Agilent Technologies

Advanced Design System 2005A

Quick Start

August 2005

Notice

The information contained in this document is subject to change without notice.

Agilent Technologies makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Agilent Technologies shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Warranty

A copy of the specific warranty terms that apply to this software product is available upon request from your Agilent Technologies representative.

Restricted Rights Legend

Use, duplication or disclosure by the U. S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013 for DoD agencies, and subparagraphs (c) (1) and (c) (2) of the Commercial Computer Software Restricted Rights clause at FAR 52.227-19 for other agencies.

395 Page Mill Road, Palo Alto, CA 94304 U.S.A.

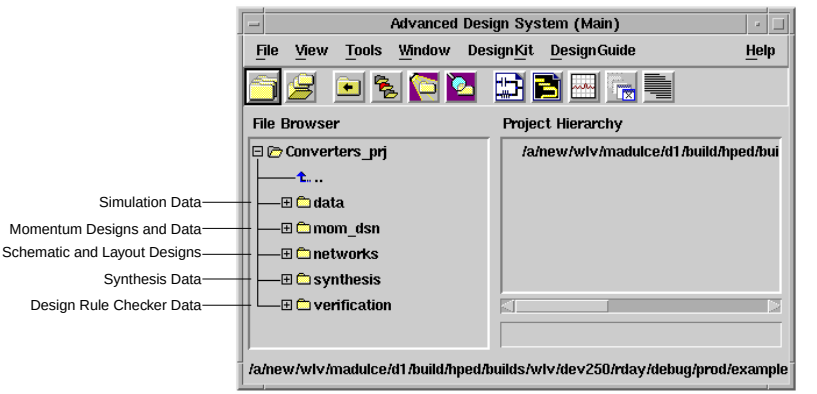
Acknowledgments

Contents

Chapter 1: Projects

Advanced Design System (ADS) uses projects to automatically organize and store data that is generated when you create, simulate, and analyze designs to accomplish your design goals.

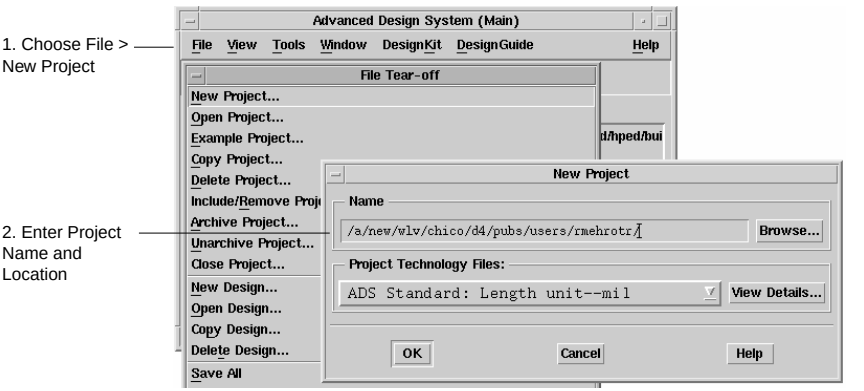
A project includes schematic, layout, simulation, analysis, and output data for the designs that you create, along with any links you add to other designs and projects.



Creating Projects

Use the Main window to create a project that you can then use to organize your designs.

To create a project:



Opening Projects

Use the Main window to open a project. Only one project can be open at a time. When you begin to open a new project, you are prompted to save any changes you have made in a currently open project before it is automatically closed.

To open a project:

1. Choose **File** > **Open Project** and choose a project.

Many example projects are provided in ADS; to open an example project:

1. Choose **File** > **Example Project** and choose a category.
2. Use the *Directories* pane of the *Open Example Project* window to choose a project.

Sharing Projects

Use the Main Window to reuse and share projects without having to manually include all the individual parts that make up a project.

- Add links to create a hierarchical project

  Choose **File** > **Include/Remove Projects** and use the dialog box to locate and link to the project.

- Create a copy to replicate a project

  Choose **File** > **Copy Project** and use the dialog box to locate and copy the project.

- Archive/Unarchive to transfer a compact project archive

  Choose **File** > **Archive Project** and use the dialog box to locate and archive the project.

Chapter 2: Designs

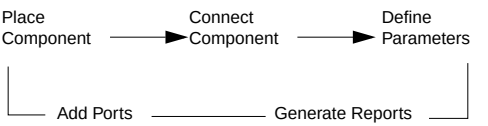
ADS uses designs to store schematic and layout data. A design can consist of a single schematic or layout, or it can consist of a number of schematics and layouts embedded as subnetworks within a single design.

In a design you can:

- Create and modify schematics and layouts
- Add variables and equations
- Place and configure components, shapes, and simulation controllers
- Specify layer and display preferences
- Include annotations using text and illustrations
- Generate layouts from schematics (and schematics from layouts)

Creating Designs

The basic process of creating a schematic or layout is illustrated:



To create a new design:

- From the Main window, choose **Window** > **New Schematic.**
- From a Schematic (Layout) window, choose **File** > **New Design**.

To create a design from an ADS template:

- From a Schematic window, choose **Insert** > **Template** and select a template.

  When you use a template, most of the initial setup and configuration for the schematic, simulation, and data analysis are automatically done.

Opening Designs

You can use the Main window or the Schematic (Layout) window to open a schematic (layout) design.

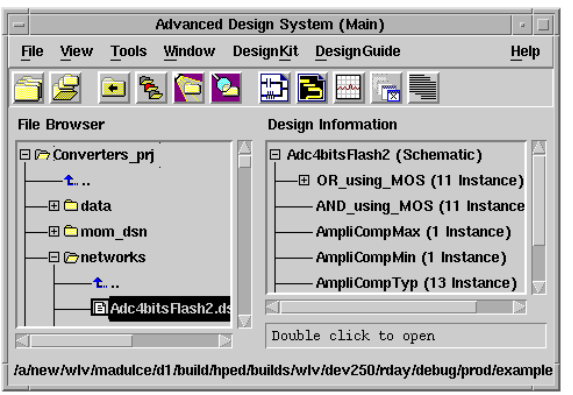
- From the Main window, use the *File Browser* pane to locate and open the schematic (layout).
- From a Schematic (Layout) window, choose **File** > **Open Design** and use the dialog box to locate and open the design.

Design Lists and Design Hierarchies

Designs that you open remain in memory (even after you close all windows) until you explicitly clear them or exit ADS.

To display designs in a project:

1. Double-click the *networks* directory in the Main window to display all designs; double-click a design to list its schematic, layout, and hierarchical information.



To view design hierarchies within a project:

1. Choose **View** > **Design Hierarchies** from the Main window to open the *Design Hierarchies* window.

To view component hierarchy from a Schematic window:

1. Choose **Tools** > **Hierarchy** to open the *Hierarchy* window.

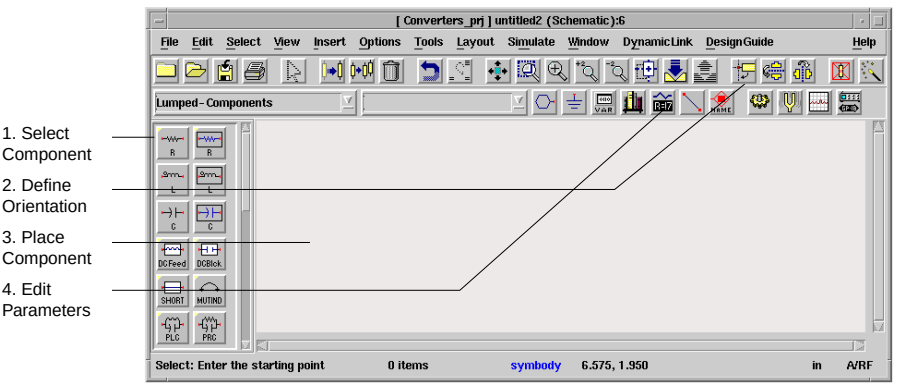
Adding Components

You can place, connect, and configure the following items in the drawing area of your Schematic window to create your design.

