accurate analysis. This is due to the fact that quarter-wave resonant lines are broken down into much smaller lines that do not resonate, and interpolation is possible. For example, a 7th order interdigital filter can often be simulated with just 5 frequency points in the EMPOWER run while 100 points are displayed in the output sweep!
- Ability to simulate problems too large to otherwise run.

The main disadvantages of decomposition are:

- Tedious to setup circuit. The simulation requires multiple EMPOWER runs combined with a schematic.
- Box modes and other phenomena related to the entire problem are not modelled. However, since EMPOWER uses mode space to model coupled line connections, this is less of a problem that it would be with other simulators.
- Losses in the connecting lines are not modelled.

Basics

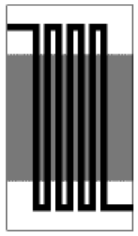
Decomposition can be applied to circuits with parts which are connected via single or multiple transmission lines. Some typical circuits which can be broken apart are shown below. In each of the circuits, the unshaded areas are simulated individually. The pieces are then combined using multi-mode transmission lines to connect the pieces, representing the lines in the shaded area.



Spiral Inductor

Edge Coupled Filter

Interdigital Filter



Meander Line

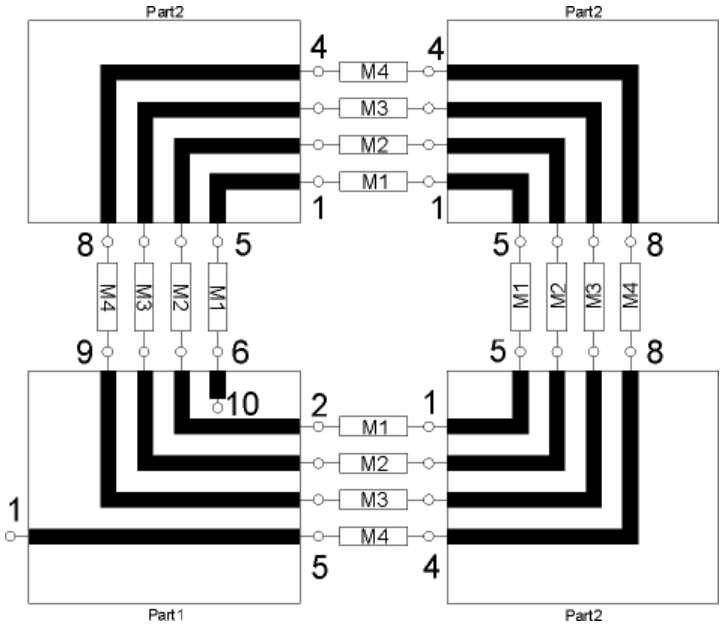
For decomposition to be possible, you must be able to break the circuit down into rectangular areas which are interconnected with transmission lines. For example, the spiral inductor above is broken down into four rectangular areas, one for each corner. These sections are then connected with multi-mode transmission lines.

In each of the circuits above, the three main advantages of decomposition can be seen:

- The lengths of the connecting transmission lines can be varied. In the spiral inductor, this allows the size of the spiral (and the inductance) to be tuned or optimized in GENESYS.
- Far fewer points need to be analyzed. This is because each of the pieces is simpler and interpolation works well. For example, in the edge coupled filter, each of the pieces contain only open ends and small sections of lines which do not resonate. As a result, this filter only needed 5 frequency points for a good analysis.
- With any of these circuits, The grey areas can easily get so large that the problem requires hundreds of megabytes to analyze. In the meander line, if the lengths of the coupled lines (grey areas) gets very long, the EMPOWER simulation could take a long time. When the circuit is decomposed, simply changing one length value in GENESYS gives a virtually instant analysis, no matter how long the coupled sections are.

Spiral Inductor Example

As a first decomposition example, we will analyze a spiral inductor . The first step is to come up with a plan for decomposition as shown here. We strongly recommend that you write a similar plan on paper when you setup a problem for multi-mode analysis.



The first step is to create workspace with a layout for each unique piece. In this example, there are two unique pieces: The lower left corner is the first, and each of the other three corners which are identical. There are two basic methods for creating these pieces:

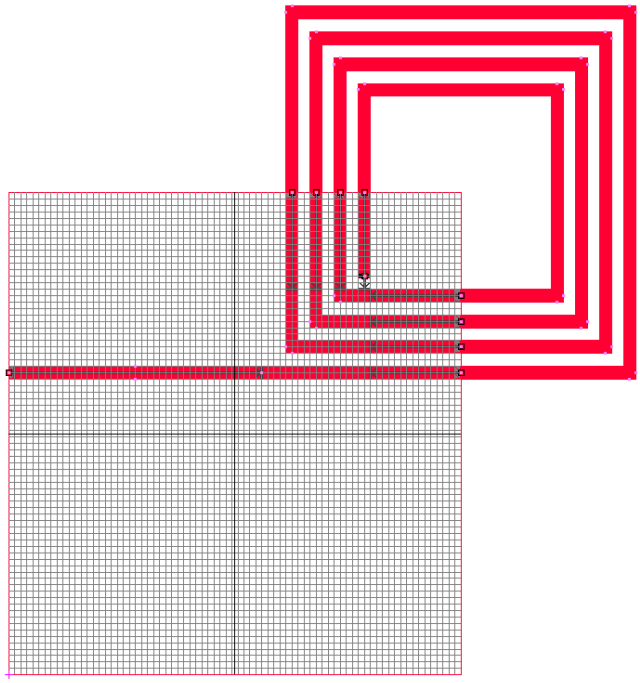
- Create the pieces individually, drawing only the part that will be simulated in each piece. In this case, each individual layout will look like the parts shown above.

Or,

- Create a complete layout of the entire problem first. Then, make the box smaller so that only the desired piece is simulated. This is the method we will use for the spiral.

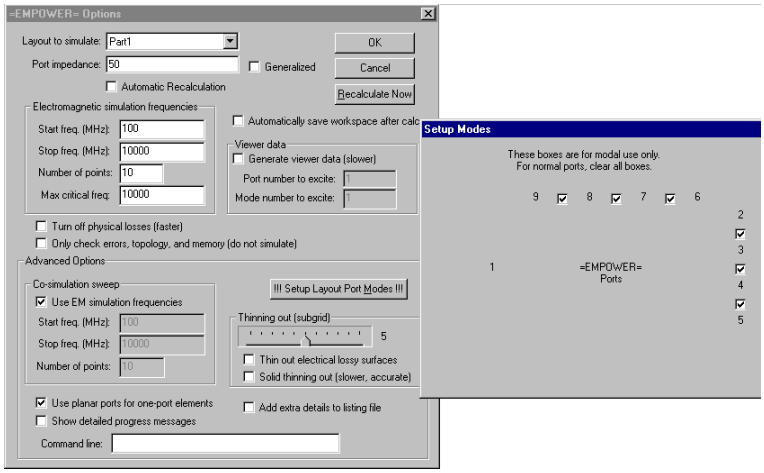
We have created a layout of the entire spiral inductor as a starting point (\EAGLE\EXAMPLES\DECOMP\FULL.WSP). This file was created by starting with an MRIND element so that the layout was created mostly automatically. The only addition was the extra length leading to port 1 and the EMPorts. Notice that the reference plane for port 1 is shifted to the actual start point of the spiral model. Port 2 is an internal port. This circuit can be analyzed directly, but it requires minutes per frequency point and 37 megabytes of RAM.

This file was then saved as COMBINE.WSP. The box was shrunk and the circuit was moved so that only the bottom left quarter of the circuit is in the box. The number on the internal port on the end of the spiral was changed to 10. Ports 2-5 on the right and 6-9 were added. Since these ports are in the middle of a line instead of on the end, their width must be set manually. Also, the reference planes on the ports were shifted in. The resulting layout for the first piece is shown below.



EMPOWER was run for Part1. The settings are as shown below. *Note that only 5 points are needed since the individual parts are not resonant.* The "Setup Layout Port Modes" button was clicked, and the checkboxes in the Setup Modes dialog box were set to indicate that those inputs are modally related.

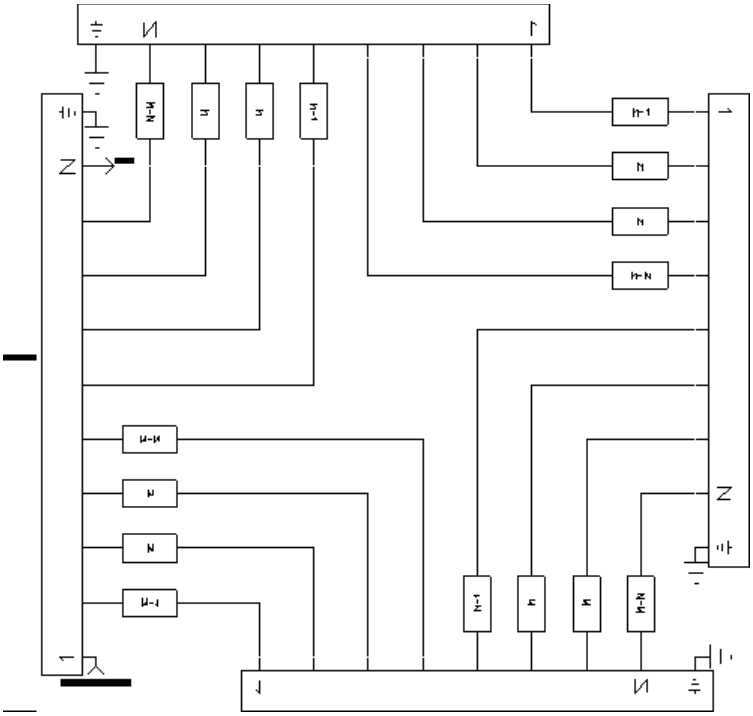
> **Caution:** Do not forget to setup the modes when you are analyzing by decomposition. The Mode Setup box turns red if any inputs are modally related. Improper mode setup is one of the most common errors in decomposition.



A similar set of steps was followed for Part2.

The final step in decompositional analysis is to combine the pieces. The Schematic (COMBINE) which does this is shown here. The pieces used are NPO10 and NPO8 blocks (under "DEVICE" in SCHEMAX) for the data in PART1 and PART2 plus MMTLP8 models (multi-mode physical transmission lines, found under "T-LINE") for the interconnecting lines. The data for the NPO10 is in WSP:Simulations\EMPart1\EMPOWER.SS, and the data for the NPO8 is in WSP:Simulations\EMPart2\EMPOWER.SS.

Note: Some users may find it easier to write a text Netlist to combine the pieces. At Eagleware, we find it easier to use SCHEMAX for this purpose, but you may use whichever you feel most comfortable with.

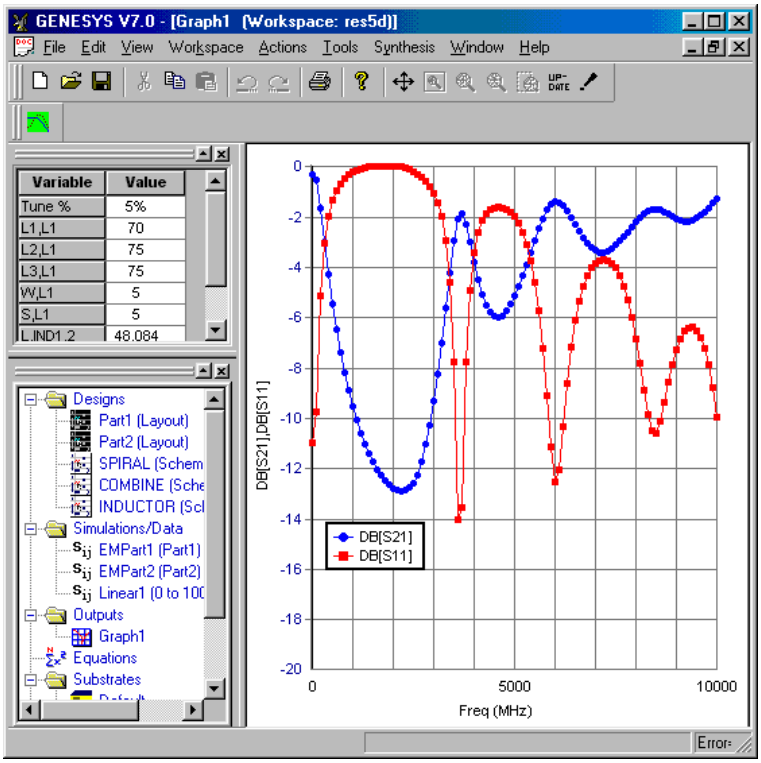


Whenever deembedded ports are used, data files suitable for the SMTLP and MMTLP models are automatically created during the LINE portion of the EMPOWER run. For the MMTLP8 lines, the file WSP:Simulations\EMPart1\EMPOWER.L2 was used. This corresponds to the second set of inputs for PART1. You should view the listing file (Right-click on EMPart1) and look at the port numbers to determine which EMPOWER.L* file contains the line data you need.

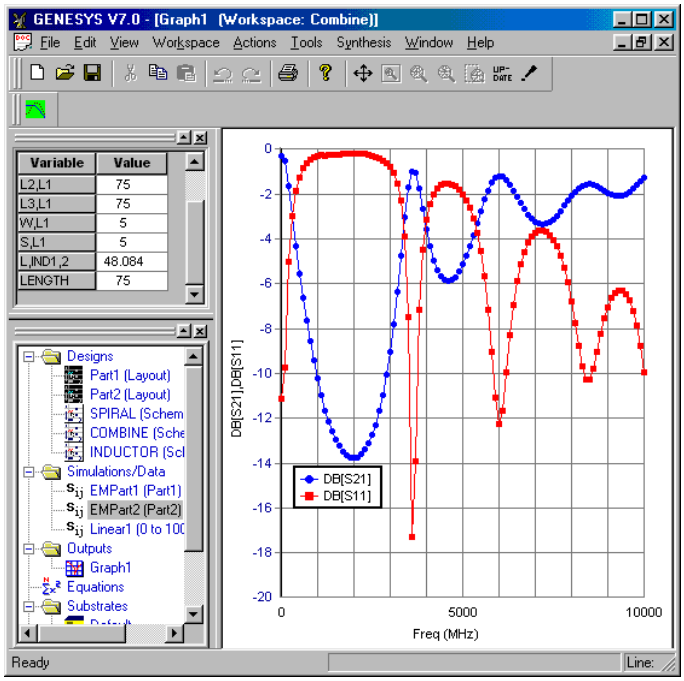
Note: Files with names like WSP:Simulations\EMPart1\EMPOWER.L2 are taken from within the current workspace. For a complete explanation of how these files are names, see the File Formats section in this manual.

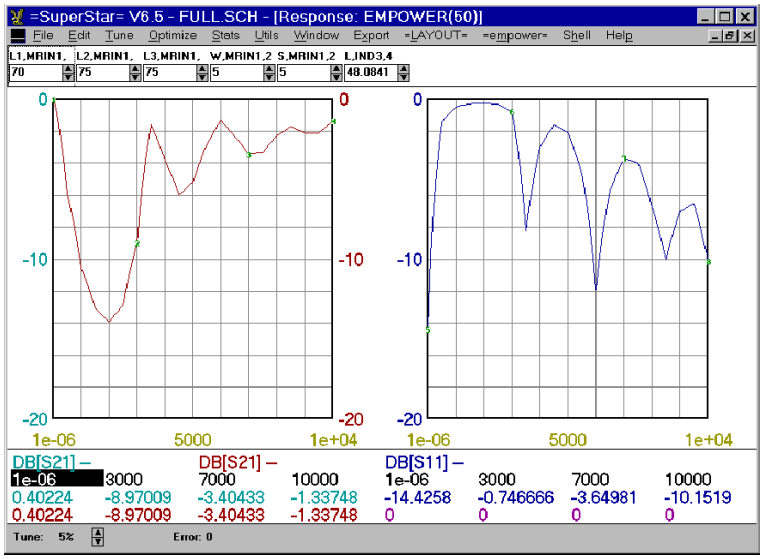
The substrate must also be specified, but only the UNITS parameter is used by the MMTLP8 model.

A variable was setup (LENGTH) so that the lengths of line can be tuned in GENESYS simultaneously, changing the size of the spiral and thus the inductance very quickly.



The results from this are shown above. Notice that even with only 5 analysis points across the band, the interpolation is very good. To illustrate this, the spiral inductor was recalculated with 10 points below. You can see quite good agreement between the two. To test the validity of the decompositional analysis, the entire spiral was analyzed, and the results are given in the second figure below. This full analysis took hours on a 266MHz Pentium II, and if the lengths of the lines in the spiral are changed, it must be rerun.





Losses

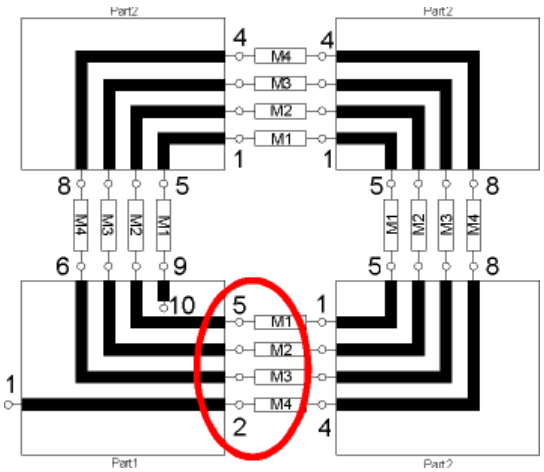
A current limitation of decomposition is that losses are not taken into account in multi-mode transmission line sections or in reference-plane shifts. For the spiral inductor, this means that the losses as calculated are accurate for the nominal dimensions, but any modification to the lengths using the multimode lines will not affect the calculated loss.

In general, if the decomposed pieces cover the circuit completely (as is the case in the spiral inductor), then the losses will be accurate. If the pieces do not completely cover the circuit (if sections of line are left out of the EMPOWER analysis and are added with MMTLP sections, then the losses will not include these sections. This is true regardless of the reference plane shifts used, since these shifts do not affect the loss.

Port Numbering

You must be very careful when setting up and numbering ports for decompositional analysis. The following rules must be followed:

- Never connect anything other than MMTLP lines or other identical modal inputs to inputs which are modally related. Connecting lumped elements to modal inputs is incorrect and will give bad results.
- Ports which will be modally related must have sequential numbers. They must also all have the same reference shift.
- Ports for mode-space inputs must be marked type "Normal," not "No deembed" or "Internal." Correspondingly, their numbers must be lower than any "No-deembed" or "Internal" ports.
- The order of ports used must correspond between the pieces and the MMTLP lines used. The lowest port number in a modally related set of inputs should connect to Mode 1 in the MMTLP line, and the highest port number in the set should connect to Mode N on the MMTLP line. Also, port ordering should be exactly the same on both pieces connected through the MMTLP. ***The figure below shows an incorrect numbering of the spiral inductor.*** In this example, PART1 and PART2 are inconsistently numbered, since on PART1 the outermost inputs (numbers 2 and 6) are the lowest number while in PART2 the innermost inputs (numbers 1 and 5) are the lowest number.



- Pieces can be connected directly together without using MMTLP. In this case, the lowest ports in each modally related set of inputs are connected to each other.

Chapter 23 EMPOWER: Lumped Elements and Internal Ports

Overview

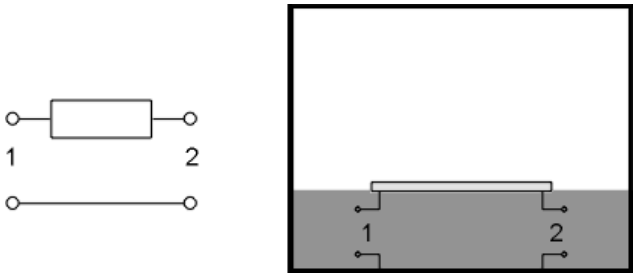
As described in the External Ports section, every EMPOWER circuit must contain at least one port. This section will cover lumped elements and internal ports (ports inside the box). External ports (along a sidewall) were also covered in that section.

Placing Internal Ports

The process of placing an internal port is similar to the process of placing an external port. To summarize: An internal port is placed in LAYOUT by selecting EMPort from the toolbar. Internal ports can be placed anywhere in the box. When the EMPort Properties dialog box appears, first select "Internal" in the Port Type combo box. Next, fill in the width and length of the pad. Press OK to complete the placement.

> **Note:** The rest of the options in the EMPort Properties dialog box were covered in the section entitled "Port Options." You may want to review these options now.

The figure below shows a comparison between ports in circuit theory and internal ports in EMPOWER. In the circuit theory schematic on the left, there are two ports. Each port has two terminals, with the bottom terminal generally being ground.



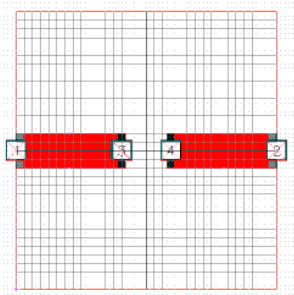
In the EMPOWER illustration there are two z-directed ports, one at each end of the line. These z-directed ports are mapped onto the grid along Z, much in the same way as a viahole would be mapped. (See the Basics section for more information on mapping to the grid.) As in the circuit theory schematic, there are two ports, and each port has two terminals. The bottom terminals, which are true ground in the circuit schematic, are connected to the bottom wall (ground plane), a physical representation of ground.

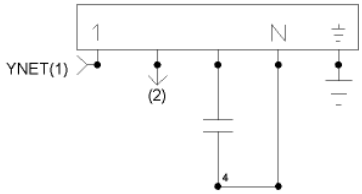
Z-directed internal ports can be used in GENESYS to connect elements, just like a node in SCHEMAX or in a text file. In other words, components like resistors and transistors can be connected directly to these ports. You simply place a z-directed port in the center of the pad for the component in these cases. Note: SCHEMAX does this automatically as is described later in this section.

Manually Adding Lumped Elements

Note: GENESYS will automatically add lumped elements to your simulation if components are on your layout. This section is for background information and advanced applications

The circuit shown below contains an EMPOWER circuit which was drawn completely in LAYOUT. (The schematic for this network was blank.) It has 4-ports; ports 1 and 2 are external, and ports 3 and 4 are internal. EMPOWER will create a 4-port data file for this circuit.