- Components
- Data items
- Measurement sources
- Simulation controllers

You can also add entire circuits as subnetworks to create hierarchical designs. Keep in mind that when you begin a design using a template, most of the simulation and analysis setup and configuration is done for you automatically.

To add a component:



Drawing Shapes

To create a layout, you can draw and modify shapes in your Layout window. You can also add traces to represent electrical connectivity.

To draw a shape:

1. Choose **Insert** > **Shape** > (<*shape*>) or click the shape icon on the toolbar.
2. Tips are provided in the status panel; other details are provided here.

   Drawing a Polygon: Enter line segments, and double-click to automatically create the closing segment.

   - *Erasing a Newly Drawn Segment or Arc* To backtrack to the previous point choose **Insert** > **Undo Vertex**.

   Drawing a Polyline: Enter line segments, and double-click to end the final segment.

   - *Erasing a Newly Drawn Segment or Arc* To backtrack to the previous point, choose **Insert** > **Undo Vertex**.

   Drawing a Rectangle: Enter two corners.

   Drawing a Circle: Enter the center point, then a point on the perimeter.

   Drawing an Arc: Enter the point where the arc begins, the center of the arc, and the point where the arc ends. You can draw an arc clockwise or counter-clockwise.

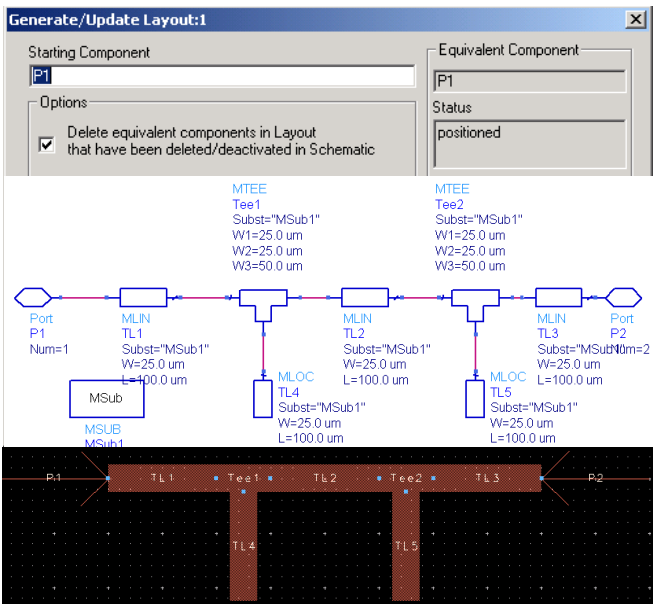
Note Only closed shapes (polygons, circles, rectangles) can be filled; shapes created with polylines (such as arcs) cannot.

Synchronizing Designs

To generate and synchronize your schematic and layout artworks and symbols use *design synchronization*:

- Schematic window > **Layout** > **Generate/Update Layout**
- Layout window > **Schematic** > **Generate/Update Schematic**

The window where you invoke the synchronization operation acts as the source from which the destination representation is generated or updated.



Synchronization Modes

Synchronization can be complete or incremental and can be done to and from a schematic and a layout.

Generate	Update	Place Component
Place all activated components, including those with no artwork, connected to the starting component.	Update a previously generated design by placing components that have been modified.	Place items that have no counterparts in the other representation.
Components with fixed location status are not moved.	Components with fixed location status are not moved.	Use the *Current Rep only* component placement mode.
Components that are not placed in the other representation are highlighted.		*Wire guides* show connectivity in the other representation.
Any component can serve as the starting point for which the location, orientation can be specified.		Use the *Options* > *Variables* command to override the default resolution path for variable and substrate references.

Cross-Probing

To see the layout representation of a specific wire or pin in your schematic, select **Layout > Show Equivalent Node** and click the wire or pin that you would like to see in the layout. When you view the layout, the wire or pin that you selected in the schematic will be highlighted in red.

To see the schematic representation of a specific wire or pin in your layout, select **Schematic > Show Equivalent Node** and click the wire or pin that you would like to see in the schematic. When you view the schematic, the wire or pin that you selected in the layout will be highlighted in red.

Electronic Notebook

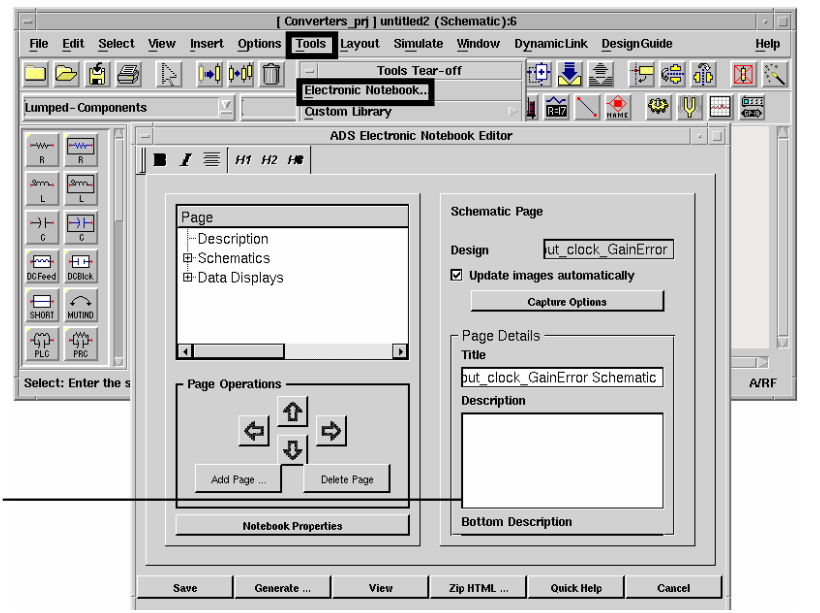
ADS includes an *Electronic Notebook* tool that you can use to generate a portable *notebook* containing:

- Screen captures of schematic and layout designs as well as data display for a given project.
- Text and graphics from other sources.

You can generate html documentation that can be distributed and viewed outside of ADS.

To document designs:

1. Choose Tools > Electronic Notebook



2. Document Design

Design Kits

Design kits, originally developed by popular foundries to provide ADS components and translated model files for distribution by the foundry to the IC designer, can now be developed by anyone.

The ADS design kit user interface provides a standard methodology to package reusable libraries. This library structure can be used for any technology or process to package and distribute a reusable set of components.

From *DesignKit* in the ADS Main window, you can:

- Install a design kit
- Set up the design kit installation at different levels
- List multiple design kits simultaneously
- Define a project and specify a design kit to be used for technology files

The Design Kit Development documentation provides information for first-time design kit developers, troubleshooting legacy design kits, and updating design kits to a new standard.

Chapter 3: Design Analysis

ADS simulation controllers are used to simulate, optimize, test, and analyze designs. Add (and configure) one or more controllers to a design based on the type of design to be simulated and the type of analysis.

A signal processing design simulation requires a Data Flow Controller; an analog/RF design simulation requires one or more of various controllers. An overview of simulation controllers is given in Table 3-1. Details for each controller are given in "Simulation Controllers" on page 4-1.

Optimization and statistical design controllers are used in conjunction with analog/RF and signal processing simulation controllers. An overview of optimization and statistical design controllers is given in Table 3-2. Details for each controller are given in "Optimization and Statistical Design Controllers" on page 4-27.

Design templates (choose *Insert* > *Template* from a Schematic window) are provided that contain controllers appropriate for designs in those templates. When you use a template, most of the initial setup and configuration for the schematic, simulation, and data analysis are done automatically.

Table 3-1. Overview of Simulation Controllers

Simulation Type (ADS Controller Name)	Description	Design Use
Data Flow (DF)	Controls the flow of mixed numeric and timed signals for digital signal processing simulations using the ADS Ptolemy simulator.	All signal processing designs
DC (DC)	Fundamental to all analog/RF simulations. It performs a topology check and an analysis of the DC operating point.	All Analog/RF designs
AC (AC)	Obtains small-signal transfer parameters like voltage gain, current gain, and linear noise voltage and currents.	Filter Amplifier
S-Parameter (S_Param)	Provides linear S-parameter, linear noise parameters, transimpedance, and transadmittance. Can be used to achieve many goals of the AC simulator.	Filter Oscillator Amplifier
Harmonic Balance (HarmonicBalance)	Uses nonlinear harmonic-balance techniques to find the steady-state solution in the frequency domain.	Mixer Oscillator Power amplifier Transceiver
Circuit Envelope (Envelope)	Uses a combination of frequency- and time-domain analysis techniques to yield a fast and complete analysis of complex signals such as digitally modulated RF signals.	Mixer Oscillator Power amplifier Transceiver Phase-locked loop
Large-Signal S-Parameter (LSSP)	Performs large-signal S-parameter analyses to represent nonlinear behavior. The accompanying P2D simulator can be used to speed up subsequent analyses.	Power amplifier

Table 3-1. Overview of Simulation Controllers (continued)

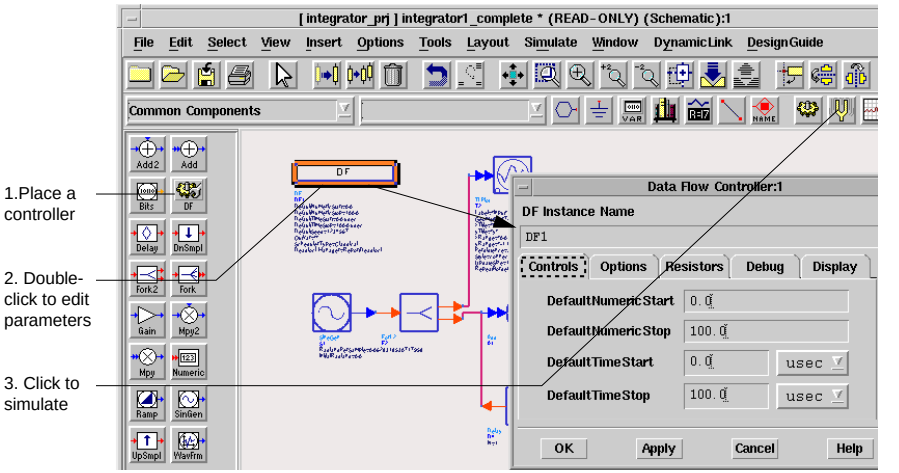
Simulation Type (ADS Controller Name)	Description	Design Use
Gain Compression (XDB)	Seeks a user-defined gain-compression point at which an actual power curve deviates from an idealized linear power curve.	Power amplifier Mixer
Transient/Convolution (Tran)	Solves a nonlinear circuit entirely in the time domain using simplified models to account for the frequency-dependent behavior of distributed elements.	Mixer Power amplifier Switching circuits
RF Budget (Budget)	Determines the linear and nonlinear characteristics of an RF system made up of a cascade of 2-port, 2-pin linear or nonlinear components.	Mixer Nonlinear amplifier

Table 3-2. Tuning, Optimization, and Statistical Design Controllers

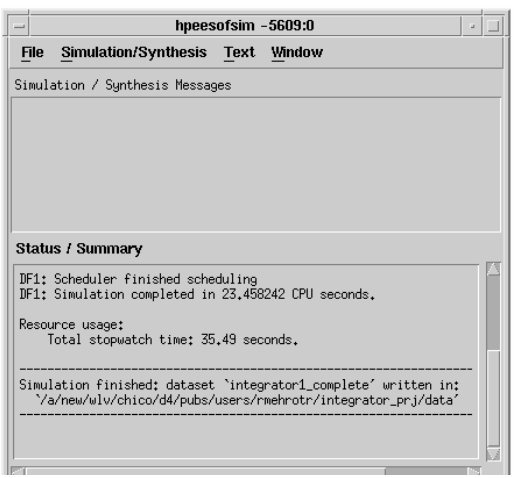
Controller Type (ADS Controller Name)	Description	Used With
Nominal Optimization (Optim)	Used to compare calculated and desired responses and modify parameter nominal values to bring the calculated response closer to the desired optimization goals.	A Goal component must also be used to specify the optimization goals.
Monte Carlo (MonteCarlo)	Uses the Monte Carlo method to simulate a design over a given number of trials in which the statistical variables have values that vary randomly about their nominal values with specified probability distribution functions.	A YieldSpec component can optionally be used to specify the desired yields. A StatCorr component can optionally be used to specify statistical correlation between statistical design variables.
Yield Analysis (Yield)	Uses the Monte Carlo method to determine the manufacturing yield. For each trial, the calculated response is compared to the coresponding yield specification, and a pass/fail decision is made.	A YieldSpec component must also be used to specify the acceptable performance. A StatCorr component can optionally be used to specify statistical correlation between statistical design variables.
Yield Optimization (YieldOptim)	Used to analyze multiple yield analyses and adjust the nominal values to maximize the yield estimate of the statistical design variables.	A YieldSpec component must also be used to specify the acceptable performance.
Design of Experiments (DOE)	Used to sequentially and iteratively improve the statistical performance of a design by identifying variables that contribute significantly to performance variation and honing in on the target statistical response.	A DOEGoal component must also be used to specify the desired goals.

Basic Simulation

To simulate a design:



The status of the simulation will be displayed in a message window.



Signal Processing Simulation

The ADS Ptolemy simulator is used for signal processing and mixed-signal cosimulation. The ADS signal processing environment features:

- Accurate RF system models for faster development of system specifications.
- Extensive behavioral model set for RF and DSP system modeling that helps engineers rapidly create and optimize larger designs.
- Co-design between DSP, analog and RF portions of the signal path.
- Hundreds of DSP and analog models for development of algorithms.
- Propagation and matrix models for modeling complete wireless systems.
- Data export/import to/from measurement instrumentation helps verify designs using virtual prototyping concepts.
- IP reuse of MATLAB, HDL, and C++ models.

Systems designer can architect a communications system using behavioral models to validate a concept. A designer can then design and substitute lower levels of abstraction to verify the RF/mixed-signal design down to the circuit level and export the design to a variety of manufacturing tools. Available statistical design capabilities enable the user to make difficult tradeoffs during the design process in order to optimize performance or manufacturing yield.

A large array of behavioral RF/analog/DSP models work with the ADS Ptolemy simulator to provide leading-edge simulation accuracy during the design process. Propagation and matrix models facilitate modeling of the complete wireless system. ADS communications library modules support the latest communications standards (3GPP, DTV, WLAN, and others). These libraries can be used at the front-end of the design process when the system architecture is conceptualized, during the design and implementation process, or at the back-end of the design process during final verification.

Connections to Agilent Technologies test and measurement instruments provide virtual prototyping verification for designs prior to final implementation or tape out. For example, a new analog/RF/DSP transmitter design modeled in the DSP schematic can be verified by connecting the simulation output with an Agilent ESG signal generator. The resulting real-world signal produced in a virtual environment will include all signal distortions, noise, and propagation effects modeled into the design. This signal can then be fed into an Agilent signal analysis component or

real-world receiver circuit to provide virtual prototyping, and design tuning using real-world hardware and analysis.