Note: Internal ports and "no-deembed" ports must always have higher numbers than normal, external, deembedded ports. In the figure above, the internal ports are numbered 3 and 4, while the external ports are numbered 1 and 2.

The data file created by EMPOWER can then be used in GENESYS. The circuit on the right above uses the resulting data in a complete network. First, a FOU (four port data) device was placed on the blank schematic. The name assigned to this FOU block was the name of the internal file from the EMPOWER run ("WSP:Simulations\EM1\EMPOWER.SS").

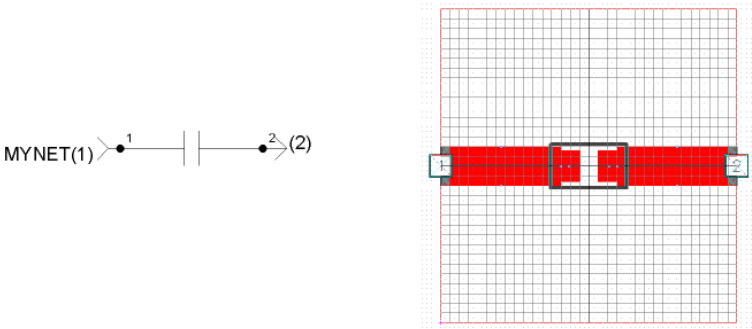
An input and output were added on nodes one and two of the FOU block, the ground was added to the ground node, and a capacitor was connected across ports 3 and 4. This has the effect of putting the capacitor "into" the EMPOWER simulation. This capacitor can then be tuned and optimized, just like any other element in GENESYS.

When the S-Parameters of MYNET are displayed, you see the resulting S-Parameters of the entire circuit.

Automatic Port Placement

One advantage of EMPOWER is its true integration. In most electromagnetic simulators, you would have no choice but to go through the complicated steps above. (Imagine how tedious this would be if you had 10 lumped elements, 2 transistors, and an op-amp chip in the box!) Fortunately, when EMPOWER is combined with SUPERSTAR, SCHEMAX, and LAYOUT, the internal ports and lumped elements can be generated and added automatically.

The circuit below uses automatic port placement. Initially, the circuit on the left is drawn in SCHEMAX. The layout on the right of the figure was then created: The footprint for the chip capacitor was automatically placed. The lines and EMPorts were then manually added. When EMPOWER is invoked, internal ports are automatically added, so the circuit simulated is virtually identical to the one on the left below, and the result is a 4-port data file.



EMPOWER then automatically creates a network which is identical to the network shown in the previous section. This result is fundamentally the same as the result from MYNET below. When the capacitor below is tuned or optimized, the networks MYNET and EMPOWER are both updated simultaneously.

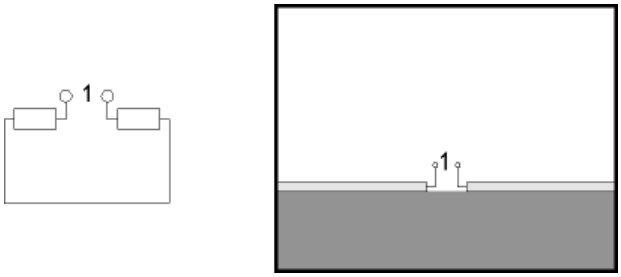
Even if you create a file with a layout only (no schematic), you can still use automatic port placement. Simply put the parts down onto a blank schematic, connecting them into a dummy network. The parts will now show up in LAYOUT and can be moved as needed, ignoring any rubber bands. (The rubber bands come from the meaningless connections in the dummy network.) When you display the EMPOWER simulation results, it will include the components. You do not need to display the results from the schematic.

Planar (X- and Y-Directed) Ports

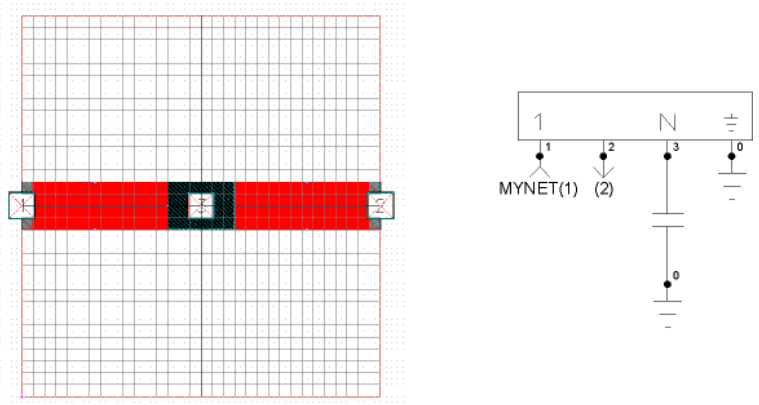
Note: EMPOWER will create planar ports for lumped elements if the "Use Planar Ports for one-port elements" box is checked in the EMPOWER options dialog. See your reference manual for details.

In some situations, you may want to place internal ports with X- or Y-directed currents. These ports are much trickier to use manually, since they are not referenced to ground. For components in your layout, EMPOWER will automatically place planar port and lumped elements, so this section is for background or advanced applications.

This figure shows the configuration of these ports. These ports can be more accurate for manually connecting lumped elements to EMPOWER data since the ports are a more accurately represent the physical connection of lumped elements.



The circuit shown in below contains an EMPOWER circuit which was drawn completely in LAYOUT. (The schematic for this network was blank.) It has 3-ports; ports 1 and 2 are external, and port 3 is internal with current direction "Along X." EMPOWER will create a 3-port data file for this circuit, however, you must be aware that port 3 will be a series connected port and ***cannot be used in the normal manner.***



The data file created by EMPOWER can then be used in GENESYS as described in the previous section using "WSP:Simulations\EM1\EMPOWER.SS". The circuit on the right uses the resulting data in a complete network. First, a THR (three-port data) device was placed on the blank schematic, using the EMPOWER.SS file from the EMPOWER run. An input and output were added on nodes one and two of the THR block, the ground was added to the ground node, and a capacitor was connected from port 3 to ground. This has the effect of putting the capacitor across port 3 in the EMPOWER simulation. The rules to follow for "Along X" and "Along Y" internal ports are simple:

- Do not attempt to use them for transistors or other 3-terminal (or more) devices.
- Set the Current Direction of the EMPort to "Along X" (along the x-axis) if the current along the component flows from left to right, as on the layout on the left above. Set the Current Direction of the EMPort to "Along Y" (along the y-axis) if the current along the component flows from top to bottom, as if the capacitor were turned 90 degrees from the one on the layout above.
- Connecting a lumped element in SCHEMAX from the port to ground when you use the resulting data is equivalent to connecting the lumped element accross the

length of the port in the LAYOUT. ***This does not mean that the component is grounded. It simply means that the component is connected accross the port.*** This concept is key to understanding X- and Y- directed ports.

When the S-Parameters of MYNET are displayed in a graph, you see the resulting S-Parameters of the entire circuit..

Resonance

Often, when a circuit contains lumped elements, you can use very few frequency points for the EMPOWER runs. Since the lumped elements are not included in the EMPOWER data, there are generally many fewer resonances, and the data interpolates much more accurately. In this case, you may want to only use 2 or 3 points in the electromagnetic analysis while showing the results of the entire network with 100 points or more (specified in the Co-Simulation Sweep in the EMPOWER Options Dialog box).

For a complete example which takes advantage of this property, see the Narrowband Interdigital example.

Chapter 24 EMPOWER: Viewer and Antenna Patterns

Overview

This section describes how to launch the EMPOWER viewer program and how to use it to visualize and interpret currents (or voltages) generated by EMPOWER. It also describes the viewer interface.

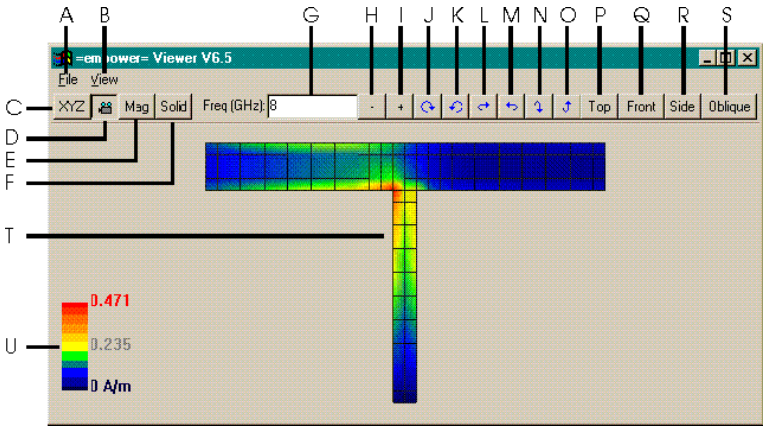
The EMPOWER viewer helps you visualize current distribution and densities in a board layout. It processes current density magnitude and angle and plots them as two or three dimensional static or dynamic graphs. These plots provide insight into circuit behavior and often suggest modifications which improve the performance. Most electromagnetic simulators include visualization tools. The EMPOWER viewer has distinct advantages such as three dimensional graphs, true animation capabilities, and precise information about current phase. The full potential of the EMPOWER viewer is realized with practice so we encourage you to investigate your circuits with the viewer and reflect on the results you observe.

The viewer is started by selecting Run Viewer from the right-click menu of an EMPOWER simulation (Workspace Window in GENESYS)

Interface

This section describes the viewer menu items and buttons. It can be used to become acquainted with the interface in general as well as as a reference section.

A sample viewer screen is shown below. The objects in this figure are described below.



A - File Menu

Open - Opens a new viewer data file.

Exit - Exits the viewer.

Toggle Background Color - Toggles the background from black to white, or white to black. A white background is normally selected before a screen or window print.

Print Screen - Sends a copy of the entire screen to a bitmap file or to a printer.

Print Window - Sends a copy of the viewer window to a bitmap file or to a printer.

B - View Menu

The objects in this menu affect how the current image is displayed.

Top (Home) - Shows a top-down view of the current image. This option can also be selected by pressing Home.

Front (Ctrl+Home) - Shows a "front" view of the current image. This view is from the y-axis, at z=0. This option can also be selected by pressing Ctrl+Home.

Side (Ctrl+End) - Shows a "side" view of the current image. This view is from the x-axis, at z=0. This option can also be selected by pressing Ctrl+End.

Oblique (End) - Shows an oblique view of the current image. This view is top-down on the x-y plane with a slight offset. This option can also be selected by pressing End.

Rotate - The objects in this sub-menu rotate the current image.

Rotate - Left (Left Arrow) - Rotates the current image clockwise in a horizontal plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

Rotate - Right (Right Arrow) - Rotates the current image counter-clockwise in a horizontal plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

Rotate - Up (Up Arrow) - Rotates the current image forward in a vertical plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

Rotate - Down (Down Arrow) - Rotates the current image backward in a vertical plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

Rotate - Clockwise (PgDn) - Rotates the current image clockwise in the plane of the screen. The center of the viewer image window is always the center of rotation. This option can also be selected by pressing Page Down.

Rotate - Counter-Clockwise (PgUp) - Rotates the current image counter-clockwise in the plane of the screen. The center of the viewer image window is always the center of rotation. This option can also be selected by pressing Page Down.

Pan - The objects in this sub-menu shift the apparent location of the viewer window relative to the current image.

Pan - Left (Ctrl+Left) - Moves the viewer location to the left (relative to the current image). This moves the image to the right in the viewer window.

Pan - Right (Ctrl+Right) - Moves the viewer location to the right (relative to the current image). This moves the image to the left in the viewer window.

Pan - Up (Ctrl+Up) - Moves the viewer location up (relative to the current image). This moves the image down in the viewer window.

Pan - Down (Ctrl+Down) - Moves the viewer location down (relative to the current image). This moves the image up in the viewer window.

Pan - Zoom In (Ctrl+PgUp) - Moves the viewer location closer to the current image. This increases the size of the image in the viewer window.

Pan - Zoom Out (Ctrl+PgDn) - Moves the viewer location away from the current image. This decreases the size of the image in the viewer window.

Toggle - The objects in this sub-menu toggle the available options listed below.

Toggle - Absolute Value Display - When selected, the viewer displays absolute values only. If not selected, an actual value with information about flowing direction is displayed. The difference is that absolute value is always positive, whereas the actual current values can be positive for forward directed currents and negative for backward directed currents. Negative amplitudes are drawn below the x-y plane. This option has a checkmark beside it when selected.

Toggle - Animation - When selected, the viewer "animates" the image in real or angle mode. This is accomplished by multiplying the individual currents by exp(jw), where w cycles from 0 to 2pi and showing a sequence of snapshot images for increasing w. This option has a checkmark beside it when selected.

Toggle - Scale - When selected, the viewer displays the scale in the lower-left of the viewer window. This option has a checkmark beside it when selected.

Toggle - Value Mode (Real,Mag,Ang) - This option selects the current display option. The options include the Real current value for current distribution snapshots and animation, Magnitude for time averaged current values, and Angle for the current phase delay distribution snapshots.

Toggle - Wireframe - When selected, the viewer displays a wireframe version of the current plots. A wireframe is created by drawing the outlines of the EMPOWER grid currents without filling the resulting polygons. When this option is not selected, the viewer fills the polygons, resulting in a solid surface plot of the current patterns. This option has a checkmark beside it when selected.

Load - From User View #(1-10) - Loads the previously saved viewer settings for the selected view #. Saved settings can also be restored by pressing the number key corresponding to the desired setting #.

Save - To User View #(1-10) - Saves the current viewer settings into the selected view #. The settings can be restored later by selecting the desired # from the load sub-menu described above. The options in this menu can also be selected by pressing Shift + the number key corresponding to the desired save #.

Tip: The save and load functions are extremely useful. If you rotate and pan to a view that you like, press Shift plus a number (not an arrow) to save that view. Simply press the number by itself to return to that view. These views are remembered even if you exit the viewer, so you can easily store your favorite views.

C - X / Y / Z / XY Button

Pressing this button toggles between the four possible modes:

X - Displays the x-directed current density distribution.

Y - Displays the y-directed currents density distribution.

Z - Displays the z-directed currents.

XY - Displays additive surface current density distribution function.

D - Animate Button

This button toggles viewer "animation" on the current image. When this option is selected, the button appears pressed. The viewer animation is accomplished by multiplying the individual currents by exp(jw), where w cycles from 0 to 2pi and plotting snapshot graphs for sequential time moments. What is animated is controlled by the Display Option Button (see E below).

E - Display Option Button

This button selects the current display option.

Real - Displays the real portion of the current values.

Mag - Displays the magnitude or time averaged values of the currents.

Ang - Displays the phase delay of the current values.

F - Solid/Wire Button

This button toggles the type of surface plot to display.

Wire - Displays a wireframe version of the current patterns. A wireframe is created by drawing the outlines of the EMPOWER grid currents without filling the resulting polygons.

Solid - Displays a solid surface plot of the current patterns. This is created by filling the wireframe polygons.

G - Freq (GHz)

This box shows the simulation frequency (in GHz) for which the current image data is being displayed. This box is restricted to frequencies that EMPOWER has created data for. The value can be increased by clicking the "+" button (see I below), and decreased by clicking the "-" button (see H below).

H - Decrease Frequency Button

Decreases the current frequency (see G above). If you are already at the lowest calculated frequency then this button has no effect.

I - Increase Frequency Button

Increases the current frequency (see G above). If you are already at the highest calculated frequency then this button has no effect.

J - Clockwise Button

Rotates the current image clockwise in the plane of the screen. The center of the viewer image window is always the center of rotation. This option can also be selected by pressing Page Down.

K - Counter-Clockwise Button

Rotates the current image counter-clockwise in the plane of the screen. The center of the viewer image window is always the center of rotation. This option can also be selected by pressing Page Down.

L - Rotate Right Button

Rotates the current image counter-clockwise in a horizontal plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

M - Rotate Left Button

Rotates the current image clockwise in a horizontal plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

N - Rotate Down Button

Rotates the current image backward in a vertical plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

O - Rotate Up Button

Rotates the current image forward in a vertical plane perpendicular to the screen. The center of the viewer image window is always the center of rotation.

P - Top Button

Shows a top-down view of the current image. This option can also be selected by pressing the Home key.

Q - Front Button

Shows a "front" view of the current image. This view is from the y-axis, at z=0. This option can also be selected by pressing Ctrl+Home.

R - Side Button

Shows a "side" view of the current image. This view is from the x-axis, at z=0. This option can also be selected by pressing Ctrl+End.

S - Oblique Button

Shows an oblique view of the current image. This view is top-down on the x-y plane with a slight offset. This option can also be selected by pressing End.

T - Current Plot

Shows the color-coded current patterns for the loaded EMPOWER generated viewer data file. The menus and toolbar buttons control how this image is displayed.

U - Color Scale For Current Plot

This scale shows the relative current and current density magnitudes based on the color used to draw the plot patterns.

Far-Field Radiation Pattern Viewer

The EMPOWER far-field radiation data describes the electric field patterns in the far-zone region radiated from a structure. The far-zone is defined as the region where $2\pi R/\lambda >> 1$, where R is the distance from the structure and lambda is the wavelength of the signal exciting the structure. Far-field radiation patterns are described in the spherical coordinate system, where phi is the angle on the xy plane from the positive x-axis, and theta is the angle from the positive z-axis. The distance is not specified since it is assumed to be in the far-zone.

Assumptions Made when Generating Far-Field Radiation Data

Data for radiation in the far-field is generated using equations that make simplifying assumptions about the layout of the structure. It is, therefore, necessary to take these assumptions into account and follow them to get accurate solutions:

- The walls of the box are assumed to be infinitely far away from the structure.
- If a substrate is used, it is also assumed to extend infinitely in the lateral dimensions.
- Fields generated from z-directed currents are not taken into account, therefore it is not recommended that you include vias in the layout.

Setting Up the EMPOWER Box

To get good results for the far-field radiation patterns, the following rules must be observed:

- The structure should be centered in the box.
- The walls of the box should be far away from the structure.
- Only one layer of metal must be used.
- Exactly one substrate or an Air-Below layer must be under the metal layer, not both.

There are 3 different antenna types for which far-field radiation patterns can be generated:

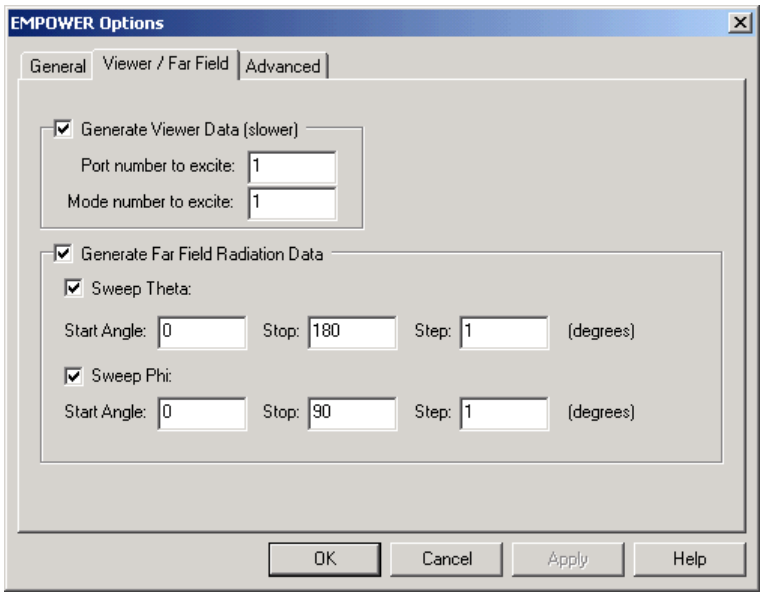
- Antenna in free-space

- Antenna above a ground plane
- Microstrip antenna above a substrate and ground plane

To simulate an antenna in free space, no substrate should be used and the only layer below the metal layer should be Air-Below. The height of the Air-Below layer in this case is irrelevant. Both the Top Cover and Bottom Cover should be set to Electrical type, with surface impedance set to 377 ohms. (377 ohms is the intrinsic impedance of free-space).

To simulate an antenna above a ground plane with no substrate, the Air-Below layer should be set to the height the antenna is to be above the ground plane. The Bottom Cover should be set to Lossless type, and the Top Cover should be set to Electrical type, with surface impedance set to 377 ohms.

To simulate a microstrip antenna, the Air-Below layer should not be used. The substrate layer, instead, should be used. The Bottom Cover should be set to Lossless type, and the Top Cover should be set to Electrical type, with surface impedance set to 377 ohms.



Specifying Sweep Parameters

In order to generate far field radiation data, *Generate Viewer Data (slower)* and *Generate Far Field Radiation Data* must be checked. You may then select either Theta, Phi, or both, to be swept. Data is generated for all points between Start Angle and Stop Angle for both Theta and Phi, with a step size specified in the Step field. All angles are in degrees.

In the above figure, data is being generated sweeping both Theta and Phi. Theta is being swept from 0 to 180 degrees in 1 degree increments, while Phi is being swept from 0 to 90 degrees, also in 1 degree increments.

Measurements and Plotting

Once far-field radiation data is generated, the following measurements can be plotted:

ETHETA(*phis, thetas, freqs*) - the theta component of the total electric field. *Phis*, *Thetas*, and *freqs* can either be single values or ranges of values.

EPHI(*phis, thetas, freqs*) - the phi component of the total electric field. *Phis*, *Thetas*, and *freqs* can either be single values or ranges of values.

ETOTAL(*phis, thetas, freqs*) - the magnitude of the total electric field. *Phis*, *Thetas*, and *freqs* can either be single values or ranges of values.