ADS Ptolemy simulation requires a Data Flow Simulation Controller and source or sink components to control the duration of the simulation.

- Sources are components with no inputs that read data from files, instruments, and data sets. When a source controls the simulation, it will keep the simulation running long enough to output all data.
- Sinks are components with no outputs. When a sink controls the simulation, it will keep the simulation running long enough to satisfy its start and stop times. When a sink is not controlling the simulation, it will begin collecting data at *start*, then collect as much data as the simulation produces.

There are two basic types of DSP components: *timed* components have a notion of sampling rate, carrier frequency, and envelope; *numeric* components process integers, matrixes, floats, fixed-point numbers, and model the DSP portions of a design.

For more information on:

- Cosimulation with analog/RF designs, refer to the Cosimulation section of the ADS Ptolemy Simulation documentation.
- Connecting to instruments, refer to the Connection Manager documentation.
- Cosimulation with MATLAB IP import, refer to the MATLAB Cosimulation documentation.
- Cosimulation and HDL IP import, refer to the HDL Cosimulation documentation.
- C++ IP import, refer to the Model Builder documentation.

Analog/RF Simulation

Analog/RF simulation calculates the response of a circuit to a particular stimulus by formulating a system of circuit equations and then solving them numerically. Each simulation technology accomplishes this analysis as follows.

DC Analysis

- Solves a system of nonlinear ordinary differential equations
- Solves for an equilibrium point
- All time-derivatives are constant (zero)
- System of nonlinear algebraic equations

Transient Analysis

- Solves a system of nonlinear ordinary differential equations
- Time-derivatives replaced with a finite-difference approximation (integration method)
- Sequence of systems of nonlinear algebraic equations (one system at each timepoint)

Harmonic Balance

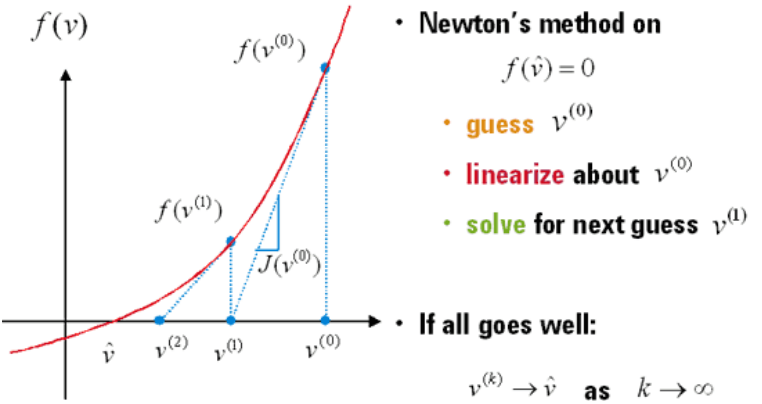
- Solves a system of nonlinear ordinary differential equations
- Steady-state method
- Solution approximated by truncated Fourier series
- System of nonlinear ordinary differential equations becomes a system of nonlinear algebraic equations in the frequency domain

Solving Nonlinear Algebraic Equations

Nonlinear algebraic equations are solved using the Newton-Raphson algorithm (Newton's method) as follows.

- Convert the problem to a sequence of systems of linear equations

- Quadratic convergence near the solution (error squared at each iteration)



S-Parameter Test Lab

An S-parameter test lab enables you to calculate the S-parameters of multiple N-port networks in a single simulation run.

An S-parameter test lab is a schematic that contains one S-parameter test lab component and one or more test benches. A test bench is a schematic that contains an N-port network and terminations for each port of the network. Its use is best illustrated in multiple stage circuit designs where viewing the inter-stage circuit behavior of all stages simultaneously is desired. In such situations the S-parameter test lab can be used to terminate each stage in the applicable input/output impedances of adjacent stages rather than in the standard 50 ohms.

RefNets can also be used in conjunction with the S-parameter test lab feature.

Design Sequencer

A Design Sequencer controller enables you to sequence multiple simulations in a single simulation run using a test bench that includes simulation controllers and the top-level design file.

Typical applications for a Sequencer controller are:

- Optimizing a variable across multiple simulations
- Enabling complex instrument control in Ptolemy
- Running a series of verifications tests on a design

Table 3-3. Differences Between S-parameter Test Labs and Sequencer

Sequencer	Test Lab
DC, SP, AC, HB, Tran, ENV, Ptolemy	SP only
Uses Test Bench Controllers	Uses Test Lab Controller
Different temps per test bench possible	One simulation temp for all
Opt/Stat/ParamSwp at top level	
RefNets supported	

RefNets

A RefNet (reference network) component enables the port impedance from another design file in the system (the referenced network) to be referenced as a terminating impedance for the current design file under test.

RefNet components *RefNetTB* and *RefNetDesign* have the same functionality and are supported under DC, AC and S-parameter analysis, with these differences:

- RefNetTB supports nested network referencing while RefNetDesign does not.
- RefNetTB uses a test bench as the reference design while RefNetDesign uses a standard (non-test-bench) schematic design.

Typical RefNet applications are:

- **Inter-stage circuit analysis and design** In some design applications it is desirable to simultaneously evaluate the performance of individual circuit stages terminated in the input and output impedances of adjacent stages. To accomplish the termination of an individual stage referenced to a specific port of other stages in the design chain, the RefNet is used in the S-parameter test lab.
- **Design-specific termination** For some top-level DC, AC, or S-parameter design files, it may be necessary to terminate a port whose impedance is characterized by data, from an external file (S-, Z-, or Y-parameters, for example) or some other network.

Common Circuit Simulation Methods

Backward Euler

- First-order method that assumes the solution waveform is linear over one time step
- One-step method (needs one previous time point solution only)
- Adapts faster to abrupt signal changes
- Stable on all stable differential equations and some unstable ones
- Exhibits heavy numerical damping, increases loss
- Require smaller time step to maintain accuracy

Trapezoidal Rule

- Second-order method, assumes the solution waveform is quadratic over one time step
- One-step method
- May exhibit point-to-point ringing on circuits that have very small time constant comparing to time step (stiff circuit)
- Stable only on stable differential equations
- Exhibits no artificial numerical damping

Backward Difference Formulas (Gear's methods)

- Multiple order polynomial over one time step
- Only the first six orders are available in ADS
- First-order method is identical to backward Euler
- Higher-order polynomials allow a larger time step without sacrificing accuracy, are efficient for smooth waveforms
- Higher-order methods (order > 2) may exhibit stability problems on lightly damped circuits
- Second-order backward difference formula (Gear 2)
- Two-step method

- Stable on all stable differential equations and some unstable ones.
- Exhibit some numerical damping

Truncation Error

- The error made by replacing the time derivatives with a discrete-time approximation. This error is difficult to estimate and depends on the type of circuits and the time steps.

Local Truncation Error (LTE)

- The truncation error made on a single step

Global Truncation Error (GTE)

- Maximum accumulated truncation error
- The circuit with long time constant is sensitive to these errors
- Logic and bias circuits are not sensitive to these errors

Convergence Criteria

Newton's iteration is converged if the approximate solution first satisfies the residue criteria at the end of each Newton iteration and the Update criteria once the residue criteria is satisfied.

- **Residue criteria** KCL satisfied to a given tolerance. This is enforced at each node and is important when impedance at a node is small.
- **Update criteria** Difference between the last two iterations must be small. This is important when impedance at a node is large.

Continuation Methods

Continuation methods provide a sequence of initial guesses that are sufficiently close to the solution to ensure Newton's method convergence.

- Choose a natural or contrived continuation parameter that controls a modification of the circuit
- Step the continuation parameter from 0 to 1 (the original circuit configuration), using the solution from the previous step as the starting point

As long as the solution changes continuously as a function of the continuation parameter and the steps are small enough, Newton's method will converge. Keep in mind that source and gmin stepping will fail if the continuation path contains a limit point.

- **Source stepping** Uses a fraction of the source voltages and currents applied to the circuit as the continuation parameter.
  - Turn off all sources when the continuation parameter equals 0
  - Raise source levels to their final levels slowly, generating a sequence of circuit configurations
  - Use the solution from the previous configuration as an initial guess for the current configuration
- **Gmin stepping** Uses the continuation parameter to control the value of the gmin resistors.
  - Start with a large gmin for an easy-to-calculate solution because nonlinear device behavior is muted by the presence of small resistors
  - End with very small gmins for large resistors that no longer affect the circuit
  - Remove the gmins to calculate the final solution
- **Arc-length continuation** Works best for complicated continuation paths and limit points using a continuation parameter that is a function of the arc-length parameter.
  - Travel same distance at each step, as specified by the arc-length
  - Increase or decrease the continuation parameter along the path in each step

Preventing Convergence Problems

Convergence problems typically arise as a result of errors in circuit connectivity or unreasonable (out of range) model or component values. To avoid convergence problems, you can:

- Turn on the topology checker
- Turn on warnings
- Act on messages in the ADS *Status Server* window
- Eliminate small floating resistors (or increase I_AbsTol) because any error in calculated voltages for nodes with small resistors results in large error currents
- Avoid very large and very small resistances connected to a node because large resistances are lost during Jacobian construction due to numerical round-offs

Analog/RF Smart Simulation Wizard

ADS also provides an interface for circuit simulation. The *Smart Simulation Wizard* can be used to:

- Create Analog/RF designs
- Set up and run simulations
- Display simulation results

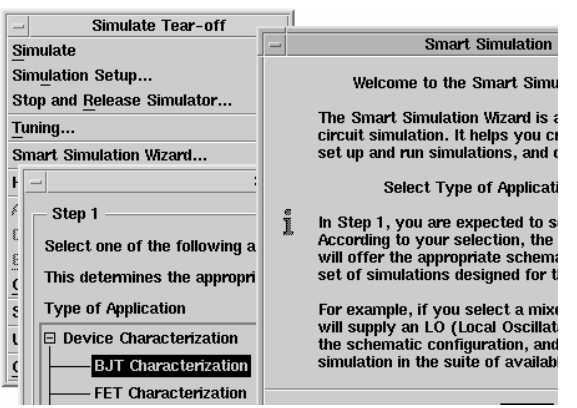
To *smart* simulate a design:

1. Choose Simulate > Smart Simulation Wizard

2. Specify Circuit Configurations

3. Specify Simulation Options

4. Display Results



Instrument Connectivity

Connection Manager enables the sharing of signals, measurements, algorithms, and data between ADS simulations and Agilent instruments (signal generators and analyzers). Using Connection Manager, you can:

- Access and control instruments from ADS dialogs
- Measure devices and construct ADS data sets from measurement data
- Create simulation models based on measured data
- Use real-time instrument-generated stimulus and measurement during simulation

Momentum Simulation, Optimization, and Visualization

Momentum includes simulation, optimization, and visualization tools for predicting the performance of multilayer high-frequency circuit boards, antennas, hybrids, multichip modules, and integrated circuits.

- **Momentum** is an electromagnetic simulator that calculates S-parameters for general planar circuits, including microstrip, slotline, stripline, coplanar waveguide, and other topologies.
- **Momentum Optimization** varies geometry parameters automatically to help you achieve the optimal structure that meets circuit or device performance goals.
- **Momentum Visualization** provides a 3D perspective of simulation results, enabling you to view and animate current flow in conductors and slots, and view both 2D and 3D representations of far-field radiation patterns.

Using Momentum, you can:

- Simulate when a circuit model range is exceeded or the model does not exist
- Identify parasitic coupling between components
- Go beyond simple analysis and verification to design automation of circuit performance
- Visualize current flow and 3D displays of far-field radiation

Chapter 4: Simulation and Optimization Controllers

ADS simulation controllers are used to simulate, optimize, and test designs. Add (and configure) one or more controllers to a design based on the type of design to be simulated and the type of analysis.

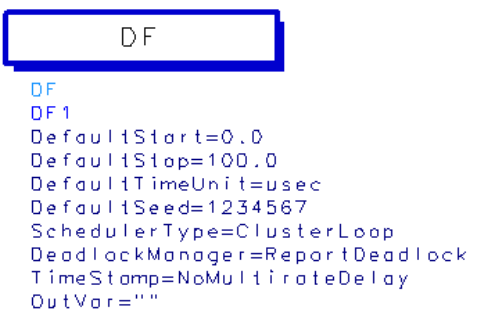
Optimization and statistical design controllers are used in conjunction with analog/RF and signal processing simulation controllers.

Simulation Controllers

A signal processing design simulation requires a Data Flow Controller; an analog/RF design simulation requires one or more of various controllers. Each ADS controller is described in the following sections.

Data Flow Simulation Controller

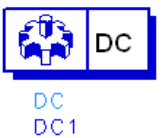
The Data Flow (DF) controller is used to control the flow of mixed numeric and timed signals for all digital signal processing simulations.



Important Beginning with ADS 1.5, multiple DF controllers on the schematic are not allowed. Multiple controllers were used to simulate the same design with different controller parameters; you can achieve the same effect by using single-point sweeps on the parameter you are interested in varying.

DC Simulation Controller

The DC controller is used for single-point and swept simulations. Swept variables can be related to voltage or current source values, or to other component parameter values. By performing a DC swept bias or a swept variable simulation, you can check the operating point of the circuit against a swept parameter such as temperature or bias supply voltage.



Use the DC controller to:

- Verify the proper DC operating characteristics of the design under test.
- Determine the power consumption of your circuit.
- Verify model parameters by comparing the DC transfer characteristics (I-V curves) of the model with actual measurements.
- Display voltages and currents after a simulation.

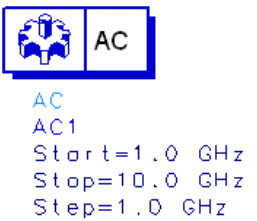
A DC simulation is the first analysis for most other analyses. It uses a system of nonlinear ordinary differential equations (ODEs) to solve for an equilibrium point in the linear/nonlinear algebraic equations that describe a circuit once:

- Independent sources are constant valued
- Capacitors and similar items are replaced with open circuits
- Inductors and similar items are replaced with short circuits
- Time-derivatives are constant (zero)

Linear elements are replaced by their conductance at zero frequency.

AC Simulation Controller

A linear AC analysis is a small-signal analysis. For this analysis the DC operating point is found first and then the nonlinear devices are linearized around that operating point. Small-signal AC simulation is also performed before a harmonic-balance (spectral) simulation to generate an initial guess at the final solution.



Use the AC controller to:

- Perform a swept-frequency or swept-variable small-signal linear A simulation.
- Obtain small-signal transfer parameters, such as voltage gain, current gain, transimpedance, transadmittance, and linear noise.

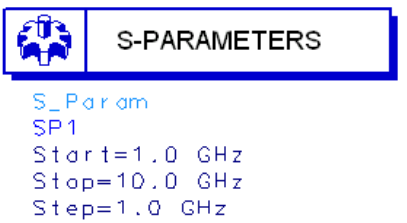
An AC simulation also offers a linear noise simulation option that can include the following noise contributions in its simulation:

- Temperature-dependent thermal noise from lossy passive elements, including those specified by data files.
- Temperature and bias-dependent noise from nonlinear devices.
- Noise from linear active devices specified by two-port data files that include noise parameters.
- Noise from noise source elements.