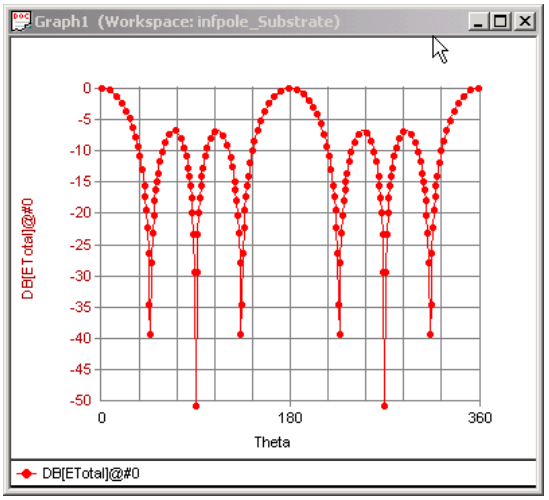
ELHCP - E-field Left Hand Circular Polarization

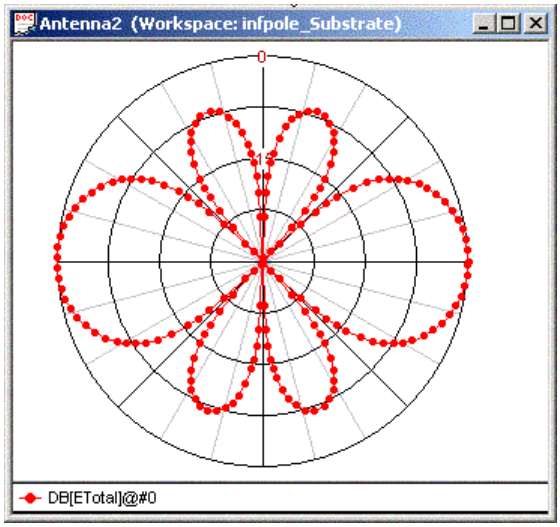
ERHCP - E-field Right Hand Circular Polarization

EAR - E-field Axial Ratio

The measurement wizard can be used to to select these measurements, and the proper syntax is automatically generated.

Rectangular, Antenna (Polar), and 3D charts may be generated to display the antenna data. Only one variable out of Phi, Theta, and Frequency may be swept when displayed on the two-dimensional charts, and two variables may be swept when displayed on the 3D chart. Below is both a rectangular and Antenna plot (polar) of the ETOTAL measurement, where Theta is being swept from 0 to 360 degrees, Phi is held constant at 0, and the frequency is held constant. This particular antenna is a very small dipole located one wavelength above a ground plane, on top of a substrate.



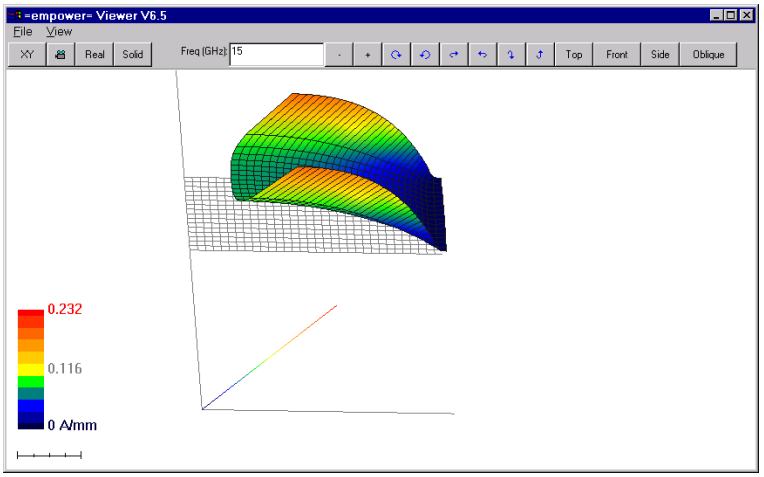


Examples

This section illustrates the use of the EMPOWER viewer using a number of examples. The WSP files for the examples are located in the subdirectory PROGRAM FILES\GENESYS\EXAMPLES\VIEWER. You may load them as you follow along with the following examples.

The viewer displays current distributions as two or three dimensional graphs. The viewer has several modes that are used to view various components of the currents from different view perspectives. The best view of most problems is often found by minor adjustments of the view orientation. The following examples include a few examples of such adjustments. The examples are simple problems selected because the results are predictable. Nevertheless, they are interesting and illustrate concepts which may be applied to more complex problems.

Consider the possible graphs for a simple line segment analysis. The schematic file for this example is METR16.WSP. It contains description of a segment of the 50-Ohm standard stripline [Rautio, 1994] that is also discussed in the Examples Chapter. The segment is 1.4423896 mm wide by 4.996540 mm long and the box size along the z-axis is 1 mm. The segment length is 90 degrees at 15 GHz and 180 degrees at 30 GHz. Load METR16.WSP in GENESYS. Run the viewer by selecting “Run EMPOWER Viewer” from the right-click menu of Simulation “EM1". The default plot seen in the main window is an animated surface electric current density distribution function reflecting the surface currents in the strip plane. At the initial time t=0 it will look similar to the graph shown below.



To get this snapshot we stopped animation by clicking the Animation camera icon, adjusted the view slightly, and toggled the background color to white. To obtain this view, simply press the Oblique button on the toolbar after starting the viewer. All other settings are the default:

- Show XY current density distribution (XY/X/Y/Z button).
- Show Real part of the current density distribution (View Menu/Switches/Value Mode or Value Mode button).
- Show Absolute values of the current density quantities (View Menu/Switches/Absolute Value Display).
- Animation is off and time is set to initial (View Menu/Switches/Animation or Animation Camera button).
- Scale is on (View Menu/Switches/Scale)
- Solid polygons view (View Menu/Switches/Wireframe or Solid/Wire button)

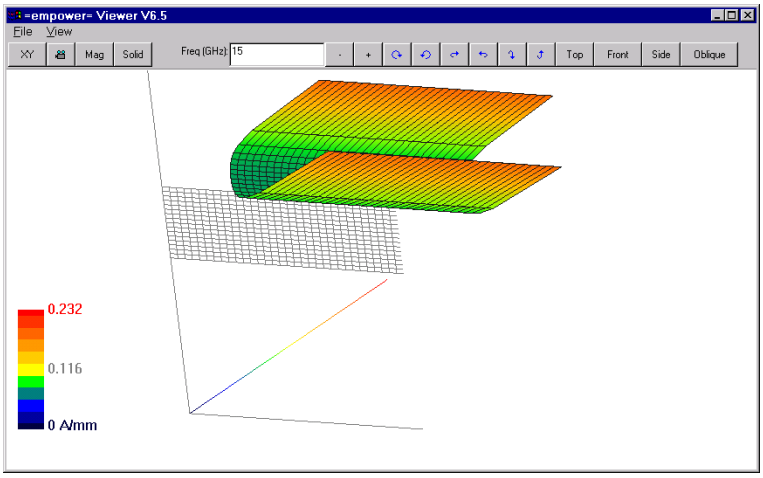
Note: For printing, Toggle Background Color from the File menu was also used to change the background to white. To reset the time to zero, the animation was turned off, and the Real/Mag/Angle button was clicked three times, returning the mode to real but resetting the time.

The resulting picture in the main viewer window is a 3D plot of the surface current density shown with the grid generated to solve the problem. The axes in the metal plane (grid plane) correspond to the X and Y axes in the box. The origin of the coordinates X and Y correspond to the geometrical origin of the box, (0,0) in LAYOUT. The z-axis perpendicular to the metal plane corresponds to the plotted current/voltage values. The red color on the axis is for high values and dark blue is for zero. The color coded scale makes it possible to evaluate actual values of current density. The plotted values are an additive function of interpolated X- and Y-components of the current density. The

current components are calculated along the cell sides, not at the corners of the cells. The X and Y current components are interpolated to the grid corners and are then added.

The X-Y current display provides general insight into circuit behavior. Again consider the view given above. The dominant eigenwave of the stripline is excited at the left input of the structure. Observe the typical current distribution in the cross section X=0 (click the side view button for a better look at this). At this time the current declines to almost zero at the right output (click the Front view button). This confirms a line length of 90-degrees. Next, animate the response by clicking on the Animate button again. Notice how the dominant stripline wave propagates in the structure. The animation is a simple set of snapshots for the subsequent time moments. The time will vary between zero and the period of the incident wave (1/f seconds).

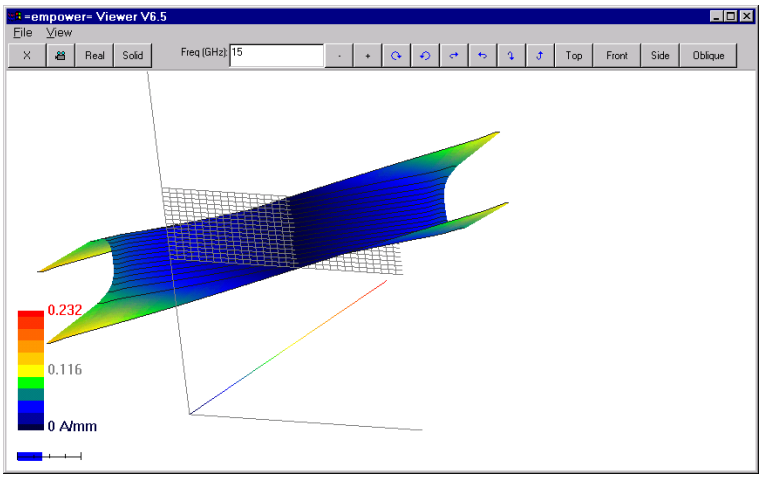
The previous example illustrates the propagation of the wave. For simple evaluation of the high and low current density region the time average values of the current density is more practical. To obtain this plot switch to Magnitude mode by clicking on the Real button. The viewer in this mode is shown below. The results are as expected for a transmission line segment. The current density is highest at the edges and lowest in the middle. Note that the absolute values of the current density at the edges are greatly affected by the grid cell size used. A smaller grid cell size increases the edge current density. However, integrated values of current density are nearly invariable, as they should be [Mexiner, 1972]. If the exact current density values are required we recommend choosing a grid cell size equal to the metalization thickness.



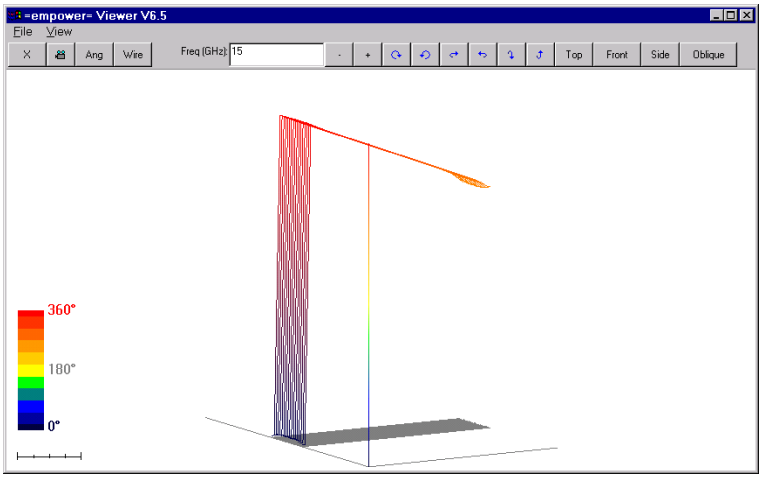
To investigate the various current components you may switch from the XY mode to the X mode (XY/X/Y/Z button). You see only a small change in the graph because the current flows primarily along the line segment as expected. Note however, the component visualization modes (X, Y or Z) are more accurate because the values displayed correspond directly to the values calculated by EMPOWER; no interpolation is necessary for these modes.

The absolute value of the current density is currently displayed. Switch to the Real mode using the menu: **View Menu/Switches/Absolute Value Display** and select **Real** mode.

Animation should be turned on also (Animation camera button). A snapshot of the plot is shown below. The Real mode displays both current density values and direction. Current flows in the positive X direction if the displayed values are above the metal layer (the color coded axis direction). The current flows in the opposite direction if the displayed values are below the metal plane.



To obtain even additional insight the phase of the signal along the line may be displayed. Stay in X component mode, turn off animation, and switch to the Angle mode by clicking the Display Option button until it reads Ang. You may view the wireframe mode by clicking the Wire/Solid button until it reads Wire. At the initial time t=0 and with a matching rotation you will a display similar to the one below. It displays delay of the current densities along the structure in terms of a complex vector rotation angle. 360 degrees of phase corresponds to a one wavelength delay period. The difference of the current phases at the input and output again confirms a 90 degrees line segment.

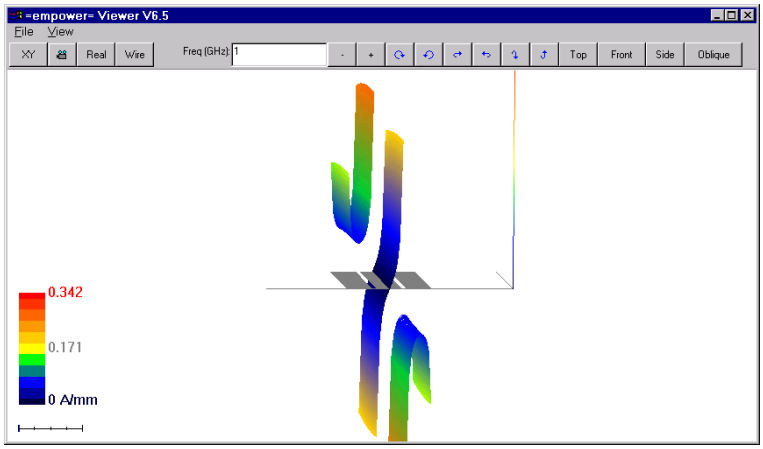


The line segment example was prepared at two frequency points. All graphs and explanations given here used the first frequency point 15 GHz. The second point is 30

GHz and the corresponding segment length is a half of the wavelength. You may display results at 30 GHz by clicking the + button and then choosing the views of your choice.

MultiMode Viewer Data

This example illustrates the eigenwave multi-mode excitation capabilities of EMPOWER. A three conductor coupled microstrip line segment from [Farr, Chan, Mittra, 1986] is described in the schematic file LNMIT3.WSP. Three microstrips are 1 mm wide and 0.2 mm apart. They are on a 1 mm substrate with relative permitivity of 10. The segment is 8 mm long. The structure has three modally coupled inputs at opposite segment sides. We expect at least three propagating modes. Load the example in GENESYS. The listing file (Right-Click on the EMPOWER simulation in the Workspace Window and select "Show Listing File") gives information about the propagating waves. The first eigenmode is an even mode with integral current distribution pattern +++, the second eigenmode is odd (pattern +0-), and the third one is again even (pattern +-+). To excite the odd mode as an example, select **Generate viewer data** and enter 2 in the "Mode Number to Excite" box of the EMPOWER properties dialog. Run the viewer. A snapshot of the calculated current density function is shown here. All settings except two are the same as in the previous example. The initial view was set to the side view (View Menu/Side or Side button), and the polygon view was set to wireframe (View Menu/Switches/Wireframe or Solid/Wire button).

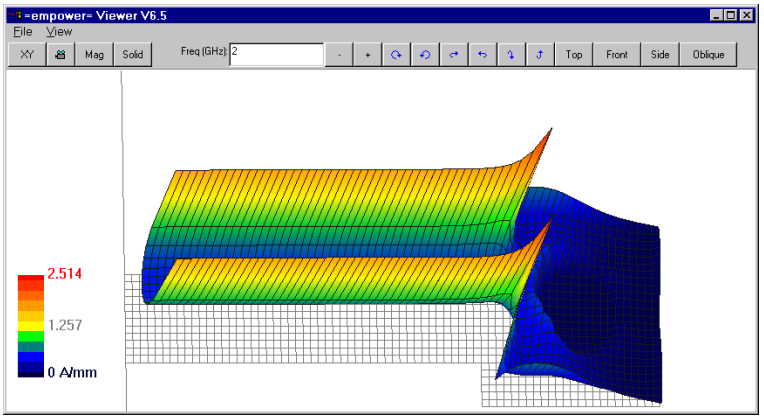


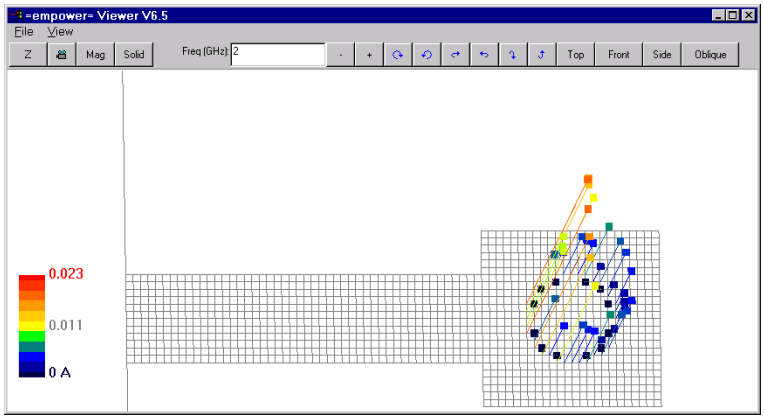
The plot confirms that this is an odd mode and shows the typical current density distribution. If currents on the left strip flows in the forward direction, the currents on the right strip flow in the backward direction and the center strip currents flow in opposite directions at the opposite strip sides. For a dynamic view, turn on the animation and rotate the plot for a better view of the propagating wave.

To calculate the viewer data for the other eigenwaves run EMPOWER and the viewer twice more with "Mode Number to excite set to 1 and 3. Note that newly calculated data will overwrite the previous ones. To avoid this and to keep viewer data for all excitation experiments you need to save a copy of the existing workspace (LNMIT3.WSP in this case) before the next run.

Via Hole Viewer Example

The last visualization example shows a structure with non-zero X, Y, and Z current components. A segment of microstrip line terminated by a via hole from [Swanson, 1992] is described in the file VIA.WSP. The line is 12 mil wide and is terminated by a metal square 24 by 24 mil with a 13 mil diameter circular via hole in the center. The substrate height is 15 mil and the relative permitivity is 9.8. The box size is 120 by 120 mil. Load this example in GENESYS and run the EMPOWER viewer. The first figure below shows the time averaged plot (View Menu/Switches/Value Mode or Value Mode button) for additive (XY) current density distribution. The view point is the oblique view with a few minor adjustments. The plot shows how the dominant microstrip line mode currents spread across the square metal pad. You can see the typical peaks in the current density function in the vicinity of the metal internal corners where the surface current changes flowing directions. Toggling to the X and Y components of the current (XY/X/Y/Z button), you can investigate how the surface currents change direction in different parts of the structure. Switching to the Z-current visualization mode will show a plot like the second figure below. Note that the scale for the Z-directed currents is in Amperes and not current density. Each current represents a volume current density integrated across the grid cell. They are shown as lines connecting the corresponding geometrical point in the grid plane and the point corresponding to the actual current values. If a via hole surface shape is known, using the current in Amperes it is possible to estimate a current density on the via hole surface. It is obvious from the picture that the current density is higher on the via hole side that is closer to the microstrip line segment.





Viewer Theory

The EMPOWER viewer is a program designed to read, to process and to visualize the current distribution data created by EMPOWER. To obtain a current distribution inside a structure the excitation condition must be defined. This mirrors a real measurement where there are incident and reflected waves. The viewer depicts the case with one incident wave at one input. The excitation conditions are passed to EMPOWER in the command line when running EMPOWER text files. When EMPOWER is launched from GENESYS the excitation conditions are automatically defined from the EMPOWER Setup dialog box when the Generate Viewer Data check box is active. If Generate Viewer Data is selected, the default incident wave is the first eigenwave of the first input. The input number can be changed in the Port number to excite box of the EMPOWER Setup dialog, and the input mode number can be changed in the Port mode to excite box. The control information about what input and what mode are actually excited in the structure is printed out in the listing file (see "PPLT: Input __ mode ___ will be incident." in the listing file).

An output binary file with the extension .EMV is created by EMPOWER to pass data to the viewer program. (In a GENESYS Workspace, the internal name of this file is EMPOWER.EMV) An optional self-documented ASCII data file with extension .PLX can also be written for import into other programs.

To understand the viewer, a review of EMPOWER input and mode representations is helpful. A circuit can have external and/or internal inputs. External inputs are transformed to eigenmode space, de-embedded, and normalized to characteristic impedances of eigenmodes. They could be one-mode or multimode (modally coupled) and the incident wave for these inputs can be one of the input eigenmodes. The incident wave is a harmonic function of time. Its magnitude is unity, and it corresponds both to one Watt instantaneous power and 1/2 Watt time averaged power. The initial phase of the incident wave is zero. Other eigenmodes of the structure are terminated by their characteristic impedances and are perfectly matched. It numerically represents a row of the generalized scattering matrix.

The internal ports are often locations where lumped elements will be included by GENESYS.Parameters of the lumped elements are not required for the EMPOWER

simulation. Thus internal ports default to 1 ohm normalization. In this case, the viewer data may not be as useful, since the lumped elements are not taken into account by the viewer.

It is also possible to use an internal port as a source of energy to excite a structure. The termination impedance can be specified using the option -NI<n>. In this case, the internal inputs are terminated by virtual transmission lines with the specified characteristic impedance. The unit incident wave is excited at the specified input. Note that if option -NI<n> is used then the external inputs are also terminated by transmission lines (or loads) with this impedance after de-embedding and transformation into the mode space if necessary.

If the excitation conditions are defined, EMPOWER calculates the scattering matrix S with default or defined normalization first. Then it creates an excitation vector A=[0...,1,....,0] that contains only one unit element corresponding to the specified input. The other elements of the vector are zeros. Reflected waves vector B are calculated from the equation:

$$B = S * A$$

Then the simulator defines normalized voltages and currents in mode space, denormalizes them, and restores the grid currents and voltages inside regions corresponding to all input surface current regions. Finally, using the input region variables, the program calculates non-zero grid currents Ig for strip-like structures or voltages Vg for slot-like structures. The grid currents and voltages are locally defined model currents and voltages (see the Theory section), and their units are Amperes and Volts accordingly. The grid currents and voltages together with their coordinates on the grid are stored in the .EMV file. (The same data can be written in the self-documented text file with the extension .PLX) The viewer reads the .EMV file and to displays data.

Note that the initial current (voltage) distribution is a model representation and is treated using complex number conventions. The currents (voltages) are complex quantities and harmonic functions of time. So, their magnitudes are maximal values for the excited wave period. The real component corresponds to instantaneous values of currents, and their phases reflect the phase delays of currents at the initial time t=0. Using these initial data the current distribution is calculated versus time. If f is the incident wave frequency, the current distribution Ig(t) at time t is given by expression:

$$Ig(t) = Ig(0) \exp(j*2*pi*f*t)$$

The same formula is valid for the voltage distributions. Advancing time displays snapshots of the current or voltage distribution thus animating the display.