The noise simulation calculates the noise generated by each element, and then determines how that noise affects the noise properties of the network.

S-Parameter Simulation Controller

The S-parameter (S_Param) controller is used to define the signal-wave response of an n-port electrical element at a given frequency. It is a type of small-signal AC simulation that is commonly used to characterize a passive RF component and establish the small-signal characteristics of a device at a specific bias and temperature.



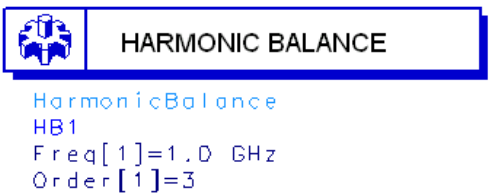
Use the S-Parameter controller to:

- Obtain the scattering parameters (S-parameters) of a component or circuit, and convert the parameters to Y- or Z-parameters.
- Plot, for example, the variations in swept-frequency S-parameters with respect to another changing variable.
- Simulate group delay.
- Simulate linear noise.
- Simulate the effects of frequency conversion on small-signal
- S-parameters in a circuit using a mixer.

S-parameter simulation normally considers only the source frequency in a noise analysis. Use the Enable AC Frequency Conversion option if you also want to consider the frequency from a mixer's upper or lower sideband.

Harmonic Balance Simulation Controller

The harmonic balance (HarmonicBalance) controller is used for simulating analog RF and microwave circuits.



Within the context of high-frequency circuit and system simulation, harmonic balance offers benefits over conventional time-domain transient analysis:

- Harmonic balance captures the steady-state spectral response directly while conventional transient methods must integrate over many periods of the lowest-frequency sinusoid to reach steady state.
- Harmonic balance is faster at solving typical high-frequency problems that transient analysis cannot accurately solve or can solve at prohibitive costs.
- Harmonic balance is more accurate at solving high frequencies where many linear models are best represented in the frequency domain.

Use the HarmonicBalance controller to:

- Determine the spectral content of voltages or currents.
- Calculate quantities such as third-order intercept points, total harmonic distortion, and intermodulation distortion components.
- Perform power amplifier load-pull contour analyses.
- Perform nonlinear noise analysis.

Harmonic balance is a frequency-domain analysis technique for simulating distortion in nonlinear circuits and systems. It obtains the frequency-domain voltages and currents to calculate the spectral content of voltages or currents in the circuit. The harmonic balance method is iterative; it is based on the assumption that for a given sinusoidal excitation there exists a steady-state solution that can be approximated to satisfactory accuracy by means of a finite Fourier series.

The Harmonic Balance solution is approximated by truncated Fourier series and this method is inherently incapable of representing transient behavior. The

time-derivative can be calculated exactly with boundary conditions, v(0)=v(t), automatically satisfied for all iterates.

The truncated Fourier approximation + N circuit equations results in a residual function that is minimized.

N x M nonlinear algebraic equations are solved for the Fourier coefficients using Newton's method and the inner linear problem is solved by:

- Direct method (Gaussian elimination) for small problems
- Krylov-subspace method (e.g. GMRES) for larger problems

Nonlinear devices (transistors, diodes, etc.) in Harmonic Balance are evaluated (sampled) in the time-domain and converted to frequency-domain via the FFT.

Convergence

Nonconvergence is a numerical problem encountered by the harmonic balance simulator when it cannot reach a solution within a given tolerance, after a given number of numerical iterations. While there is no one specific solution for solving convergence problems, consider the following guidelines:

- Increase the Order (or other harmonic controls); this is the most basic technique for solving convergence problems if the time penalty for doing so is acceptable.
- Use the Status server window as the main tool in solving convergence problems (set StatusLevel=4). For each Newton iteration the L-1 norm of the residuals throughout the circuit is printed: * indicates a full Newton step (vs. a Samanskii step).
- Convergence criteria are controlled by Voltage relative tolerance, and Current relative tolerance (in the Options component, under the Convergence tab). In general, convergence speed is improved by increasing these values, but at the expense of accuracy. Similarly, the smaller these values are, the more accurate the results but the slower the convergence.
- Newton convergence issues with Krylov methods (linear problem solutions can only approximate) can be improved by using better preconditioners.
- Set the Oversample parameter to a value greater than 1.0, such as 2.0 or 4.0.

  While this can often solve convergence problems it does so at the cost of computer memory and simulation time. For multiple-tone harmonic balance simulations, assign the largest signal in the circuit to Freq[1]; the simulator's

FFT algorithm is set up so that aliasing errors are much less likely to affect Freq[1] than any other tone.

- When using a direct linear solver, blocks of the Harmonic Balance Jacobian inherit the Jacobian matrix ordering from the DC solution process. This matrix ordering can greatly affect the efficiency of the Harmonic Balance Jacobian factorization and show noticeable simulation slowdown in some circuits. To avoid this, use a DC convergence mode that has not changed, e.g. DC_ConvMode=3.
- For non-convergence due to tight tolerances, monitor the residuals in the Status Server window.
  - Increase I_AbsTol if the circuit is converging to within a few pA but not quite to I_AbsTol=1pA
  - Increase I_RelTol if the problem is with nodes associated with large currents
  - Increase I_AbsTol if the small current nodes are the issue
  - Relax voltage tolerances for failure in the Newton update criterion
- The ADS Analog RF Simulator (ADSsim) runs from a netlist. ADS writes a netlist file (*netlist.log*) before invoking ADSsim. The order of components and model definitions in the netlist determine the initial Jacobian matrix ordering. This matrix ordering can affect the efficiency of the Jacobian factorization and cause either a simulation slow down or non-convergence.
- For convergence problems due to errors in the component model equations (incorrect derivatives, etc.) make sure ancient Berkeley MOSFET Level 1, 2, 3 are not the culprit and that the latest model version is used (especially BSIM3 models). Model problems can cause the Newton residual to hit a threshold (greater than the convergence criteria tolerances) and stale the convergence process or even exhibit random jumps (sudden increase in value). Set the device's Xqc parameter to a nonzero value to allow the simulator to use a charge-based model for the gate capacitance. This often enables convergence, but at the cost of extracting an extra SPICE model parameter.

Sweeps as Convergence Tools

Continuation methods provide a sequence of initial guesses that are sufficiently close to the solution to ensure Newton's method convergence in Harmonic Balance. Sweeps can be used to formulate a specialized continuation method for a particular circuit problem.

Sweep a circuit element that, when set to some different value, makes the circuit more linear. For instance, in an amplifier circuit there may be a resistor that can be used to lower the amplifier's gain. The simulator may be able to find a solution to the circuit under a low-gain condition. Then, if the component's value is swept toward the desired value, the simulator may be able to find a final solution. Start with a value that works, and stop with the desired value. Also, select Restart, under the Params tab. Usually, a better initial guess at each step helps the simulator to converge.

The main ways to perform sweeps are:

- HB sweep within the HB controller; for all sweeps except frequency.
- Parameter sweep using a separate sweep controller.

Convergence and Samanskii Steps

The Samanskii steps can significantly speed up the solution process. However, using an approximate Jacobian, particularly for a larger number of iterations, may result in poor or even no convergence. The constant is used in two ways: first, it becomes a more absolute measure when it is smaller; then, it approaches the requirement that each iteration reduces the relevant norm by one-third.