As we mentioned above, the viewer reads the grid currents (or voltages) with their coordinates and prepares them for plotting. The preparation stage includes a transformation of the grid variables to more general current density functions (surface electric current density function for strip-like problems or surface magnetic current density function for slot-like problems). The units for the electric current density magnitudes are Amperes per millimeter (A/mm). The units for the magnetic current densities are Volts per millimeters (V/mm). We choose millimeters to scale graphs to more readable values. The current density functions are created only for the currents in

the signal or metal layer. Viaholes and z-directed ports are always represented as z-directed currents in Amperes.

Summary

To summarize viewer behavior:

- If **Generate Viewer Data** is selected, the default incident wave is the first eigenwave of the first input.
- Define the input number and mode number in the EMPOWER properties dialog
- An incident wave is a time harmonic function with unit magnitude and zero initial phase.
- The external ports are terminated by corresponding mode characteristic impedances while the internal ports are terminated by 1 Ohm if another termination is not defined by the option -NI<n>.
- The instantaneous power of the incident wave is 1 Watt and time average power is 1/2 Watt.
- Surface current density functions are used for the signal or metal layer and integral currents are used for viaholes and z-directed inputs.

Chapter 25 EMPOWER: Box Modes

Overview

A fully enclosed rectangular box acts as a cavity resonator. At frequencies near each resonance mode significant coupling exists between the desired signal metalization and the cavity. Because this coupling is reciprocal coupling occurs between segments of the signal metalization. This is nearly certain to perturb the circuit responses as the operating frequency approaches or exceeds the first resonant frequency of the cavity. While EMPOWER inherently predicts these effects, they may have a significant destructive effect on the performance of your designs. Box modes are clearly illustrated in this example.

Homogeneous Rectangular Cavity

In the formulation which follows we use definitions from the section on Geometry. The height of the box in the z direction is h, the length of the box in the x direction is a and the width of the box in the y direction is b.

The resonant wave number for a rectangular cavity is

$$k_{mnp} = \left[\left(\frac{m\pi}{a} \right)^2 + \left(\frac{n\pi}{h} \right)^2 + \left(\frac{p\pi}{b} \right)^2 \right]^{1/2}$$

(MKS units) and the resonant frequency when homogeneously filled with material with a relative dielectric constant of e_r is

$$f_{mnp} = \frac{k_{mnp} c}{2\pi \sqrt{\varepsilon_r}}$$

where c is the velocity of light in a vacuum, 2.997925x10^{8}m/sec. The frequency of the dominant mode is f_{101} (lowest resonant frequency) and in a vacuum we have

$$f_{101} = \frac{c}{2} \sqrt{\frac{1}{a^2} + \frac{1}{b^2}}$$

In air, with linear dimensions in inches and the frequency in megahertz

$$f_{101} = 5900\, MHz \bullet inches \sqrt{\frac{1}{a^2} + \frac{1}{b^2}}$$

With linear dimensions in millimeters and the frequency in gigahertz

$$f_{101} = 149.8GHz \bullet mm\sqrt{\frac{1}{a^2} + \frac{1}{b^2}}$$

For example, in air (er=1.0006) with a 2´4 inch, 0.5 inch high box, b=101.6mm, a=50.8mm and h=12.7mm. Then k_{101}=69.14 and f_{101}=3297 MHz.

Higher Order Box Modes

It is interesting to note that if *h*<*a* and *h*<*b* then the frequency of the dominant mode is not a function of the cavity height. This is not the case for certain higher order modes. The mode which is next higher in frequency than the dominant mode is a function of the relative values of *h*, *a* and *b*.

Consider for example the previous 2x4x0.5 inch box (or, any size box with the size ratios *b*=2*a* and *h*=I/4). Therefore the wave numbers are

$$k_{nmp} = \frac{\pi}{a}\sqrt{\left(m^2 + 16n^2 + \frac{p^2}{4}\right)}$$

The wave numbers for the lowest frequency modes for this shape box and the resonant frequencies with *a*=2 inches are listed here:

Mode	Wave #	Freq (MHz) @ a=2 inches
101	1.118pi/a	3299
102	1.414pi/a	4173
103	1.803pi/a	5319
201	2.062pi/a	6083
104	2.236pi/a	6598
105	2.693pi/a	7945
301	3.041pi/a	8974
106	3.162pi/a	9331

Notice that higher order modes occur frequently after dominant mode resonance. It is possible to minimize perturbations in narrowband applications by operating between resonant frequencies. However, the above analysis assumes a pure homogeneous

rectangular cavity and dielectric. Partial dielectric loading and signal metal within the cavity will influence the frequency. A more conservative and safer approach is to enclose the circuit in a box with the dominant resonant mode higher than the highest frequency of interest.

Partial Dielectric Loading

If the cavity is not homogeneous, but instead is partially filled with a dielectric and the remainder of the cavity is filled with air then the dominant mode resonant frequency is reduced and may be approximated using a filling factor [Johnson, 1987]. Assuming the substrate is mounted on the floor of the cavity, the resonant frequency of a partially filled rectangular cavity, $f_{partial}$, is

$$f_{partial} = f_{101}\sqrt{1-\frac{\varepsilon_r - 1}{\varepsilon_r}\left(\frac{t}{h}\right)}$$

where t is the thickness of the substrate and h is the height of the cavity without a substrate. For example, f_{101} for the 2x4 inch box is reduced from 3299MHz to 3133MHz with t=62mils and e_r=4.8.

This expression is approximate because the electric field lines are not parallel to the z axis and a component of these lines terminate on the side walls. This mode is referred to as a quasi-TEM_{101} mode.

Signal Metal Effects

Relatively sparse signal metal has little effect on the resonant frequency. Larger metal segments, particularly when grounded, significantly reduce the resonant frequency. To obtain a feel for the significance of signal metal you may add extraneous metal to the substrate in Example 10, Box Modes, and observe the shift in the transmission peaks.

Top Cover

Transmission line discontinuities disturb current flow and energy is lost from the transmission structure. While this lost energy is typically small, the Q of the resonant cavity is high and coupling at these frequencies is significant. Removing the cover of the enclosure causes energy to be lost to free space and resonance effects are reduced. This greatly reduces coupling between metal segments of the circuit and it is evident in the responses given in the Box Mode example cited earlier with the cover removed. Effects of removing a top cover are illustrated in the Examples "EdgeCoupledOpen.WSP" and "Box Modes.WSP". See your Examples manual for details.

Cavity Absorber

A similar benefit may be derived by placing absorber material on the cover or in the cavity. While the poor ultimate rejection in the stopbands of filters is not recovered, heavy

coupling between segments is avoided. This is sometimes necessary to eliminate oscillations of high gain amplifiers in oversize enclosures.

By far the most elegant and safest approach to minimizing box mode problems is placing circuits in small enclosures.

Chapter 26 EMPOWER: Theory

Overview

This section gives a technical description of the basic EMPOWER algorithms. Unlike most similar tools on the market, EMPOWER is based on the method of lines (MoL) and comprises a set of numerical techniques designed to speed up calculations while increasing accuracy of computations. Incorporation of geometrical symmetries (including rotational), reduction of problem complexity using thinning out and linear re-expansion procedures, and multimode deembedding by the simultaneous diagonalization method are outlined here. This theory section is for EMPOWER users familiar with numerical electromagnetics foundations. We have added this material because MoL is less well known than the method of moments or the finite difference method.

MoL can be represented as a simple combination of both method of moments and finite difference method. Thus we have skipped common parts and given our attention to the original parts of the algorithm. More details on particular algorithm parts, accuracy and convergence investigation results can be found in publications listed in the References section in the EMPOWER Engine Theory and Algorithms section.

Basically, the theory behind the simulator can be reduced to the following: An initial 3D problem in a layered medium is reduced to a 2D problem through a partial discretisation of the Maxwell's equations and its solution for a homogeneous layer in a grid spectral domain. The resultant matrix relating local grid currents and voltages is reduced to an immitance matrix relating integral currents and voltages in ports. To extract a generalized scattering matrix of the problem from the immitance matrix, the method of simultaneous diagonalizations is used.

After this introduction we are ready to formulate the reasons for using MoL as a basis for an electromagnetic simulator. The 3D problem is discretized only in two directions and reduced to a 2D one that corresponds naturally to the planar MIC structures. In contrast with the method of moments, the MoL gives a self-regularized solution with only one variable (grid cell size) defining all parameters of the numerical model. That eventually leads to monotonic convergence of calculated data and predictable errors of calculations. The high grade of internal symmetries of the MoL based algorithms makes it possible to substantially reduce the numerical complexity of the main matrix computation stage. The main restriction of using a regular grid related with its potentially excessive number of variables has been overcome by introducing thinning out and re-expansion procedures. Basically, the discrete analogue of a problem is processed in a way similar to the method of moments but in discrete space like the finite difference approach which facilitates different aspects of the solution and programming.

Thus, the main advantages of the MoL are reliable solution with the predictable calculation error, relatively straightforward algorithms that facilitate development of general purpose programs, and a lot of possibilities to speed up calculations and to increase accuracy of solutions. For these reasons and others, we decided to use it for the

electromagnetic simulator. This section summarizes the theoretical backgrounds with emphasis on the problem formulation and acceleration techniques.

Historical Background

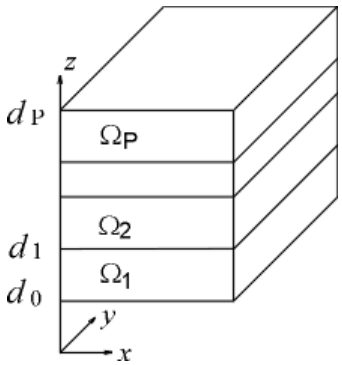
Most commercial electromagnetic (EM) simulators designed for MIC and MMIC work are based on integral equations and the method of moments (MoM). EMPOWER is based on the method of lines (MoL). This technique has excellent error convergence properties and submits well to code optimization to minimize numeric complexity.

The root of EMPOWER is work which began in 1987 at the Novosibirsk Electrical Engineering Institute. This lead to the commercial development of TAMIC in 1991 in Moscow. TAMIC saw commercial use in the Soviet Union and elsewhere. In late 1996, Eagleware acquired TAMIC and the principle contributor joined Eagleware to begin significant improvements. The code was integrated into the GENESYS environment at release Version 6.5 in 1998.

Problem Formulation

This section describes a general mathematical formulation of the boundary value problem to be solved. It defines all restrictions in the problem domain. You can use this section to decide whether your particular problem fits the formulation or not.

For analysis, a passive MIC structure is confined inside a three dimensional rectangular volume bounded by electric or magnetic walls. The volume is filled by a layered medium that may consist of an arbitrary number of isotropic homogeneous dielectric or magnetic layers as shown below.



The electric (**E**) and magnetic (**H**) field vectors are related by Maxwell's system of equations:

$$\left.\begin{aligned} rot\,\mathrm{H} &= i\omega\varepsilon_p \mathrm{E} + \mathrm{Jz} \\ rot\,\mathrm{E} &= -i\omega\mu_p \mathrm{H} \\ div\,\mathrm{E} &= 0 \\ div\,\mathrm{H} &= 0 \end{aligned}\right\} (x,y,z) \in \Omega_p \qquad \text{(A-1)}$$

Here **Jz** is the volume density vector of z-directed currents inside a media layer. e_p and mu_p are permittivity and permeability of the media layer. e_p is a complex value for a lossy media. The z-directed currents are constant values inside a layer, but they can change from layer to layer, which gives a possibility to discretize the problem along the z-axis. Thus we have all six components of the electric and magnetic fields inside a layer with constant current across it. X and y current components can exist only in a signal layer z=dj, parallel to medium layer interfaces. Generalized boundary conditions for the signal layer are:

$$\begin{aligned} 1_z * \{\mathrm{H}(+dj) - \mathrm{H}(-dj)\} &= \eta \\ 1_z * \{\mathrm{E}(+dj) - \mathrm{E}(-dj)\} &= 0 \end{aligned} \qquad \text{(A-2)}$$

The signal layer plane can contain arbitrarily shaped regions of perfect metalization, regions with complex surface impedances (lossy metal), resistive films, and regions modeling lumped element connections. All regions have zero thickness. The top and bottom walls of the box can be ideal electric & magnetic walls or walls with surface impedance. The structure can also be terminated by semi-infinite rectangular waveguides in the planes of the box top and bottom walls. A clarification of the boundary conditions for the media layer interfaces (A-2) are given in the following table.

1.	Region without metalization	$\eta = 0$
2.	Lossless metalization	$1_z * \mathrm{E}(d_j) = 0$
3.	Surface Impedance	$\eta = \sigma\,\mathrm{E}$
4.	Port Region along X-Axis or Internal Port (Lumped Element Region) along X-Axis (the same for y-axis). C is region cross-section, l is region length.	$\int_C \eta\, dy = Y_i \int_l \mathrm{E}_x\, dx$

5.	Internal Port along Z-axis	$\iint_C j_z \, dxdy = Y_l \int_l E_z \, dz$

Input ports in the structure are modeled by line segments approaching the outer boundaries (line conductors) and surface current sources in the regions where line conductors approach the walls of the volume. It is assumed that the currents inside the input and the lumped element regions are constant in the direction of current flow and the corresponding electric field component along the region is constant across it. Thus, the integral of current across the region gives an integral current, and integral of the electric field along the region gives an integral voltage for the region.

The desired solution of the electromagnetic problem is an immitance matrix relating the integral voltages and currents in the port and lumped element regions. This is actually a kind of Green's function contraction on the port and lumped element regions. After connection of the lumped elements the immitance matrix can be transformed into a generalized Y- or S-matrix using the simultaneous diagonalization method (see the de-embedding section).

Thus, we have a problem formulation that is appropriate for a wide range of microwave and mm-wave devices such as planar filters, dividers/combiners, matching circuits, phase-shifters, attenuators, diplexers, amplifiers as well as their components.

Method of Lines

The method of partial discretization (later called the method of lines [MoL]) is as old as partial differential equations and the finite difference approach to their solution. Traces of it can be found in the 18th century works of J.-L. Lagrange. Its first conscious usage for the numerical solution of elliptical problems could be attributed to M.G. Slobodianskii [1939]. An almost complete reference on the MoL development and applications in the period from the beginning up to sixties are given in Liskovets' paper [1965].

The network analogue method of B.L. Lennartson [1972] is probably the first technical application of the MoL to the static numerical analysis of planar multiconductor lines. It was not quite straightforward when it was published, and the actual exploration of the method for microwave integrated circuit structures began in the early eighties in works of German scientists H. Diestel, R. Pregla, U. Schulz, S.B. Worm and others [Pregla, Pascher, 1989].

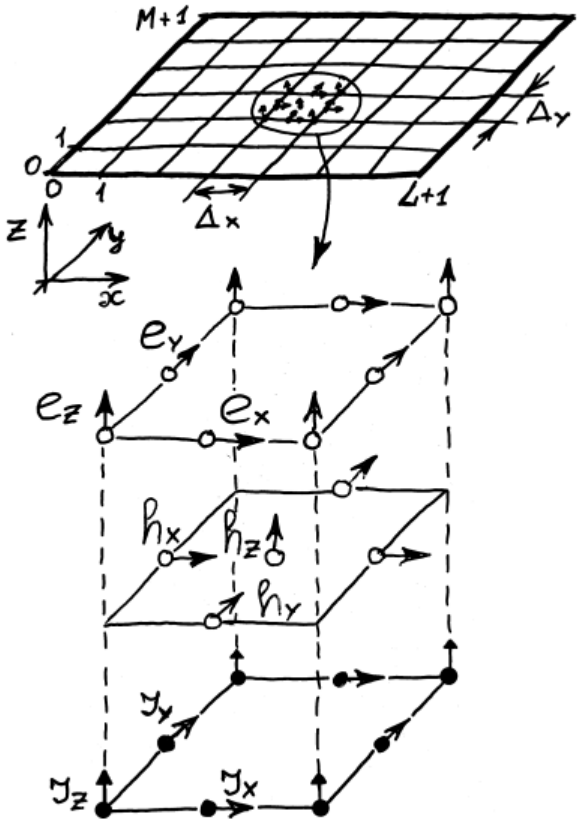
The EMPOWER algorithms can be also classified as MoL because of its semi discrete nature. Originally the network impedance analogue method [Kron, 1944/ Sestroretzkiy, 1977] and a grid spectral representation inside homogeneous layers were used to analyze the layered three dimensional structures [Sestrorezkiy, Kustov, Shlepnev, 1988] that correspond to a combination of the 3D finite difference approach and the spectral domain technique. Later, only the discretisation in the metal plane was left, but the method still retains some advantages of the network impedance analogue method. That is why we sometimes refer to the EMPOWER numerical techniques as the impedance interpreted method of lines.

Here are the main solution stages of the impedance interpreted MoL:

- Partial discretisation of the Maxwell's equations, only in the plane of metalization (x-y plane).
- Grid spectral representation of the EM fields in the homogeneous layers.
- Building Grid Green's Function (GGF) matrix in spectral domain using impedance form of the solution in a layer.
- Representation of each GGF matrix element as a sum of four elements of an auxiliary array obtained using DFFT technique.
- Equidistant grid transformation to a non-equidistant grid using thinning out and linear re-expansion procedures.
- Automatic detection of symmetry for symmetrical and nearly symmetrical problems (reflection and 180° rotational).
- Solution of the main system of linear algebraic equations using partial inversion.
- Resolution to Y- or Z-matrix relating integral grid currents and voltages in the input and lumped element regions.

Mapping on the Grid

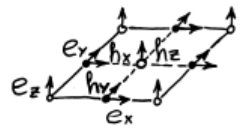
To map a boundary value problem for a partial differential equation on the grid basically means to substitute the problem with solution defined in a space of continuous functions by a problem with a solution defined in a discrete space. The model solution must be as close to the continuous one as possible. To solve the problem we approximated the partial derivatives in the signal plane by finite differences applied to grid analogues of the field components. The corresponding grid is shown here.



There are L+1 equidistant cells along the x-axis and M+1 cells along the y-axis. The grid equivalents of the electric (e) and magnetic (h) fields are defined as corresponding continuous function values in offset grid points as is shown for a grid cell above. The grid functions are continuous along the z-axis. Grid x- and y-directed current variables (Jx,Jy) are defined as integrals of the surface current in the metal plane across the grid cell. Grid z-directed currents (Jz) are defined as surface integrals of the volume current density jz across the grid cell.

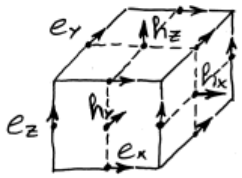
The first offset model of Maxwell's equations was apparently proposed by G. Kron [1944]. The cells below show a summary of the similar models implemented by different authors. The resultant system of differential-difference equations approximates the initial system with the second order locally inside a layer. The initial boundary value problem can contain infinitesimally thin metal regions with consequent singularities of the field and conductivity currents at the metal edges [Meixner, 1972]. That is why a global approximation order of the problem is usually lower and the largest calculation error part for integral parameters of a structure (Y, S-matrix elements, characteristic impedance) decreases usually proportionally to the grid cell size. That is, the monotonic convergence was observed for almost all problems solved on the initial equidistant grid. This makes it possible to use such powerful convergence acceleration techniques as Richardson's extrapolation [Richardson, 1927; Marchuk, Shaidurov, 1979]. Note that this is an

observation and it cannot be proven to work for all problems. The technique used here for the descriptor matrix evaluation using current sources in the metal plane is empirical. The evaluation accuracy depends on parasitic high order modes that could be excited by current sources and if they are close to their cutoffs or even are propagating, the estimated descriptor matrix could be far away from the correct one. This can be expected, however, since real circuits which have unexpected high order modes near the cutoff usually do not work properly either.

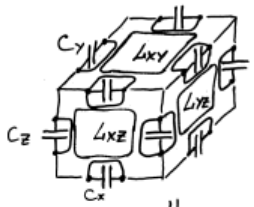


=empower=

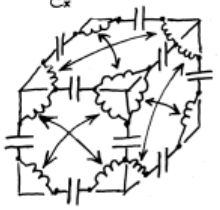
S. B. Worm,
R. Pzegla, 1984



K. S. Yee, 1966
Also T. Weiland, 1977



B. Sestroretzkiy, 1977



G. Kron, 1944

Grid Green's Function

The Grid Green's Function (GGF) has been mentioned quite a few times. The GGF is a solution of the differential-difference analogue of Maxwell's equations (A-1) excited by a unit grid current (Jx,Jy or Jz). The solution or response function is a discrete function in the xy plane and continuous inside layer along the z-axis. Actually to solve the formulated problem we need just a contraction of the GGF to the signal plane and to the regions with non-zero z-directed currents. This contraction is a matrix due to the discretization.

To find the GGF matrix we used a spectral approach similar to one used in the spectral domain technique or in the method of moments [Nikol'skii, 1982; Vesnin, 1985; Jansen,1985; Rautio, Harrington,1987; Dunleavy, Katehi,1988]. Instead of continuous TE

and TM rectangular waveguide eigenwaves [Samarskii, Tikhonov, 1948], their grid analogues are used as a basis to expand the electromagnetic field inside a layer. The number of the grid TE and TM waves is finite and their system is complete. This means that instead of a summation of series as in the spectral domain approach we have finite sums. Moreover each basis grid eigenwave has a grid correction that provides convergence of sums to the series obtained by the continuous spectral domain approach. Note that a backward process is impossible and a simple truncation of the series does not give the same answer as the grid technique. The finite sums and the grid corrections are the most important things for monotonic convergence of the algorithm.

To construct the GGF matrix in the grid spectral domain, the impedance form of the solution for a layer was used. The base of the solution is a layer admittance matrix in the grid spectral domain. This matrix relates the grid analogues of the tangential electric and magnetic field components at opposite surfaces of the layer, z-directed currents and integrals of z-directed grid electric field along the z-directed current inside the layer. All of these are in the basis of the grid eigenwaves, thus we have a set of independent matrices for each pair of grid eigenwaves. Uniting those matrices for all layers in a structure gives a grid spectral GGF representation. The construction procedure is completely automated for arbitrarily layered configurations. This technique is similar to the impedance approach in the spectral domain [Uwano, Itoh, 1989]. The grid spectral GGF representation was also called a GGF eigenvalue vector, but that term is not quite correct. The dimension of the vector is about 3*L*M if there is only one signal layer.