Decreasing the Samanskii constant beyond a certain point (which, in turn, depends on the quality of the most recent Newton step) will not make a difference; however, setting the Samanskii constant to zero will effectively disable any Samanskii steps altogether.

Increasing the Samanskii constant relaxes this requirements in general, but the condition becomes more dependent on the quality of the standard most recent Newton iteration. In other words, a more rapid convergence of the Newton step would also require better convergence of the Samanskii steps.

Convergence and Arc-Length Continuation

Arc-length continuation is an extremely robust algorithm. If it fails, try all other convergence remedies first before adjusting arc-length parameters

- MaxStepRatio controls the maximum number of continuation steps (default 100)

- MaxShrinkage controls the minimum size of the arc-length step (default 1e-5)
- ArcMaxStep limits the maximum size of the arc-length step (default is 0, i.e. no limiting)
- ArcMinValue and ArcMaxValue define the allowed range for the variation of the continuation parameter

Circuit Envelope Simulation Controller

The Circuit Envelope (Envelope) controller is best suited for a fast and complete analysis of complex signals such as digitally modulated RF signals. It combines features of time and frequency-domain representation by permitting input waveforms to be represented in the frequency domain as RF carriers, with modulation *envelopes* that are represented in the time domain.

Circuit envelope is highly efficient in analyzing circuits with digitally modulated signals, because the transient simulation takes place only around the carrier and its harmonics. In addition, its calculations are not made where the spectrum is empty.

- It is faster than harmonic balance, for a given complex signal Spice, assuming most of the frequency spectrum is empty
- It does not compromise in signal complexity, unlike time-varying HB or Shooting Method Component accuracy, unlike Spice, Shooting Method, or DSP
- It adds physical analog/RF performance to DSP/system simulation with real-time co-simulation with ADS Ptolemy
- It is integrated in same design environment as RF, Spice, DSP, electromagnetic, instrument links, and physical design tools

Advantages over Harmonic Balance

- In harmonic balance, if you add nodes or more spectral frequencies, RAM and CPU requirements increase geometrically. Krylov improved this, but it's still a limitation of harmonic balance because the signals are inherently periodic.
- Conversely, the penalty for more spectral density in circuit envelope is linear: just add more time points by increasing TSTOP. The longer you simulate, the finer your resolution bandwidth.

- Doing a large number of simple 1-tone HB simulations is effectively faster and less RAM intensive than one huge HB simulation.
- With a circuit envelope simulation the amplitude and phase at each spectral frequency can vary with time, so the signal representing the harmonic is no longer limited to a constant, as it is with harmonic balance.

Limitations

1. More occupied spectrum than unoccupied spectrum.

   You're carrying more overhead with frequency-domain assumptions and harmonics than necessary. Use SPICE.

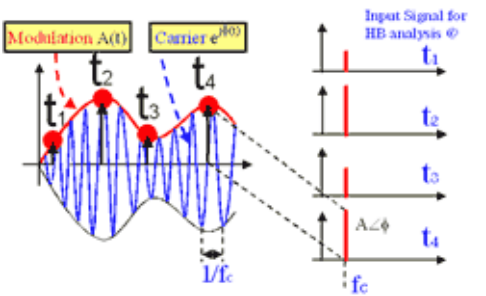
2. Everything baseband. Depends.
   - If everything linear, use AC/S-parameter (for noise or budget)
   - If everything nonlinear or digital, use SPICE.
   - If everything logic/behavioral, use PTOLEMY.

3. Occupied spectrum is relatively sparse.

   If you can do what you want using Harmonic Balance, you should. Post-processing, optimization, and yield are simpler and faster.

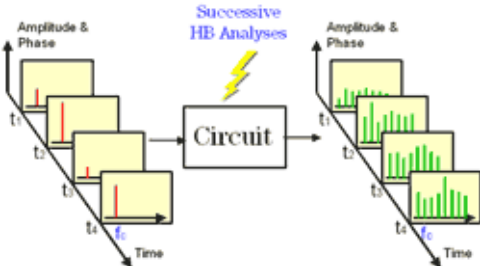
Simulation Process

1. Transform input signal



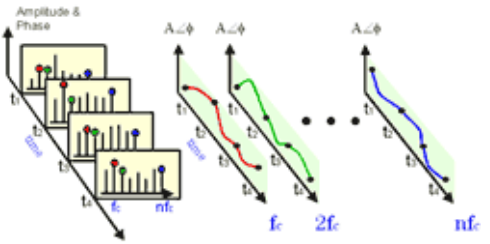
Each modulated signal can be represented as a carrier modulated by an envelope - A(t)*ejf(t). The values of amplitude and phase of the sampled envelope are used as input signals for Harmonic Balance analyses.

2. Frequency Domain Analysis



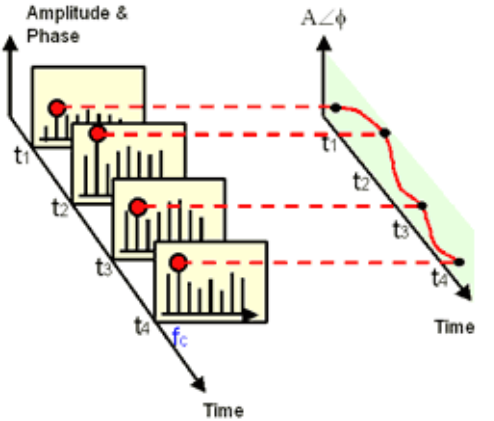
Harmonic Balance analysis is performed at each time step. This process creates a succession of spectra that characterize the response of the circuit at the different time steps.

3. Time Domain Analysis



Circuit envelope provides a complete non steady-state solution of the circuit through a Fourier series with time-varying coefficients.

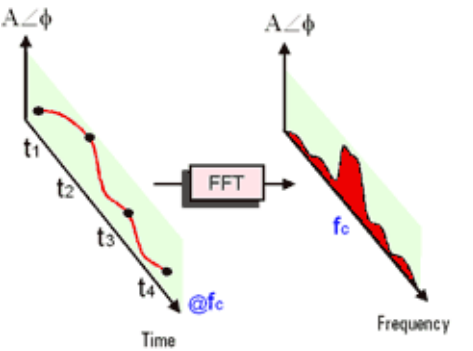
4. Extract Data from Time Domain



Selecting the desired harmonic spectral line (fc in this case), it is possible to analyze:

- Amplitude vs. Time (Oscillator start up, Pulsed RF response, AGC transients)
- Phase (f) vs. Time (t) (VCO instantaneous frequency (df/dt), PLL lock time)
- Amplitude & Phase vs. Time (Constellation plots, EVM, BER)

5. Extract Data from Frequency Domain



By applying FFT to the selected time-varying spectral line it is possible to analyze:

- Adjacent channel power ratio (ACPR)
- Noise power ratio (NPR)

- Power added efficiency
- Reference frequency feedthrough in PLL
- Higher order intermods (3rd, 5th, 7th, 9th)

Simulation Steps

1. Define baseband signal modulation
   - Predefined sources
   - Equations
   - I and Q data vs. time data from DSP simulation
2. Define RF carrier frequencies, time step and duration of the simulation
3. Calculate time-varying Fourier coefficients
4. Post-process and display results

or

1. Define input signals with modulation such as amplitude, phase, frequency, I/Q.
2. Define the time step
3. Simulator calculates Fourier coefficients versus time
4. Fourier transforms are calculated to display frequency spectrum around any tone (if necessary)

Typical Analyses

- Intermodulation distortion.
- Amplifier spectral regrowth and adjacent channel power leakage.
- Oscillator turn-on transients and frequency output versus time in response to a transient control voltage.
- PLL transient responses.
- AGC and ALC transient responses.
- Circuit effects on signals having transient amplitude, phase, or frequency modulation.
- Amplifier harmonics in the time domain.