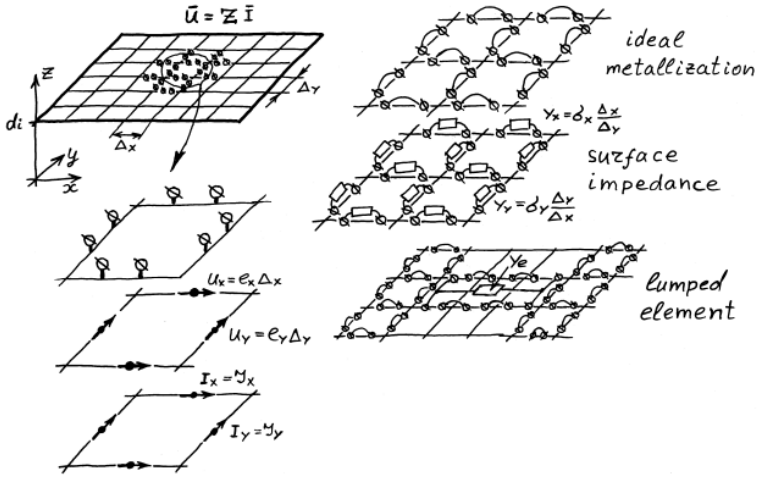
All we need now to get the GGF matrix in the initial space is to perform a backward transformation of the GGF eigenvalue vector from the grid spectral domain to the spatial domain. To do it an auxiliary array called general sums array is introduced. The dimension of the general sums array is also about 3*L*M. Each element of the GGF matrix can be obtained as a sum of four elements of the general sums array. The general sums array depends only on the box and media structure and the grid cell size. Its elements are calculated via the discrete Fourier transforms of the GGF eigenvalue vector using the Prime Factor algorithm. This stage is based on the maximal utilization of internal symmetries of the bounded equidistant grid and usually takes negligibly small CPU time. Moreover it can be done only once for all structures with the same box, media and grid. The described technique is quite similar to the main matrix filling procedure designed for the spectral domain technique [Hill, Tripathi, 1991], except that it has been done here in finite space and we calculate the GGF matrix elements without additional truncation (or series summation) errors. It can also be reformulated in matrix form in accordance with [Pregla, Pascher, 1989]. The GGF matrix can be represented by a sum of Toeplitz and Hankel matrices and their rows can be obtained directly from the general sums arrays.

Informational Multiport

The informational multiport term was introduced by B.V. Sestroretzkiy [1987] and in a nutshell means a model multiport that reflects electromagnetic properties of an object before superimposing an additional boundary condition. It comprises information about all possible structures that could be formed by different combinations of the additional conditions. The boundary condition superimposing can be represented as a set of simple manipulations with the informational multiport terminals. We have added this section to

clarify connections of the numerical electromagnetic solution with the circuit theory. This technique is also known as the impedance interpretation of boundary condition superimposition.

The GGF matrix obtained in the previous section can be represented as an impedance matrix Z of a multiport shown on the left below.

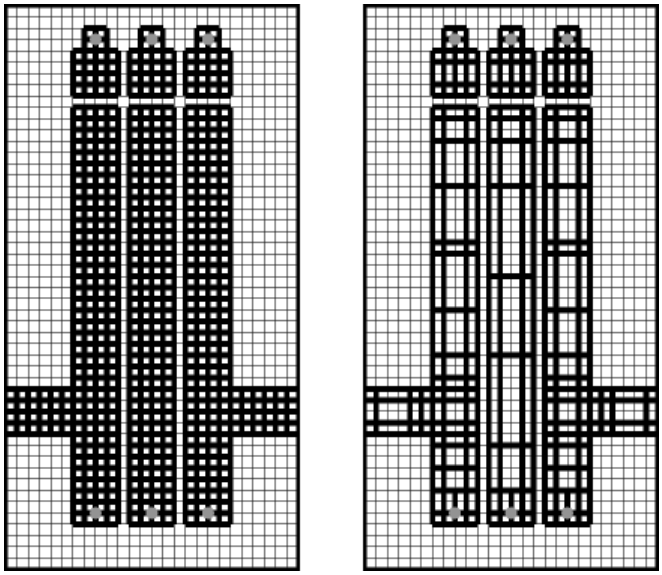


The multiport terminals are conceptual and their positions are just a schematic representation. Four conceptual ports or pairs of terminals correspond to a grid cell as shown in the figure. The total number of ports oriented along the x-axis is M*(L+1). The total number of ports oriented along the y-axis is L*(M+1). The multiport can also have a set of z-directed ports corresponding to via-holes or z-directed internal inputs. Note that we do not need to calculate all elements of the multiport impedance matrix and its order can be reduced taking into account that some ports are no-loaded or short circuited. The no-loaded terminals correspond to regions of the signal layer without any conductivity currents. The right half of the figure illustrates the correlation of other types of the boundary conditions to operations with the informational multiport terminals. Operations with the z-directed terminals are similar.

The operations in a discrete space of the informational multiport terminals are completely in accordance with the usual electromagnetic theory. To connect a lumped element for example , we performed both serial connections of terminals along the element that corresponds to the electric field integration along the element and parallel connections that corresponds to the surface current integration across the element (see the Table above). The analogies described are meant to facilitate understanding of numerical electromagnetics. Note that the examples given are not the only possible manipulations with the terminals with physical electromagnetic equivalents.

Numerical Acceleration Procedures

Before filling the reduced GGF matrix we can additionally decrease the GGF matrix order and required storage space by means of thinning out with linear re-expansion procedures and by incorporating a geometrical symmetry into the problem.



Thinning out is a simple elimination of the grid currents in metalized regions that can be represented by a smaller number of currents without loosing accuracy. As an illustrative example, the left half above shows a three resonator filter mapped on the grid. The grid cells with possible non-zero conductivity currents (metalization regions) are depicted by the thick lines. The thinning out procedure decreased the number of the currents in the problem and leaves the currents that are shown by the thick lines in the right half above. This looks like a pseudo-non-equidistant grid over the regular grid that is finer near edges, corners and via-holes and coarser inside the solid metal regions. The enlarged secondary grid cells after the thinning out consist of non-divergent current borders along each side that can be substituted by two variables on the grid using linear re-expansion. Combination of these two procedures makes it possible to overcome restrictions of the MoL with a regular grid while keeping the main advantages of the equidistant grid.

The described procedure with total elimination of some currents inside the solid metal regions is called the wire model. It basically substitutes a problem with another one with removed small metalization pieces. It certainly gives an additional error, but fortunately this error is opposite to the regular grid model error. In other words, the wire thinning out model actually increases the solution accuracy if the structure is thinned out properly. However, if too much metal is removed, the thinning out error dominates. Thus, a solid thinning out model procedure was introduced to avoid it. The solid model can be represented as a simple modification of the wire model. To explain it we start from the pseudo-non-equidistant grid of currents formed for the filter and shown above. Instead of complete elimination of the currents inside the enlarged grid cells we leave some of them to keep metal surface solid. Those currents left are also replaced with just two variables by means of linear re-expansion. The solid model is more correct but gives a larger number of variables for similarly thinned out problems in comparison with the wire model. (The solid model is actually a way to form a non-equidistant grid with the grid function re-expansion in a discrete space.)

The GGF matrix of a symmetrical problem could be reduced to a centrosymmetrical matrix (with centrosymmetrical blocks in the case of two-plane symmetry) and it is treated in the way similar to described in [Weeks, 1979]. This reduces required CPU memory from 4 to 16 times (serial allocation of partial matrices) and speeds up calculations from 4 to 16 times. One plane, two plane, and 180° rotational symmetries are included in the program.

Thereafter, the classic Gauss' inversion algorithm is used with a few changes. The result of this stage of solution is a matrix (Y- or Z-matrix) relating the grid currents and voltages in the input source regions, and thus we need to get only a small part of the inverted matrix corresponding to these variables. A partial inversion procedure performs it and gives an additional acceleration.

De-Embedding Algorithm

The method of simultaneous diagonalization (MoSD) [Shlepnev, 1990, 1998] is used to extract a multimode or generalized S-matrix. The MoSD is based on the electromagnetic analysis of two line segments corresponding to an MIC structure port to be de-embedded. The segments have different lengths and the same surface current source regions as in the initial structure. The result of the EM analysis is two Y-matrices relating integral grid currents and voltages in the source regions. These matrices, transformed from the space of the grid functions to a space of the line eigenmodes, are set equal to Y-matrices describing independent modes propagated in continuous part of the line segments. It gives the basic non-linear system of equations relating eigenwave propagation constants and characteristic impedances, a matrix of transformation from the grid functions space to the mode's space (transformation matrix) and an auxiliary matrix that helps to match propagated modes perfectly (compensation matrix). Solution of the system is based on simultaneous diagonalization of Y-matrix blocks. Each port of the MIC structure or discontinuity can be de-embedded using the pre-calculated line parameters and the transformation and compensation matrices. The main advantages of this approach are the possibility of multimode deembedding without direct spectral analysis of the line cross-section and ideal matching of line eigenmodes in the analysis of the line segment that increases the accuracy of discontinuity analysis.

Note that despite the theoretical ability to excite and to match any propagating line eigenwave using the surface current sources in the metal plane, it does not always work in the discrete models. Using a limited number of variables in the source regions it is sometimes impossible to separate different modes completely. Moreover, the success of the MoSD application depends on the high order modes that could substantially influence the result. This is the main drawback of the described MoSD application to planar structures.

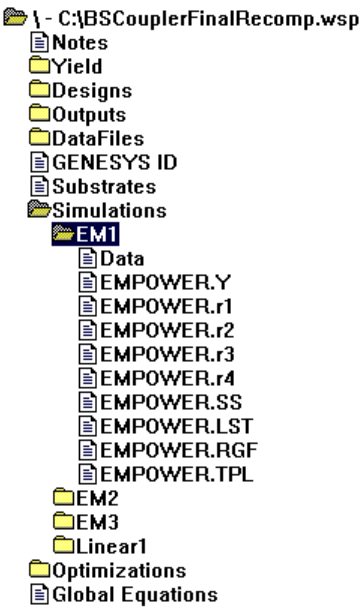
Chapter 27 EMPOWER: File Descriptions

Overview

In performing its tasks, EMPOWER creates many different types of files. An understanding these different files is very helpful in understanding the operation of EMPOWER. These files contain the topology of the circuit, external port line data, generalized S-Parameter normalizing impedances, output information, S-Parameter data, batch commands, Y-Parameter data, viewer data, and backup data.

Where are these files?

Starting with Version 7.0, GENESYS uses OLE Structured Storage for its workspace files. These files are sometimes called "file systems in a file." Structured Storage files contain internal directories and files, and copying one workspace file copies all internal files contained in it. The figure below shows the structure of a typical workspace file. Notice that within each simulation, all filenames begin with "EMPOWER".



Note: Previous versions of GENESYS used actual disk files for all internal EMPOWER files, and separate subdirectories were recommended for each circuit. This is no longer necessary for typical usage.

If you need to access these internal files in a workspace, you have two options:

- Right-click on the EMPOWER simulation on the tree and select "Write Internal Data Files". This automatically creates a directory with the same name as the simulation and places copies of the files there.

- In the same workspace, you can access internal files using a special file prefix, "WSP:", followed by folder names and the filename. For example, to access the EMPOWER.SS, use the name "WSP:Simulations\EM1\EMPOWER.SS".

The second method has the advantage of automatically updating whenever the EMPOWER simulation is re-run. The first method requires you to re-write the data files whenever you need an updated version.

Text Files vs. Binary Files

There are two basic types of data files: text (sometimes called ASCII) and binary. Text files are human readable files. They are universal and can be edited with many different programs such as NOTEPAD or DOS EDIT. Among the text files used by EMPOWER are batch, topology, listing, and S-Parameter files.

Note: Word processors can also edit text files, however, they will store binary formatting information in the file unless explicitly told not to ("Save as...Text"), so we do not recommend their use for editing text files.

In contrast, binary files are not human readable. They contain information encoded into the numbers which make up the file which are ultimately turned into ones and zeros, thus the name "binary." Unlike text files, binary files are not universal and should only be edited by a program designed for the particular type of binary file you are using. Editing a binary file in a regular word processor or text editor will undoubtedly destroy it! Some binary files used by EMPOWER and GENESYS are workspace, line, and Y-Parameter files.

File Extensions

You can normally tell the kind of file you have by looking at its extension (the part of the name after the last period). Some commonly used extensions include EXE (executable), TXT (text), and HLP (help). Each kind of file used by EMPOWER has its own unique extension. These extensions are shown here. Each of these types will be discussed individually in the following sections.

Note: Unfortunately, Windows can be setup to hide files extensions, as well as actual files, from the user. We would recommend that you turn off this "feature": Double click on "My Computer," Select Options from the View menu, Click the Viewer tab, Click "Show all files," Deselect "Hide MS-DOS file extensions for file types that are registered", and click OK. Different versions of Windows may have slightly different procedures.

Extension	Type	Purpose
EMV	Binary	Viewer data
L1, L2, etc.	Binary	Port deembedding and line data for port 1, 2, etc.
LST	Text	Listing file summarizing all EMPOWER data
PLX	Text	Text listing of viewer currents.
R1, R2, etc.	Text	Port normalizing impedances
RGF	Binary	Port deembedding and line data for a port with a user-specified deembedding file name.
RX	Text	Frequency vs. impedance data
WSP	Binary	GENESYS Workspace File
SS	Text	S-Parameter results
TPL	Text	"Netlist" for EMPOWER
Y	Binary	Y-Parameter results
~SS, ~RG, etc.	Backup	All files with either a name or an extension starting with tilde (~) are backup files and can be safely deleted.

.EMV (EMPOWER Viewer) Files

Written by: EMPOWER
Type: Binary
Can be safely edited: No
Average size: 10 to 100Kbytes, but may be larger
Use: Data for viewing currents or voltages

EMV files (EMPOWER Viewer) files are completely self-contained files containing all information needed by the viewer to display currents and voltages for a circuit. These files contain information about the box and the grid mapping of the circuit as well as actual complex current or voltage values at each frequency. EMPOWER creates an .EMV file whenever "Generate Viewer Data" is checked or the -In option is specified.

.EMV files can only be read by the EMPOWER viewer. If you want to generate viewer data for import into other programs, you should generate a .PLX text file. For more information on viewer files, the Viewer section.

.L1, .L2, ...Ln (Line Data) Files

Written by: EMPOWER
Type: Binary

Can be safely edited: No
Average size: 1 to 5Kbytes, but may be larger
Use: Internal file for EMPOWER but can also be used in the SMTLP and MMTLP models in GENESYS

EMPOWER must perform a separate line analysis for all external ports. If no filename is specified by the user, then the results from the line analysis are stored in .Ln files. These files also store all information about the box and port and are intelligent: They are only recalculated if necessary, and even then only at frequencies necessary. Even if the circuit changes they are only recalculated if the change affects the line analysis. Notes: When these files are numbered, modally related groups of ports are counted as one. Also, if two ports are identical, then only the first one will create a .Ln file.

.LST (Listing) Files

Written by: EMPOWER
Type: Text
Can be safely edited: Yes*
Average size: 50K to 200K, but may be larger
Use: Gives all calculated data and grid mapping from EMPOWER in human readable form

This file is overwritten whenever EMPOWER is run. It should be carefully checked whenever a new circuit is analyzed, especially if that circuit was described manually from a text TPL file. The following sections describe the contents of a listing file. Note: Some of the information described below is only output if "Output additional info in listing file" is checked or -La is specified.

QCHK SECTION

This section allows you check the quality of the solution. Entries include:

Min. media wavelength to mesh size ratios - should be at least 20.

Thinning out thresholds - Specifies the maximum number of lines in a row which can be thinned out.

Max box size to media wavelength ratios - If the box is too large, you will have box resonances. If this line ends with an exclamation mark (!), it may be too large. See the Box Modes section for more details.

PACKAGE STRUCTURE

This section is only present when the "Extra Details in Listing File" option is used. It gives a summary of the substrate and metal layers used as well as cell sizes.

MEMORY SECTIONS

Several memory sections throughout the listing file give memory requirements for different parts of the simulation.

MAP OF TERMINALS

This section shows the grid representation of the problem.

SDTC SECTION

Symmetry detection sections specify whether the structure is symmetrical. The symmetry processing additionally shows where any differences occurred and can be very useful in finding out where the structure is not symmetrical. The coordinates specified refer to the terminal map shown above.

LINE ANALYSIS MODE RESULTS

This area of the listing contains sections identical to those described above which pertain to the line analysis. Below these sections you will find a table of line parameters for each frequency. The entries are:

Nm - port number

Type - impedance type, real (re) or imaginary (im). Normal lines should have a real impedance.

Zo (ohm) - Line impedance

Gw(rad/m) - propagation constant

Gw/Go - propagation constant relative to free space

Comp Phase, Compensation Admittance - value of phase and impedance compensation for deembedding.

S-MATRIX TABLES

Each table gives the circuit's s-parameters at one frequency. For normal, non-multimode inputs, as an example, S21 is found in the row with input numbers 2 and 1 (in that order).

.PLX (Current/Viewer Data) Files

Written by: EMPOWER
Type: Text
Can be safely edited: Yes*
Average size: 200 Kbytes to 2Mbytes, but may vary
Use: Importing current data from EMPOWER into another application, such as Matlab or Excel.

This file contains two tables per frequency, one each for x- and y-directed currents. Each table contains 4 columns containing the x and y coordinates followed by the real and imaginary part for each current. These tables could be edited, but it would be best to leave them alone since they would be very tedious and error-prone to edit them by hand. These files should be very useful in other applications, as the engineers at Eagleware used third party applications to graph currents before our EMPOWER viewer was completed.

.R1, .R2, ... Rn (Port Impedance) Files

Written by: EMPOWER
Type: Text
Can be safely edited: Yes*

Average size: 1Kbyte
Use: Read by GENESYS when Generalized S-Parameters are requested

These files contain each port's impedance versus frequency. These ports are read by GENESYS if the keyword GEN is used in place of a termination impedance. The files are formatted just like RX files in GENESYS. GENESYS always requests these files when EMPOWER is run from GENESYS.

Notes: These files are numbered differently than Ln files. When these files are numbered, each port in a related group of ports is counted individually.

.RGF (Line Data) Files

Written by: EMPOWER
Type: Binary
Can be safely edited: No
Average size: 1 to 5Kbytes, but may be larger
Use: Internal file for EMPOWER but can also be used in the SMTLP and MMTLP models in GENESYS

These files are used in place of .Ln files if a filename was given on the PORT line in the TPL file. When run from GENESYS, this file type is not available; use the .Ln files instead. Otherwise, they are completely identical to the .Ln files described earlier.

.RX (Frequency vs. Impedance) Files

Written by: User
Type: Text
Can be safely edited: Yes
Average size: 1Kbyte
Use: Specifying electrical losses

These files are used to specify the impedance of conductors in ohms per square. These files are used in the EMPOWER layers setup dialog box or in the TPL file. The files are formatted just like RX files in GENESYS.

.SS (S Parameter) Files

Written by: EMPOWER
Type: Text
Can be safely edited: Yes*
Average size: 5 to 50 Kbytes, but may be larger
Use: Contains S-Parameter data calculated by EMPOWER

This file contains the S-Parameter data written by EMPOWER. It is in the industry standard S2P format and can be loaded into most RF and Microwave simulators. Even though these files can be edited, they will be overwritten whenever EMPOWER is rerun.

.TPL (Topology) Files

Written by: User or GENESYS
Type: Text
Can be safely edited: Yes*
Average size: 1 to 5Kbytes
Use: Describing circuit to EMPOWER

This file contains a complete description of the circuit to be analyzed by EMPOWER. GENESYS will create this file automatically whenever EMPOWER is run from the EMPOWER menu in GENESYS. Even though this files can be edited, it will be overwritten if EMPOWER is rerun from within GENESYS.

.WSP (Workspace) Files

Written by: GENESYS
Type: Binary
Can be safely edited: Yes, but only using GENESYS
Average size: 10 to 2,000 Kbytes
Use: Contains complete simulation, graph, schematic, and layout information from GENESYS

Contains a complete GENESYS workspace.

.Y (Y-Parameter) Files

Written by: GENESYS
Type: Binary
Can be safely edited: No
Average size: 2 to 25Kbytes, but may be larger
Use: Internal data file for EMPOWER

This file contains the calculated Y-parameters before deembedding. If merge (-ME) is specified, the previous data stored in this file is combined with the newly calculated data, and the .SS (S-Parameter) file is rewritten.

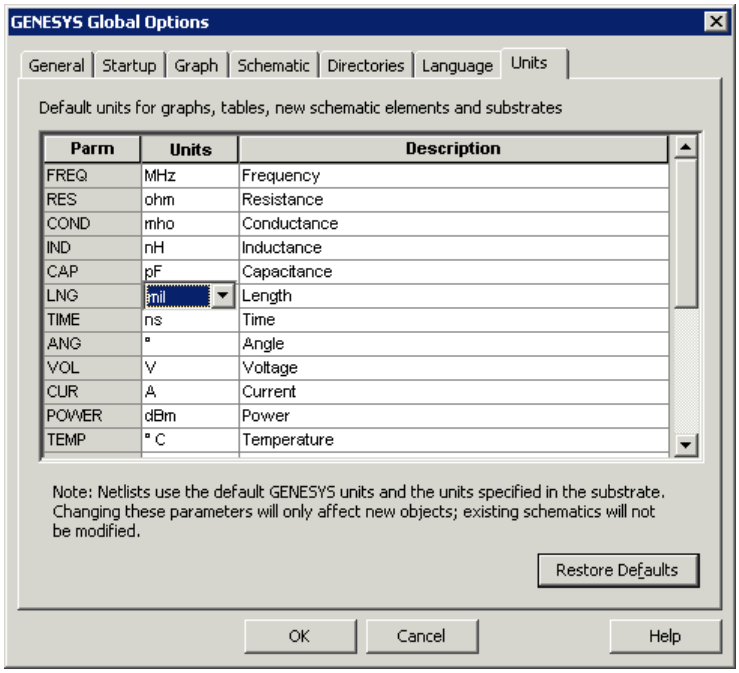
~SS, ~RG, etc. (Backup) Files

All files with a name or an extension starting with tilde (~) are backup files and can be safely deleted. Examples of these files are ~OMBINE.TPL and COMBINE.~RG.

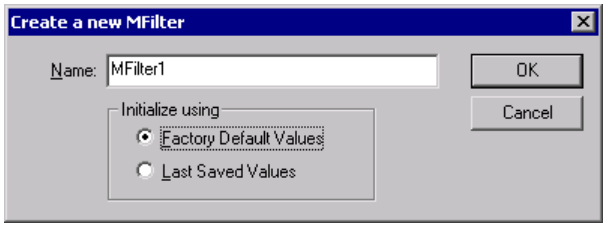
EMPOWER Advanced Example: Filter Synthesis

This advanced example shows how to combine M/FILTER, circuit simulation, and electromagnetic simulation. We will design a bandpass filter with a lower cutoff frequency of 2100 MHz and an upper cutoff frequency of 2300 MHz. We will use the M/FILTER module to design the filter, then we will perform a linear and EM simulation of the filter.

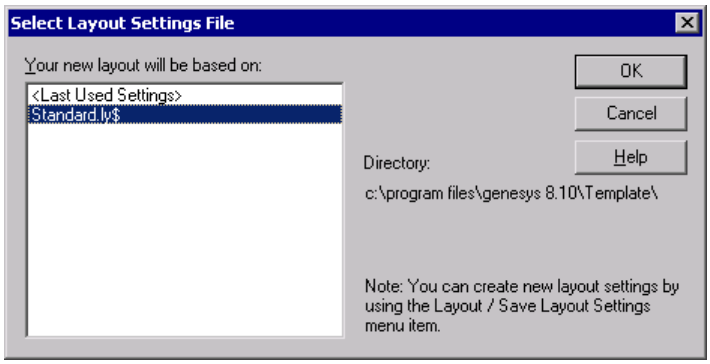
1. First, all units in this example use mils. In order to get the results in this example the default units should be changed to mils. This can be done by selecting 'Tools' and then 'Options' from the main menu then selecting the *Units* tab. Make sure the 'Length' parameter says 'mils' as shown below.



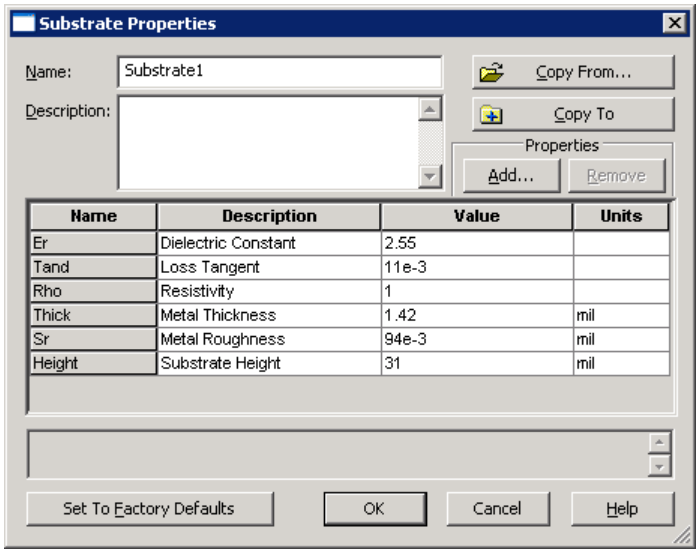
2. Next lets open the Microwave Filter module from the GENESYS tree to start the design process by selecting the New Item button and picking Synthesis / Add Microwave Filter.. In the 'Create a new Microwave Filter' dialog box change the 'Initialize using' to 'Factory Default Values' then select OK.



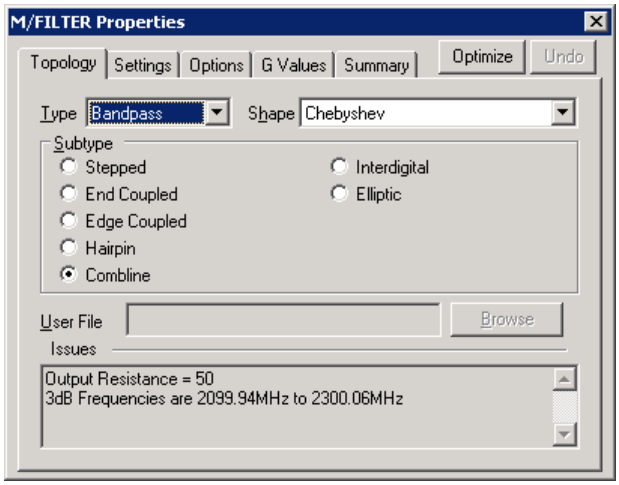
3. Now the user will be prompted for the printed wiring board layer settings in the 'Select Layout Setting File' dialog box. Select 'Standard.ly$' and then OK.



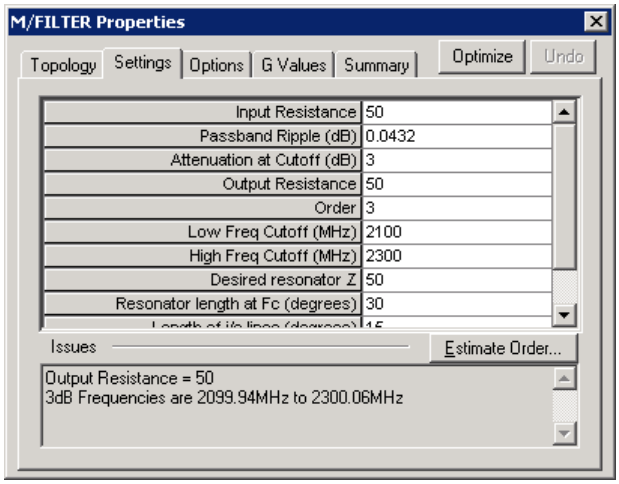
4. It will ask you to specify a substrate. For this example, just choose all default values, set Er=2.55, Height=31mil, and press OK.



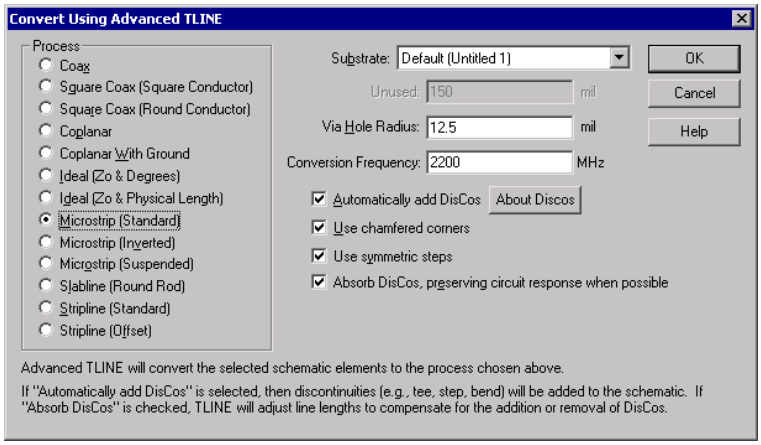
5. We want to choose 'Bandpass' as the type and 'Combline' as the Subtype. For this walkthrough we want to use the 'Chebyshev' filter shape. Your topology tab should look like what we have below. NOTE: Ignore locals errors created during this process since the design has not been completed.



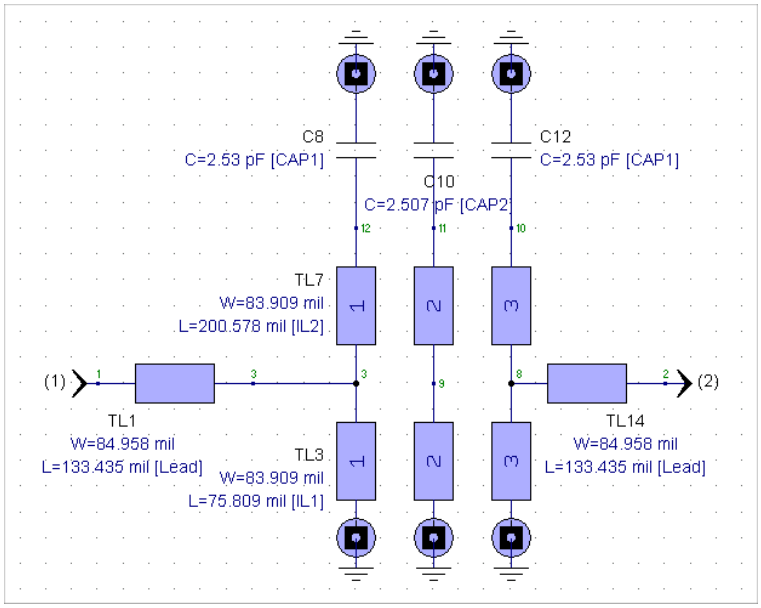
6. The next step is to specify all the filter parameters in the *Settings* tab as we have below.



7. Under the *Options* tab we should select the manufacturing process. For this example we will use microstrip (standard) as the Process. The conversion process should look similar to what we have below. Press OK and you should have an schematic that looks like ours (below).



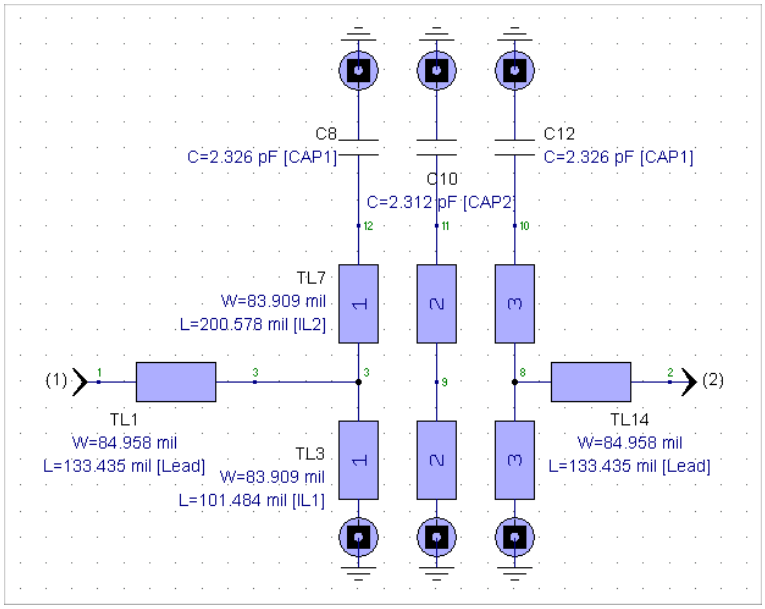
Schematic of the synthesized unoptimized filter:



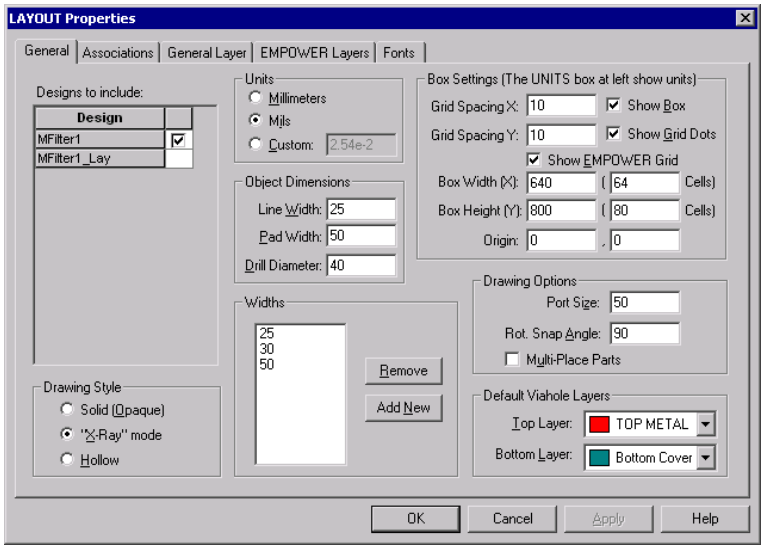
Notice how M/FILTER automatically inserts the discontinuities to model their effect.

8. The next we must optimize this filter by pressing the 'Optimize' button at the top of the MFilter dialog box. This is very important in order to obtain the expected filter performance in this example. After the optimization the schematic should now look as shown below.

Note: You should stop the optimizer by hitting the 'Esc' key once the error value is not improving much.



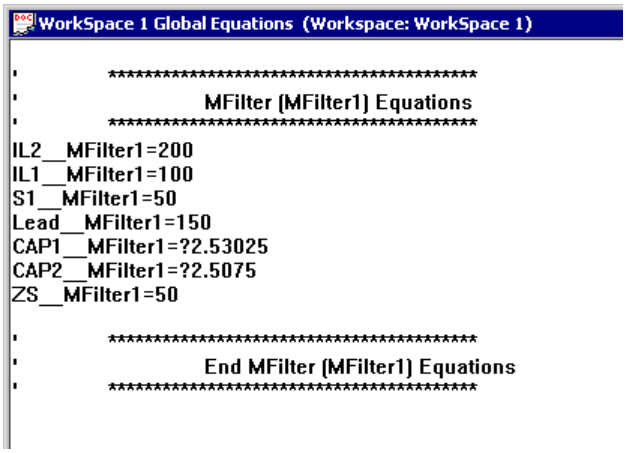
9. Now we need to set the board dimensions and the EMPOWER grid spacing. For this walkthrough we set the Grid Spacing X: 10 and the Grid Spacing Y: 10 and the Box Width (X) to 640 and the Box Width (Y) to 800. The other properties should be set as follows. We have chosen the grid spacing to be 10 because the widths and lengths of the synthesized filter are very close to multiples of 10 mils.



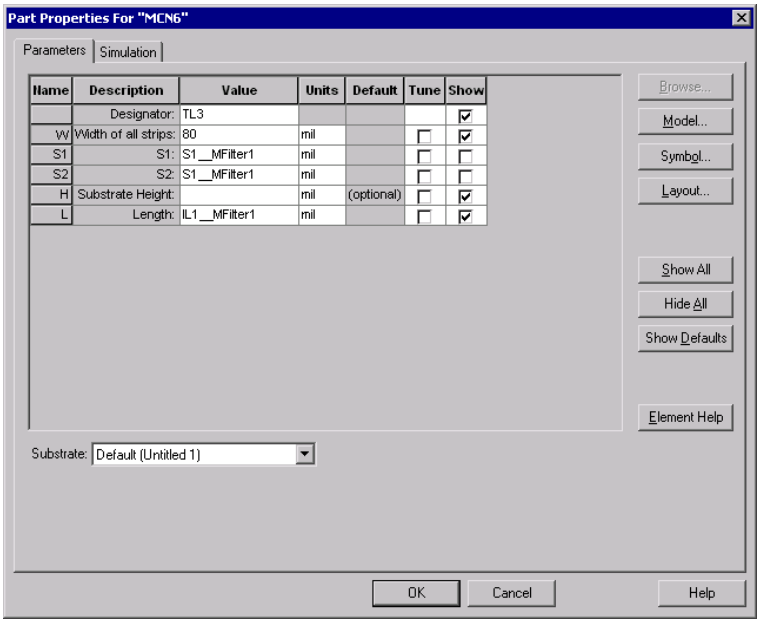
10. Now we need to slightly edit our layout. The ultimate goal is for the resonators dimensions to be an exact multiple of the grid dimensions. In this case the spacing between resonators is very important, therefore we will not change them

to much. First we need to change some of the dimensions in the equations block. Change the 'Lead_MFilter1', 'IL2_MFilter1', 'IL1_MFilter1', and 'S1_MFilter1' to the values as show below. Also be sure to remove the question mark since we will not reoptimize the layout dimensions.

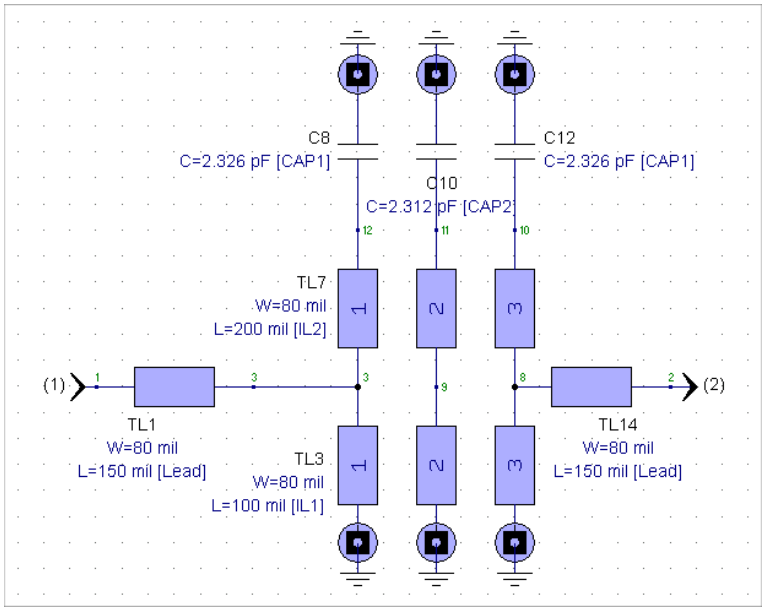
Note: The equation order may appear differently in your example.



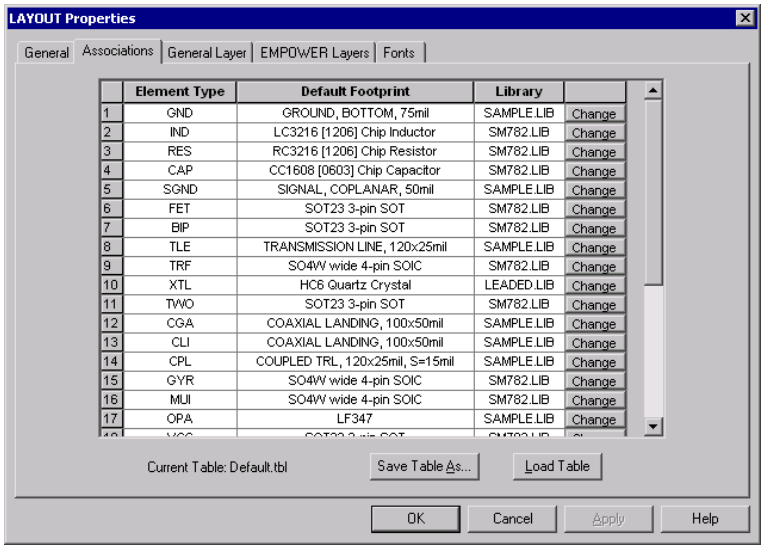
11. Next we need to change the resonator widths for all four transmission lines (TL1, etc.) by double-clicking on each of these schematic elements and changing the "Width" or "Width of all strips" to 80 mils.



12. The optimized readjusted filter schematic should now appear as follows.

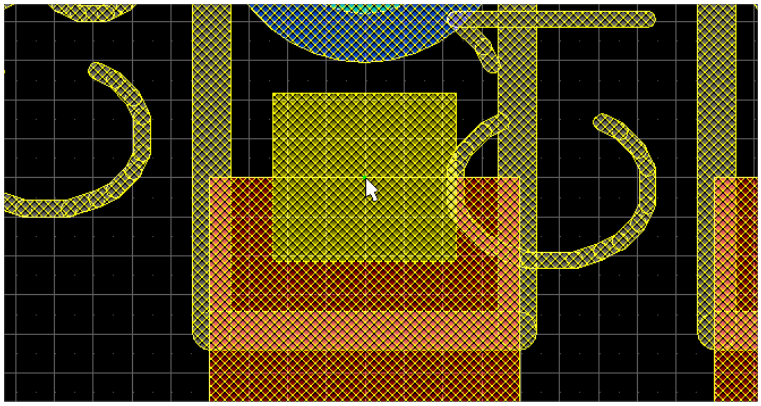


13. You also may need to change the capacitor footprints to 0603, (depending on your default footprints). This can be done by bringing up the 'Layout Properties' by double-clicking the layout background and then selecting the 'Associations' tab. Then proceed to click on the 'Change' button and choose 'CC1608 [0603] Chip Capacitor' from the 'SM782.LIB'.

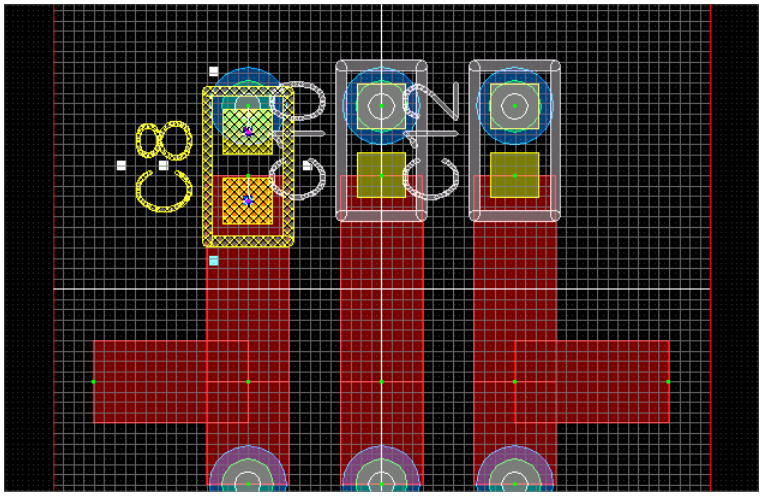


14. The next step is center the components on the PWB. This is done by selecting the 'MFilter1_Lay' Layout window and from the 'Edit' menu select 'Select All'.

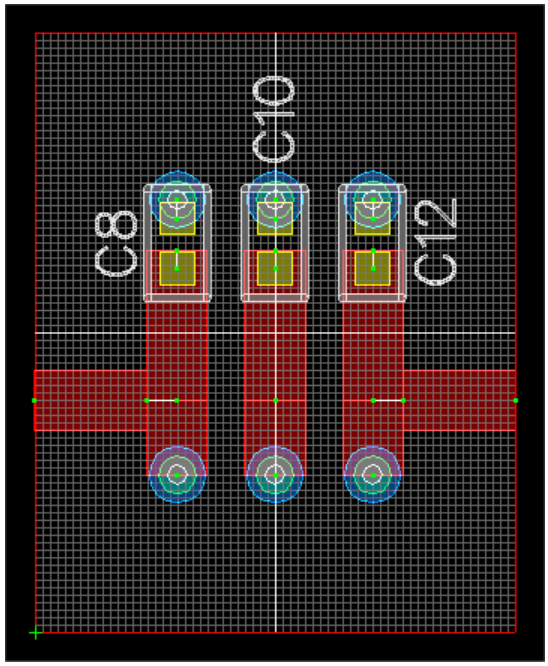
Then from the 'Layout' menu select 'Connect Selected Parts' and then 'Center Selected on Page'. Then click the zoom maximize button (crossed arrows). The layout should now be centered on the PWB. However, the transmissions lines might not be lined up exactly on grid. These can be placed on grid by placing the mouse over the center of one of the bottom capacitor pads and dragging the entire filter structure up to the nearest grid line (all parts must be selected in order to drag the entire filter).



15. We need to move the capacitor footprints so that the capacitor pads do not extend beyond the length of the resonators. Furthermore, the ground vias should be moved up so the drill hole is at the top of the upper capacitor pad, as shown on the left resonator.

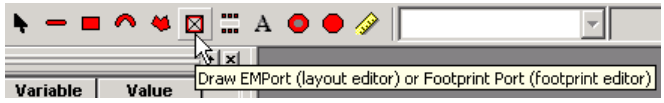


16. Next, we need to change the placement of the input and output lines. We want to pull them away from the center of the resonators to the edge to match the synthesized schematic by using the arrow keys. Your layout should look like what we have below.

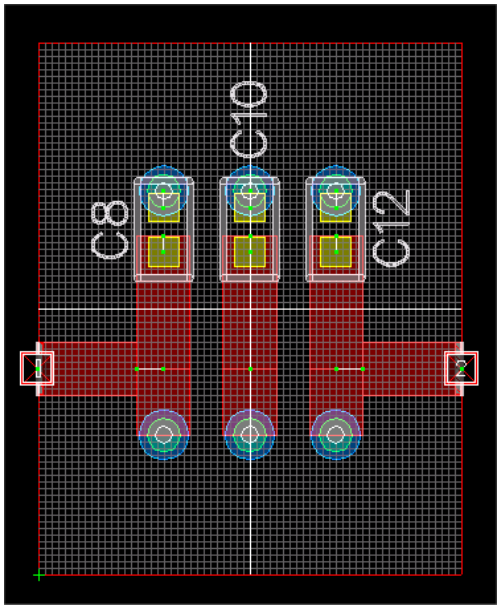


Tip: You can move the designator text, if desired, by grabbing the handle in the middle of the text block.

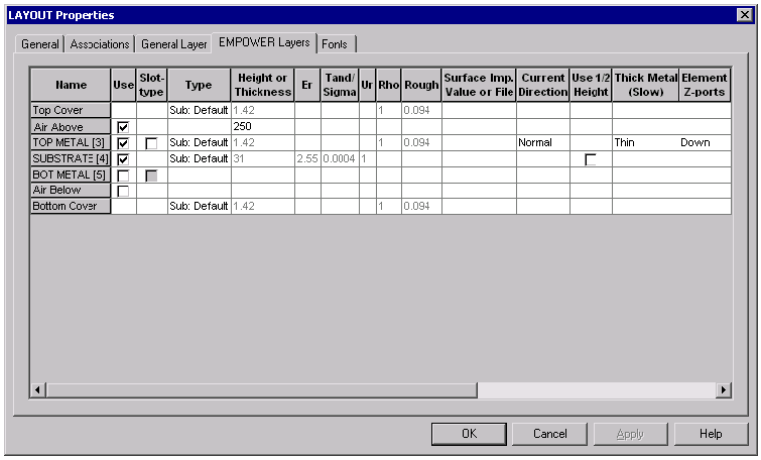
17. Now it is time to add an input and output port. EM ports are found in the GENESYS toolbar (below). The EM ports should line up exactly on the pwb edge and the gray shaded bar underneath the EM port will appear showing that they have been snapped correctly to the top metal layer.



Filter Layout of the schematic:

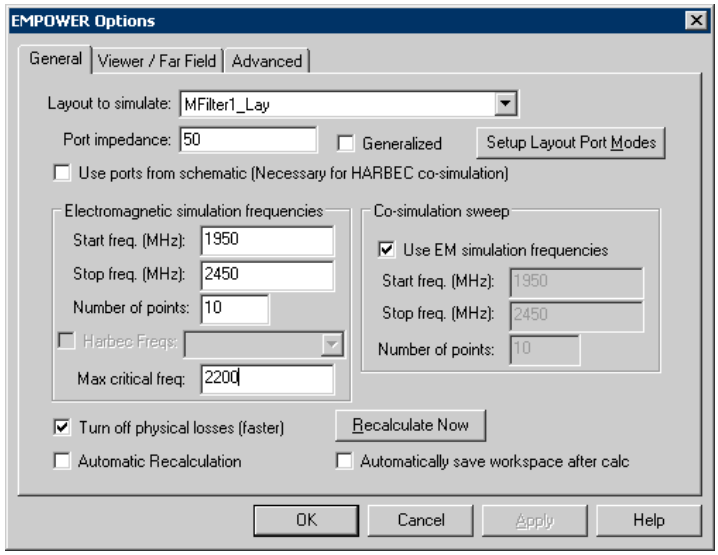


18. Next we need to set the EM layout properties as shown below. This can be done by bringing up the 'Layout Properties' by double-clicking the layout background and then selecting the 'EMPOWER Layers' tab.

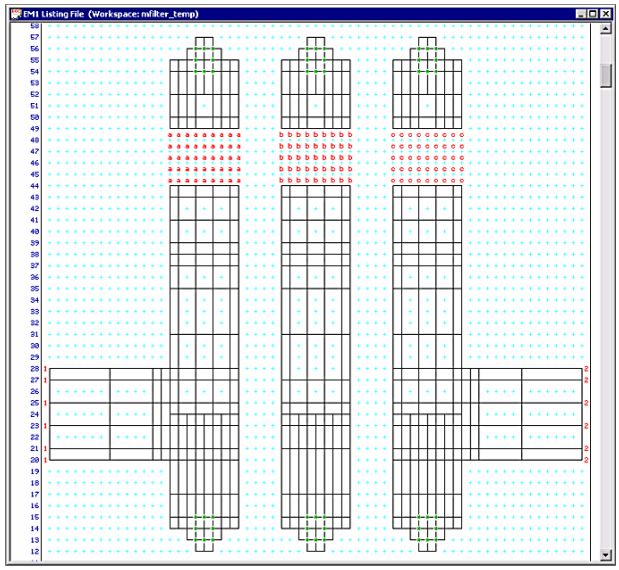


19. The next step is create an EM simulation of the layout. We do this by right-clicking on 'Simulations/Data' folder and selecting "Add Planar 3D EM Analysis". Then you should set up your simulation by changing the number of points. Your EMPOWER options should be setup just like ours below. Pressing "Recalculate Now" will start the EM simulator. This simulation takes a couple minutes on a Pentium III 500MHz CPU with 192MB of RAM.

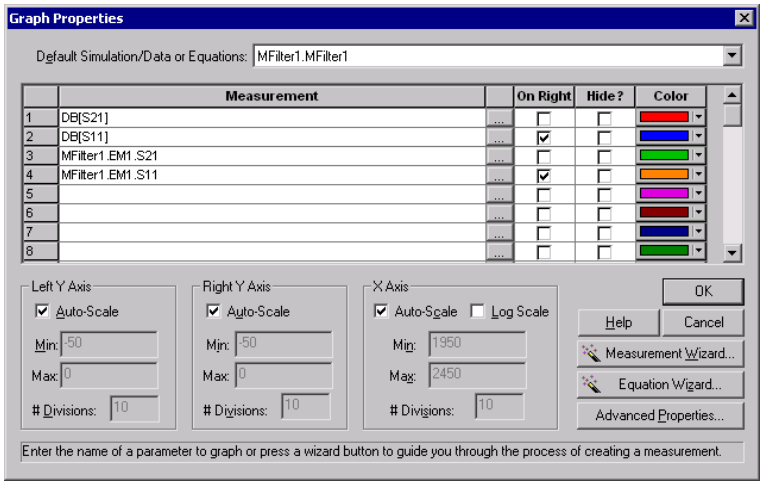
Note: We specified a wider simulation frequencies to get a bigger picture of the response.



20. After the simulation is done, you should first take a look at the Empower Listing file. Right click the Empower simulation titled EM1 (in the workspace tree) and select "Show Listing File". Inspect this file to verify the simulation geometry. As you can see, this listing matches the desired layout:

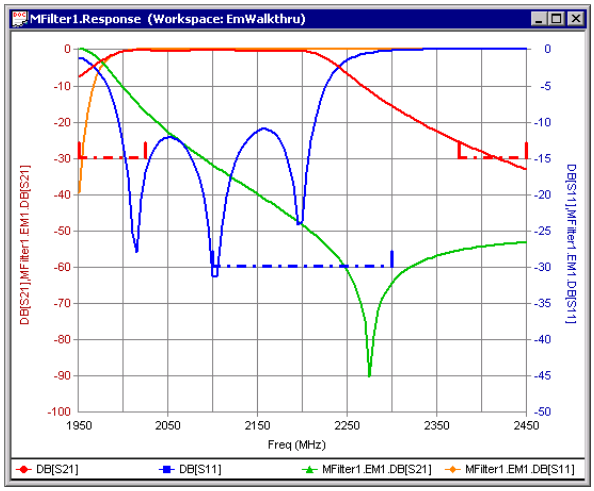


21. Now we would like to see the EM response on the same graph as the linear response to compare the two. We can do this by opening the linear graph's properties (MFilter1.Response) and typing in "MFilter1.EM1.DB[S21]" and "MFilter1.EM1.DB[S11]" on line three and four of the measurements just like it's illustrated below.



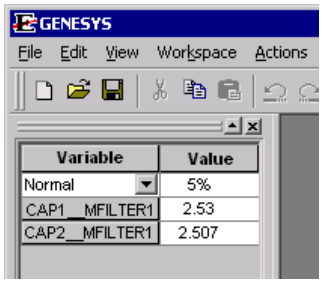
Note: You can also use the "Measurement Wizard" instead of manually typing in the measurement.

22. Now it's time to analyze what each simulation is showing us. Below is the graph which shows us both linear and EM simulations. Notice how the EM response is slightly down in frequency. The linear simulator does not take parasitic losses and box effects into account like the EM simulator does. The main reason why the EM response is shifted down in frequency is because the footprint pads (for the capacitors) actually add more capacitance to the filter. The filter responses are shown below. The red and blue response is S21 and S11 of the linear simulation, and the orange response is S21 and the green response is S11 of the EM simulation.

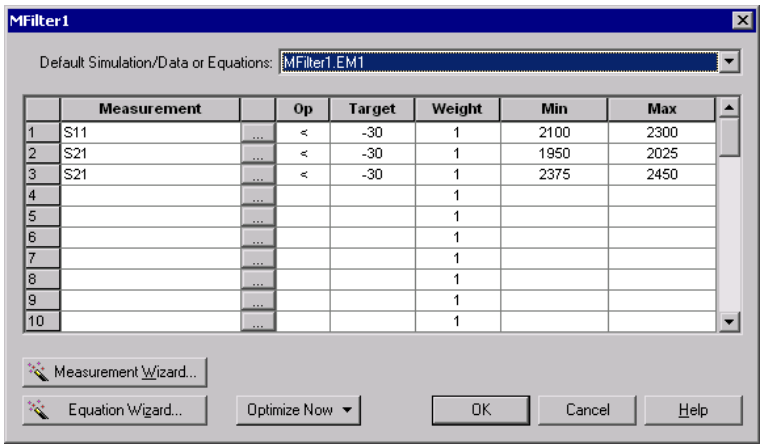


23. Now is the the time to see the true power of Eagleware's Co-Simulation. Co-Simulation allows your to tune your filter in real time without having to re-run

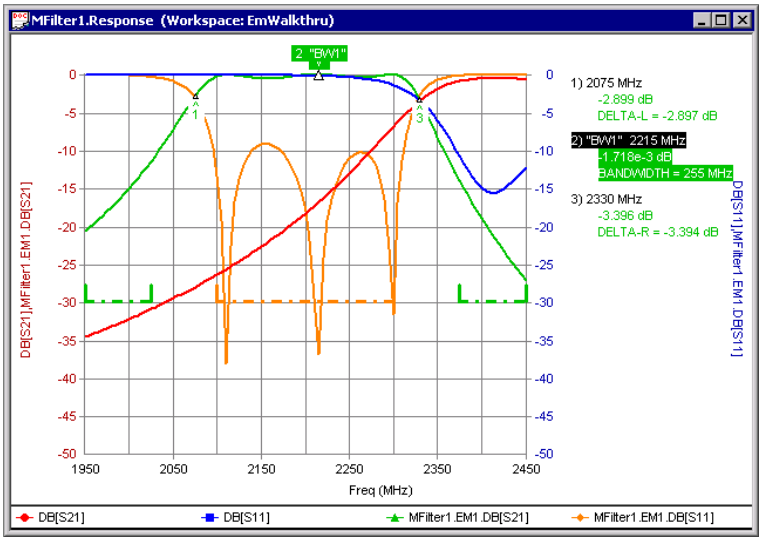
the EM simulation. In other words, you are able to tune your capacitor values without re-running EMPOWER! Since the response has shifted down in frequency we will need to decrease the amount of capacitance in all caps. We can manually change the capacitor values by tuning them in the tune window.



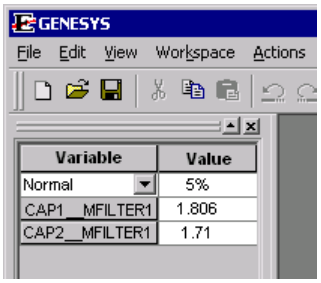
24. However, we can use the optimizer to tune the filter for us. We need to open up the optimization targets titled 'MFilter1' located under the 'Optimizations' folder and change the 'Default Simulation / Data or Equations' to 'MFilter1.EM1'. Then optimize the circuit by selecting 'Optimize Now' and 'Automatic'.



25. The final response of the optimized filter is as shown below. The last step is to press F5 on your keyboard to update the new traces. If you wish you can add a bandwidth marker to display the final result.



And the final capacitor values are:



Note that the original linear response is much higher in frequency than the electromagnetic simulation.

Chapter 29 EMPOWER: References

General Background

J.A. Stratton, *Electromagnetic theory*, McGraw-Hill Co., New-York, 1941

G. Kron, "Equivalent circuit of the field equations of Maxwell.-Part I.," Proc. of IRE, 1944, May, p. 289-299.

C.G. Montgomery, R.H. Dick, E.M. Purcell, *Principles of microwave Circuits*, McGraw-Hill Co., New-York,1948.

O. Heaviside, *Electromagnetic theory*, AMS Chelsea Publishing Co., New-York, 1950.

A.A. Samarskii, A.N. Tikhonov, "About representation of waveguide electromagnetic fields by series of TE and TM eigenwaves" (in Russian), GTF (Journal of Theoretical Physics), 1948, v. 18, p. 959-970.

P.I. Kuznetsov, R.L. Stratonovich, *The propagation of electromagnetic waves in multiconductor transmission lines*, Pergamon Press, Oxford, 1964 (originally published in Russian, 1958).

K.S. Yee, "Numerical solution of initial boundary value problems involving Maxwell's equations in isotropic media," IEEE Trans. v. AP-14, 1966, p. 302-307.

V.V. Nikol'skii, *Variational approach to internal problems of electromagnetics* (in Russian) , Moscow, Nauka, 1967.

J. Meixner, "The behavior of electromagnetic fields at edges", IEEE Trans, v. AP-20, 1972, N 7, p.442-446.

B.V. Sestroretzkiy, "RLC and Rt analogies of electromagnetic space" (in Russian), in Computer aided design of microwave devices and systems, Edited by V.V. Nikol'skii, Moscow, MIREA, 1977, p. 127-128.

T. Weiland, "Eine Methode zur Losung der Maxwellschen Gleichngen for Sechskomponentige Feleder auf Dikreter Basis", Arch. Electron. Uebertragungstech., v. 31, N 3, 1977, p.116-120.

Computer-aided design of microwave devices (in Russian), Edited by V.V. Nikol'skii, Moscow, Radio i Sviaz', 1982.

R.H. Jansen, "The spectral-domain approach for microwave integrated circuits," IEEE Trans., v. MTT-33, 1985, N 10, p. 1043-1056.

S.G. Vesnin, *Electromagnetic models for design of microstrip microwave structures* (in Russian), Ph.D. Thesis, MPEI, Moscow, 1985.

E.F. Johnsom, "Technique Engineers the Cavity Resonance in Microstrip Housing Design," MSN & CT, 1987, Feb., p. 100-102, 107-109.

J.C. Rautio, R.F. Harrington, "An electromagnetic time-harmonic analysis of shielded microstrip circuits," IEEE Trans., v. MTT-35, 1987, N 8, p. 726-730.

B.V. Sestroretzkiy, V.Yu. Kustov, "Electromagnetic analysis of multilevel integrated circuits on the base of RLC-networks and informational multiport approach" (in Russian), Voprosi Radioelektroniki, ser. OVR, 1987, N 1, p. 3-23.

L.P. Dunleavy, P.B. Katehi, "A generalized method for analyzing shielded thin microstrip discontinuities", IEEE Trans., v. MTT-36, 1988, N 12, p.1758-1766.

T. Uwaro, T. Itoh, "Spectral domain approach," in *Numerical techniques for microwave and millimeter-wave passive structures*, Edited by T. Itoh, John Willey & Sons, 1989.

R.H. Jansen, "Full-wave analysis and modeling for CAD of mm-wave MMICs," Alta Frequenza, v. LVIII, 1989, N 5-6, p. 115-122.

A. Hill, V.K. Tripathi, "An efficient algorithm for the three-dimensional analysis of passive microstrip components and discontinuities for microwave and millimiter-wave integrated circiuts", IEEE Trans., v. MTT-39, 1991, N 1, p. 83-91.

The Method of Lines

M.G. Slobodianskii, "A new method of approximate solution of partial differential equations and its application to the theory of elasticity" (in Russian), Prikladnaia Matematika i Mekhanika (Applied Mathematics and Mechanics), v. 3, 1939, N 1, p. 75-82.

O.A. Liskovets, "The method of lines, Review" (in Russian), Differenzial'nie Uravneniya, v. 1, 1965, N 12, p. 1662-1668.

B.L. Lennartson, "A network analogue method for computing the TEM characteristics o planar transmission lines," IEEE Trans., v. MTT-20, 1972, N 9, p. 586-590.

U. Schulz, "On the edge condition with the method of lines in planar waveguides," Arch. Electron. Uebertragungstech., v. 34, 1980, p.176-178.

U Schulz, R. Pregla, "A new technique for the analysis of the dispersion characteristics of planar waveguides and its application to microstrips with tuning septums," Radio Science, v. 16, 1981, Nov.-Dec., p. 1173-1178.

S.B. Worm, R. Pregla, "Hybrid-mode analysis of arbitrarily shaped planar microwave structures by the method of lines," IEEE Trans., v. MTT-32, 1984, N 2, p. 191-196.

R. Pregla, W. Pascher, "The method of lines," in Numerical techniques for microwave and millimeter-wave passive structures, Edited by T. Itoh, John Willey & Sons, 1989.

S.B. Worm, "Full-wave analysis of discontinuities in planar waveguides by the method of lines using a source approach," IEEE Trans., v. MTT-38, 1990, N 10, p.1510-1514.

Richardson's Extrapolation

L.F. Richardson,"The differed approach to the limit. 1:Single lattice," Philos. Trans. of Royal Society, London, ser. A, 226, 1927, p.299-349.

A. Premoli, "A new fast and accurate algorithm for the computation of microstrip capacitances," IEEE Trans. v. MTT-23, 1975, N 8, p. 642-647.

G.I. Marchuk, V.V. Shaidurov, *Difference methods and their extrapolations*, Spr.-Verlag, 1983 (originally published in Russian, 1979).

A.G. Vikhorev, Yu.O. Shlepnev, "Analysis of multiple-conductor microstrip lines by the method of straight lines," Journal of Communications Technology and Electronics, 1991, N 12, p. 127-129, originally published in Radiotekhnika i Elektronika, v. 36, 1991, N 4, p. 820-823.

Symmetry Processing

M. Hammermesh, *Group theory and its application to physical problems*, Pergamon Press, Oxford, 1962.

I.J. Good, "The inverse of a centrosymmetric matrix," Technometrics, Journal of Statictics for Physical Chemical and Engineering Science, v. 12, 1970, p. 925-928.

P.R. McIsaac, "Symmetry-induced modal characteristics of uniform waveguides,Part I:Summary of results,Part II:Theory," IEEE Trans., v. MTT-23, 1975, N 5, p.421-433.

W.T. Weeks, "Exploiting symmetry in electrical packaging analysis," IBM Journal of Research and Development, v. 23, 1979, N 6, p.669-674.

A.B. Mironov, N.I. Platonov, Yu.O. Shlepnev, "Electrodynamics of waveguiding structures of axisymmetrical microwave integrated circuits," Journal of Communications Technology and Electronics, 1990, N 7 p. 71-76, originally published in Radiotekhnika i Elektronika, v. 35, 1990, N 2, p. 281-286.

E.V. Zakharov, S.I. Safronov, D.P. Tarasov, "Abelian Groups of finite order in numerical solution of potential theory boundary value problems" (in Russian), GVM & MF (Journal of Computational Mathematics and MathematicalPhysics), v. 32, 1992, N 1, p. 40-58.

EMPOWER Engine Theory and Algorithms

B.V. Sestroretzkiy, V.Yu. Kustov, Yu. O. Shlepnev, "Analysis of microwave hybrid integrated circuits by informational multiport network method" (in Russian), Voprosi Radioelektroniki, ser. OVR, 1988, N 12, p. 26-42.

B.V. Sestroretzkiy, V.Yu. Kustov , Yu.O. Shlepnev, "Technique of electromagnetic analysis of microstrip devices using general purpose programs" (in Russian), Voprosi Radioelektroniki, ser. OVR, 1990, N 1, p. 3-12.

Yu.O. Shlepnev, *Method of lines in mathematical modeling of microwave integrated circuit planar elements* (in Russian), Ph.D. Thesis, NEIS, Novosibirsk, 1990.

V.Yu. Kustov, B.V. Sestroretzkiy, Yu.O. Shlepnev, "Electromagnetic analysis of planar devices with resistive films and lumped elements," Proc. of Europ. Symp. on Numerical Methods in Electromagnetics (JEE'93), Toulouse, France, 17-19 November, 1993, p. 227-234.

V.Yu. Kustov, B.V. Sestroretzkiy, Yu.O. Shlepnev, "Three-dimensional electromagnetic analysis of planar devices with resistive films and lumped elements," Proc. of 27th Conference on Antenna Theory and Technology (ATT'94), Moscow, Russia, 23-25 August, 1994, p. 352-356.

K.N. Klimov, V.Yu. Kustov, B.V. Sestroretzkiy, Yu.O. Shlepnev, "Efficiency of the impedance-network algorithms in analysis and synthesis of sophisticated microwave devices," Proc. of the 27th Conference on Antenna Theory and Technology (ATT'94), Moscow, Russia, 23-25 August, 1994, p. 26-30.

V.Yu. Kustov, B.V. Sestroretzkiy, Yu.O. Shlepnev, "TAMIC package for 3D electromagnetic analysis & design of MICs," Proc. of the 5th Intern. Symp. on Recent Advances in Microwave Technology (ISRAMT'95), Kiev, Ukraine, September 11-16, 1995, p. 228-233.

Yu. O. Shlepnev, B.V. Sestroretzkiy, V.Yu. Kustov, "A new method of electromagnetic modeling of arbitrary transmission lines," Proc. of the 3rd Int. Conference Antennas, Radiocommunication Systems and Means (ICARSM'97), Voronezh, 1997, p.178-186.

Yu.O. Shlepnev, B.V. Sestroretzkiy, V.Yu. Kustov, "A new approach to modeling arbitrary transmission lines," Journal of Communications Technology and Electronics, v. 42, 1997, N 1, p. 13-16, originally published in Radiotekhnika i Elektronika, v. 42, 1997, N 1, p. 13-16.

Yu.O. Shlepnev, "A new generalized de-embedding method for numerical electromagnetic analysis," Proceedings of the 14th Annual Review of Progress in Applied Computational Electromagnetics, Monterey, CA, March 16-20, 1998, v.II, p. 664-671.

Yu.O. Shlepnev, "Extension of the method of lines for planar 3D structures," Proceedings of the 15th Annual Review of Progress in Applied Computational Electromagnetics, Monterey, CA, 1999, p. 116-121.

Test Examples and Comparisons

E.G. Farr, C.H. Chan, R. Mittra, IEEE Trans., v. MTT-34, 1986, N 2, p. 307.

G. Gronau, I. Wolff "A simple broad-band device de-embedding method using an automatic network analyzer with time-domain option", IEEE Trans., v. MTT-37, 1989, N 3, pp. 479-483.

D.J. Swanson, "Grounding microstrip lines with via holes", IEEE Trans., v. MTT-40, 1992, p. 1719-1721.

J.C. Rautio, "An ultra-high precision benchmark for validation of planar electromagnetic analysis", IEEE Trans., v. MTT-42, 1994, N 11, p. 2046-2050.

T. Kawai, I. Ohta, "Planar-circuit-type 3-dB quadrature hybrids", IEEE Trans., v. MTT-42, 1994, N 12, p. 2462-2467.

Y. Gao, I. Wolff, "Miniature electric near-field probes for measuring 3-D fields in planar microwave circuits", IEEE Trans., v. MTT-46, 1998, N 7, p. 907-913.

Index

2

2-port 179

3

3D 139

A

ABCD 32
ABS 145
Absolute Error 43
Absolute Tolerance 44, 53
AbsTol 41
AC 41
Accuracy 261, 273, 275
ACF 208
ACP 207
Added Noise 208
Adding
  Link 157
Adding 157
Adjacent Channel Frequency 208
Adjacent Channel Power 207
Admittance 36, 179
Air Above 257, 275
Air Below 257
Algorithms 325
Amplifier 24, 100
Amplifiers 100
Amplitude Stepping 54
AN 208
Analyzer 94
AND 144
ANG 145, 181
ANG360 ..... 145, 181
Angle 181, 303
Animate button 303, 311
Annotate 41
ARCCOS 145
ARCCOSH 145
ARCSIN 145
ARCSINH 145
ARCTAN 145
ARCTANH 145
Array Index 144
Arrays 144, 145, 148, 150
Artificial intelligence techniques 54
Associations 165
ATN 145
Automatic 2 Tone 112
Automatic deembedding 1
Automatic Port Placement 298
Automatic Recalculation 41, 44
Automatically Calculate 51
Available gain circles 179

B

B134, 179
Back-annotation
  DC 41
Back-annotation 41
Balanced amplifiers 160
BASE 141
Basic Geometry 257
Batch Runs 271
Berkeley 176, 177
BESSELJ0 145
Binary Files 338
Bottom Cover 257, 264
Box 1, 242, 261, 274, 281, 297
Box Modes 1, 275, 321, 322, 323
Broadband Noise 116
Built-in Functions 145, 150
BYREF 154

C

Calculate Intermods Along Path 90, 112
Calculate Noise 90
Calling C/C++ Programs 155
Carrier to Noise Ratio 211
Cascaded Gain 209, 210, 211
Cascaded Noise Figure 133, 212
Cavity Absorber 323
Cavity resonators 321
Cell Size 261, 273, 276, 311, 325
Centering 247
CF 213
CGAIN 209
CGainAll 211
CGAINIM3 210
Chamfered corners 261
Channel (Path) Frequency 104
Channel Frequency 104
Channel Noise Power 215
Channel Power 215
Characteristic impedance 286

Characteristics 1
CIM3P 234
Circles 32, 38, 39, 179, 183
CNF 212
CNP 215
CNR 211
Coaxial T-junction 257
Coherency 106
Coherent Addition 90
COMBINE.WSP 290
Compensation Admittance 340
COMPLEX 145
Components 261, 297
Composite Spectrum 122
Compression 44, 53, 157
Concatenation 148
Conducted Third Order Intermod Power ... 234
Console Window 270
Constant noise circles 179
Constants 148, 179
Contours 205
Convergence 41, 43, 44, 53, 54, 325
Coplanar 277
Copper 257
COS 145
COSH 145
COUNT 145, 148, 150
Coupled Microstrip 315
Cover Spacing 275
Cover Type 276
Covers 257, 281, 297
CP 215
Creating
  New Data Files 158
Creating 158
Creation Equation 124
Current Dir 281
Current/Viewer Data 341
Current/voltage 311

D

Data 1
Data File 157, 158, 160
DB 150, 179, 181
DB Magnitude 183
DB10 145
DB20 145
DBANG 179, 181
DBANG360 181
DBm operator 53
DBMAG 179
DC 41
DC analaysis 41
DC Analysis Overview 41
DC Analysis Properties 41
DC biasing 1
DCP 216
DCPIM3 217
Decomposition 284, 289, 290, 295
Deembedded ports 298
Deembedding 281, 283, 343
De-Embedding Algorithm 335
Default Operator 179, 181
Default Simulation/Data 183
DEFAULT.MOD file 174
Delete This Simulation/Data 51
Derivatives 44
Desired Channel Power 216, 217
Diagonalization 325, 326
Dielectric 257, 323
Dielectric Constant 264
Dielectric loading 322
Dim 150
Dimensions 242, 257
Dimensions Tab 261
Directional Energy 118
Discontinuities 283
Discretisation 325
Discretizes metal 1
Distortion 53
Distribution 1, 316
DLLs 155
Draw Size 281
DTOR 148

E

Eagleware 1
Edit Menu 274
Effective noise input temperature 179
Eigenmode 315, 317
Eigenwave 317, 319
Eij 179
Electromagnetic 1
Electromagnetic simulation 1, 44, 257
Elements 165, 173
EMport
  Options 281
EMport 261, 265, 281
EMPOWER 1, 345
EMPOWER viewer 311
EMPOWER Walkthrough 345
EMPOWER, Viewer 251
EMPOWER, Viewing Results 250
EMV 317, 339
Engine Theory 325
EPS0 148

Equality Check 144
Equation Wizard 153
Equations ..1, 144, 145, 148, 150, 154, 165, 167, 181, 183, 184
Equivalence 144
ETA0 148
Exclusive Or 144
Exclusive-OR 154
EXP 145
EXP1 148
Exponentiation 144
Exporting Data Files 160
Exporting 160
Expression 145
Extensions 338
External Ports 281, 297, 337
Extra Details 340
Extrapolate 150

F

Fast Intermod Shape 90
Fast Newton 44, 53, 54
FFT 44
FIX 145
FN_E 145
FN_K 145
Force 1-D FFT 44
FOU 157
FREQ 150
Frequencies 31
Full Jacobian 44, 54
FUNCTION 141, 154

G

GA 35, 37
GA Circles 179
GAIN 219, 220
Gain Circles 32, 37, 179
GAINALL 220
GAINIM3 219
GD 181
General Background References 359
General Layer 257
General Layers 242
Generalized 325, 326
Generalized scattering 317
Generalized S-Parameters 286, 341
Generate Viewer Data .. 276, 303, 315, 317, 319, 339
Generated Third Order Intermod Power 235
GET 145
GETINDEPVALUE 145, 150
GETVALUE 145, 150
GETVALUEAT 145, 150
GIM3P 235
Glossary 87
GM 181
GM1 36
GM2 36
GMAX 32, 36, 37, 179, 181
GMi 179
Gmin 41, 44
GminSteps 41
GOPT 160, 179, 181
GOTO 141
GP 35, 37
GP Circles 179
Graphs 181, 303
Greater Than 144
Green's function 326
Grid 261, 274, 297, 329
Grid Green's Function 331
Grid mapping 340
Ground Plane 1, 281, 297
Gt35
GU1 37
GU1 Circles 179
GU2 37
GU2 Circles 179

H

H Parameters 179
HARBEC 5, 43, 53, 176
HARBEC Options 44, 53, 54
HARBEC Popup Menu 51
HARBEC, Convergence Issues 53
HARBEC, Measurements 53
HARBEC, Optimization 54
Harmonic Balance 43, 53, 54
Harmonic Balance Walkthrough 5
HB_dfRelRec 44
HB_dxAbs 44
HB_dxRel 44
HB_NonBinaryFFT 44
HB_Oversampl 44
Highest accurate frequency 264
Homogeneous 321, 322, 325
Hybrid Linear / Nonlinear Model 101
Hyperbolic 145

I

Identification 124
Identifying Spectral Origin 124

IF 141
IF THEN GOTO Statement....141, 154
IFF....145, 150
IF-THEN statements....150
IFTRUE....145, 150
Ignore Spectrum....97
IIP3....112, 232
IM....145, 181
IM3 Pass....112
IMAG....145
Image Channel Noise Power....222
Image Channel Power....224
Image Frequency....221
Image Noise Rejection Ratio....223
Image Rejection Ratio....225
IMGF....221
IMGNP....222
IMGNR....223
IMGP....224
IMGR....225
IMP....154
Impedances....36, 39, 179, 286, 342
Implication....144
Informational Multiport....332
Input Third Order Intercept....232
Input VSWR....32
INT....145
Integer Division....144
Interdigital....289
Intermod....111
Intermods....112
Intermods and Harmonics....90, 108
Intermodulation....44, 53
Intermodulation Distortion, SPECTRASYS112
Internal Ports....264, 265, 290, 297, 317, 326
Interpolate....150
Iprobe....181
Itl1....41

J

Jacobian....44, 53, 54

K

K 179
Krylov....44, 54

L

L1 339
LABEL....141
Layer Tab....257, 276
Layers....242, 243, 264, 279, 319, 326
LAYOUT
  Creating....240
  Drawing....245
  Simulating....248
LAYOUT....240, 245, 248
Less Than....144, 145
Level Diagrams....121
Library....177
Line Direction....281
Line impedance....275, 340
Linear Magnitude....181, 183
Linear Measurements....179
Linear Simulation....1, 31
Linear Simulation Properties....31
Linear S-parameters....32
Link....157, 158, 177
Listing File....274, 340
LN....145
LN2....148
LNMIT3.WSP....315
Load Pull Contours....205
Loaded Q....181, 183
LOG....145
Logical Operators....154
Loss....276
Loss Tangent....257
Losses....295
Lossy metals....277
Lumped Elements....255, 265, 299, 326

M

MAG....145, 181
MAGANG....179, 181
MAGANG360....181
Magnetic Wall....257
Magnitude....311
Manufacturers....157, 163
Matrices....148
MATRIX....145
MAX....145
Maximum Amplitude Step....54
Maximum Mixing Order....54
Maximum Order....90
Maximum stable gain....36
Maxwell's equations....325
MDS....223
Measurement Bandwidth....104
Measurement Wizard....153, 179
Measurements....32, 41, 53, 104, 117, 145, 150, 179, 181, 183, 184, 317
Measuring
  S-parameters....32
Measuring....32

MESFET's 176
Metal 257, 275, 279
Metalization 311
Metallization layer 257, 277
Method of Lines 328
Microstrip 1, 257, 261, 275
MIN 145
Minimum Detectable Signal 223
Mirror 274
MIXER 26
Mixer LO 97
Mixers 113
MMTLP 339, 342
Mode Setup box 290
Model 157, 165, 167, 173
Model Editor 174
Model Properties 172
MSG 36
MU0 148
Multi-dimensional 150
Multidimensional FFTs 44
Multimode 317, 325
Multi-mode 284, 289, 290, 295, 315
Multimode lines 295
Multiplication 144
Multiplier 148

N

NCI 183
NCI Circles 179
New Data Files
  Creating 158
New Data Files 158
Newton-Raphson 54
NF 179, 181
NFMIN 179, 181
NFopt 160
NFT 179
NMEAS 179, 181
No Deembedding 265
Noise
  Optimal impedance 179
Noise 116, 179
Noise Circles 38, 179, 183
Noise correlation 160, 179
Noise Data 160
noise figure decrease 133
Non-linear 1, 41
Nonlinear Device Library 163
Nonlinear Device Models 157, 175
Nonlinear JFET 176
Nonlinear Measurements 181
Nonlinear MESFET Transistors 176
Nonlinear model 52
Nonlinear MOSFETs 176
Non-standard metal 1
Normal deembedded 265
Normal ports 281
Normalized noise resistance 179
NOT 154
Notes 181
NPO 157
Numerical Acceleration Procedures 333

O

Oblique button 311
OCF 214
OCP 217
Odd Order Only 90
Offset Channel 105
Offset Channel Frequency 214
Offset Channel Power 217
OIP3 233
ONE 157
One-dimensional FFT 44
Operations 150
Operator descriptions 144
Operators 53, 148, 154, 181, 183
Opt/Yield/Recalc 51
Optimal admittance 179
Optimal gamma
  noise 179
Optimal gamma 179
Optimization 179, 181, 183
Optimizing Simulation Performance 54
OR 154
OSCILLATOR 44, 60
Oscillator Design 32
Oscillator Walkthrough 60
Out-of-bounds 148
Output 144
Output Equations 150
Output Third Order Intercept 233

P

Pads 297
Parameter Sweep 53, 139
Parameter Sweep Properties 139
Parameters 165, 179
Parasitics 1
Partial Dielectric Loading 323
Patch antenna 281
path 89, 120
Path Frequency 118
Path Spectrum 120

Paths 117
Percent Noise Figure..... 226
Percent Third Order Intermod..... 227
Permeability 326
Permitivity.....315, 316
Permittivity 257, 326
Phase Noise 1
PI 148
Planar..... 299, 325, 326
PLX..... 341
PLX text file 339
Port Impedance.....158, 341
Port Number 179
Port Type 281, 297
Ports.....247, 261, 265, 284, 296, 298, 299, 326
Post-processing.....145, 150, 153, 179, 181, 183, 184
Pours.....1
Precedence..... 144
Preferred Cell Count 277
PRIM3..... 227
PRNF..... 226
Problem Formulation..... 326
Provided Device Data..... 158

Q

QL..... 181

R

R1341
Radians multiplier..... 148
RE 145, 181
REAL 145
Real-Time Tuning..... 256
Recalculate Now 51
Recalculation button 51
Record Keeping 160
RECT 179, 181
Rectangular Cavity.....321, 322
Rectangular waveguides 326
REF..... 141
Ref Plane Shift..... 281
Reference Plane..... 284, 290, 295
Reflection Coefficient 32, 35, 36, 39
Relational 154
Relative Dielectric Constants..... 257
Relative Error..... 43
Relative permittivity 257
Relative Tolerance 44, 53
RelTol..... 41
Resistance..... 179
Resistivity..... 257
Resonance.....275, 301, 321, 323
RETURN.....141, 154
Reverse Nodes 177
RGF..... 342
Rho 257
Richardson..... 360
RN.....160, 179
RND 145
Roughness..... 257
RTOD 148
RX..... 342

S

S Parameters.....179, 183
Sample Expressions..... 145
Sample Measurements 183
SB1..... 196
SB1 Circles..... 179
SB2 Circles..... 179
Scalar/matrix combination..... 148
SDR 229
Semi-Infinite Waveguide 257
Sensitivity.....38
Setup Modes dialog box 290
SFDR..... 228
Signal Metal Effects..... 323
Simplify 1
Simulation.....31, 183
Simulation/Data 183
Simulations/Data.....1
Simultaneous match impedance 179
SIN..... 145
Single Part model.....52, 173
SINH 145
Slot-type 276, 277
Smith Chart 32, 39, 179, 181, 183
SMTLP..... 290, 339, 342
SNF..... 229
SOIP2..... 231
SOIP3..... 231
Solid/Wire button 303, 311
SOP1DB 230
SOPSAT..... 232
Sources 127
S-parameter.....35, 158, 160
S-Parameters..... 179, 185, 340
Special Options 44
Spectral Origin 124
SPECTRASYS 17, 18
SPECTRASYS, Broadband Noise..... 116
SPECTRASYS, Channel Frequency..... 104
SPECTRASYS, Coherency 106
SPECTRASYS, Composite Spectrum..... 122

SPECTRASYS, Creating a Schematic..... 17
SPECTRASYS, IIP3 Distortion..... 112
SPECTRASYS, Intermods and Harmonics. 108
SPECTRASYS, Level Diagrams..... 121
SPECTRASYS, Measurement Bandwidth ... 104
SPECTRASYS, Mixer..... 26
SPECTRASYS, Offset Channel..... 105
SPECTRASYS, Options..... 105
SPECTRASYS, Sources..... 127
SPECTRASYS, Spectral Origin..... 124
SPECTRASYS, Tone Channel Frequency... 214
Spectrum..... 120
SPICE..... 1, 52, 177
SPICE File Compatibility..... 176
Spiral Inductor..... 275, 289, 290, 295, 296
spreadsheets..... 134
Spurious Free Dynamic Range..... 228
SQR..... 145
SrcSteps..... 41
SS 342
Stability..... 179, 183
Stability Circles..... 196
Stability Factor..... 196
Stability Measure..... 196
Stage Dynamic Range..... 229
Stage Noise Figure..... 229
Stage Output 1 dB Compression Point..... 230
Stage Output Saturation Power..... 232
Stage Output Second Order Intercept..... 231
Stage Output Third Order Intercept..... 231
Strings..... 145, 148
Stripline..... 257
Subdirectories..... 337
Substrate..... 257
Substrate layer..... 257, 264
Substrate thickness..... 275
Subtraction..... 144
Superconductors..... 257
SuperStar..... 1
Surface Roughness..... 257
Suspended Microstrip..... 257
Sweep..... 31, 139
Symmetry..... 274, 325
Symmetry processing..... 340
Synthesis..... 132
System Models..... 87
System Simulation..... 88
System Simulation Parameters 23, 89, 90, 94, 97
System Simulation Parameters - Options Tab97
System Simulation Tips..... 135

T

TAN..... 145
TanD..... 257
TANH..... 145
TCF..... 214
TCP..... 218
Temperature..... 179
Term..... 158
Termination..... 158
Terminations..... 35, 179
Text..... 338
Text Model Definitions..... 174
Thick Metal..... 279
Thicknesses..... 257
Thinning Out..... 275, 325
Third Order Intermod Analysis..... 219
Third-order Intercept..... 53
THR..... 157
TIM3P..... 236
TNP..... 237
Toggle Background Color..... 311
Tolerances..... 41, 44
Tone Channel Frequency..... 214
Tone Channel Power..... 218
Top Cover..... 257, 323
Total Node Power..... 237
Total Third Order Intermod Power..... 236
TPL..... 343
TPL file..... 261
Transducer..... 35, 36
Transistor..... 38, 157, 160
Transmission Line..... 39
Transmission lines..... 1
Transmitted Energy..... 119
Triplate..... 257
TWO..... 157
Two-port..... 32, 34, 35, 37
Two-port file..... 158
Two-port S-parameters..... 32, 157

U

Undersampled..... 53
Unilateral..... 179
Unilateral Case..... 37
Unilateral gain circles..... 179
Unnormalized Y parameter data..... 158
Unstable region..... 179
Up-to-date..... 51
Use Krylov Subspace Method..... 44
Use Previous Solution As Starting Point.. 44, 53
Use thinning..... 275
User Functions..... 1, 154
User Model Example..... 167
USING..... 150
Using Equation Results..... 184

Using Non-Default Simulation/Data 183

V

VAIR 148
Value Mode button 311
Variable Values 144
Variables 145, 148, 290
VECTOR 145
Vectors 148
Vendor-supplied models 1
Viahole 261, 264, 316
Viaholes 257, 277, 319
View Menu 303
View Variables 144
Viewer 251, 276, 303, 311, 317, 319, 339
Vnode 181
Voltage 303
VSWR 32
VSWRi 179

W

Waveguide 1
Wavelength 261, 273
Wire/Solid button 311
Write Internal Data Files 337
WSP 343

Y

Y Parameters 179
Yield 183
YINi 179
YM1 36
YM2 36
YMi 179
YOPT 179
Y-parameter data 158
Y-parameters 32, 158, 343
YPij 179

Z

Z Parameters 179
Z-directed 257, 277, 279, 316, 317, 319
Z-Directed Ports 264, 297
ZINi 179
ZM1 36
ZM2 36
ZMi 179
Zo 32
ZOPT 179
ZPij 179