- Subsystems using modulation signals such as multilevel FSK, CDMA, or TDMA.
- Third-order-intercept and higher-order intercept analyses of amplifiers and mixers.
- Time-domain optimization of transient responses.

Typical Applications

Time Domain Data Extraction

Selecting the desired harmonic spectral line it is possible to analyze:

- Amplitude vs. Time
  - Oscillator startup
  - Pulsed RF response
  - AGC transients
- Phase vs. Time
  - VCO instantaneous frequency, PLL lock time
- Amplitude and phase vs. time
  - Constellation plots
  - EVM, BER

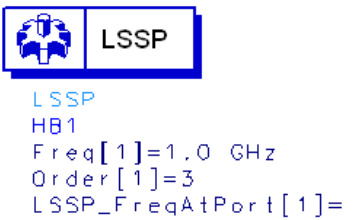
Frequency Domain Data Extraction

By applying FFT to the selected time-varying spectral line it is possible to analyze:

- Adjacent channel power ratio (ACPR)
- Noise power ratio (NPR)
- Power added efficiency (PAE)
- Reference frequency feedthrough in PLL
- Higher order intermods (3rd, 5th, 7th, 9th)

LSSP Simulation Controller

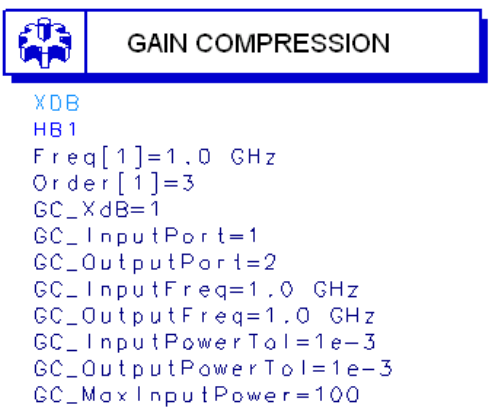
The large-signal S-parameter (LSSP) controller facilitates calculation of large-signal S-parameters in nonlinear circuits.



Large-signal S-parameters are based on a harmonic balance simulation of the full nonlinear circuit. Unlike S-parameters, large signal S-parameters can change as power levels are varied because the harmonic balance simulation includes nonlinear effects such as compression.

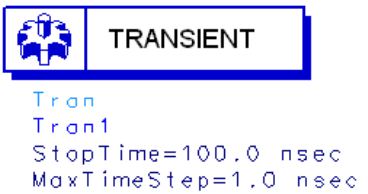
Gain Compression Simulation Controller

The gain compression (XDB) controller calculates the gain compression point of an amplifier or mixer. It sweeps the input power upward from a small value, stopping when the required amount of gain compression is seen at the output.



Transient/Convolution Simulation Controller

The transient/convolution (Tran) controller solves a set of integro-differential equations that express the time dependence of circuit currents and voltages. The result of such an analysis is nonlinear with regard to time and, possibly, a swept variable.



Use the transient/convolution controller to perform:

- SPICE-type transient time-domain analysis.
- Nonlinear transient analysis on circuits that include frequency-dependent loss and dispersion effects of linear models, or convolution analysis.

A transient analysis is performed entirely in the time-domain. It does not account for the frequency-dependent behavior of distributed elements.

A convolution analysis represents distributed elements in the frequency domain to account for their frequency-dependent behavior.

Transient Simulation and Convergence

In transient analysis a numerical integration algorithm is used at each time point to approximate the differential equations into algebraic equations. Integration methods are used to replace the time derivative with a discrete-time approximation.

Time Step Control Characteristics

Local Truncation Error

- Estimates the LTE made on every capacitor and inductor
- Determines the time step size to ensure the largest LTE remains within the accepted tolerance
- The estimated LTE is inversely proportional to TruncTol

- The accepted tolerance depends on the relative and truncation tolerances set for curent and voltage. It is proportional to I_RelTol × TruncTol and V_RelTol × TruncTol

Iteration-Count

- Determines the timestep size based on the number of Newton iterations required for the previous time point
- No direct relationship between iterations and LTE
- Effectively controlled by Max timestep (for linear circuits)

Fixed

- The timestep is fixed and equal to Max timestep

Break Points

- Generated by built-in independent sources whenever an abrupt change in slop occurs
- Ensure that corners in waveforms are not missed
- ADS always places time points on a break point (except fixed timestep)
- Backward Euler is used on time points that are the first timestep after break points
- The step size is reduced when time point is close to a break point

Transient Convergence Tips

- For initial transient analysis, use I_RelTol = V_RelTol = 1e-3; tighten these values only when higher accuracy is needed. Simulation will run much faster with these setting compared to 1e-6.
- Transient analysis convergence problems are often caused by jumps in the solution. This typically occurs in circuits with overly simplified models that exhibit positive feedback, or when the circuit contains nodes that do not have a capacitive path to ground. Add a small capacitor from the troublesome node to ground and give a complete capacitance model when specifying the nonlinear device model parameters.
- Analog circuits are generally sensitive to truncation error due to their relative long time constants. Use LTE timestep control to ensure accuracy of the results.

- Backward Euler (Gear1 or Mu=0 in Trapezoidal) and Gear2 are stable for all stable and some unstable differential equations. However, trapezoidal rule is stable only on stable differential equations. Switch to Gear1 or Gear2 when trapezoidal rule fails on unstable differential equations.

Typical Convergence Problems

Capacitor model problems

- Use simplified device models that do not include capacitance model or incomplete capacitance model give a complete capacitance model when specifying nonlinear device model parameters, in junction capacitance, include both depletion (at least) and diffusion capacitances
- Discontinuous jumps in waveforms when circuit contains nodes have no capacitive path to ground add small capacitor to ground or specify Cmin
- Capacitance model does not conserve charge GaAsFET Statz's, MOSFET Meyer's capacitance models switch to charge based model
- Large floating capacitors that are similar to the small-floating resistor problem in DC (finite precision problem) check capacitance unit, use smaller capacitance
- Discontinuous capacitance models in user defined model, SDD device fix the model

Slow Transient analysis

- Make sure I_RelTol and V_RelTol are set to 1e-3 or not set at all
- Decrease these values when higher accuracy is needed

Oscillator circuit does not oscillate

- Apply a short pulse at the beginning of the simulation
- Avoid using Gear2 or backward Euler

Circuit exhibits ringing or divergence

- Reduce Mu value from 0.5 toward 0 if trapezoidal rule is used
- Use Gear1 or Gear2

Circuit does not converge at first time point

- Reduce Min time step

Convergence Hints

Add break points

Use piecewise linear source to add break points to the region where the waveform changes abruptly

Reduce max time step

Ensure enough time points for sharp edges

Increase Max iterations per time step

Increase to 50 or more to increase the possible number of Newton iterations on each time step

Increase I_AbsTol

Try 1e-10 instead of the default 1e-12

Relax TruncTol

Increase this value 10 times or more to relax LTE tolerance

Relax I_Reltol and V_Reltol

Increase to 1e-3 to relax Newton convergence tolerance as well as LTE tolerance

Try different integration methods

Switch from trapezoidal to Gear's method

Using Convolution

- Do not set any convolution parameters (let the adaptive algorithm determine it)
- Set ImpMaxFreq first (larger than signal bandwidth)
- Set convolution parameters on component, not controller, when possible
- Do not allow measured data to be extrapolated (either set ImpMaxFreq or provide more data)

Convolution Modeling for Time-Domain Simulation

- In time-domain simulation, simulate devices that can only be defined in the frequency domain
  - Transmission lines with dispersion
  - Devices with frequency-dependent loss

    - Measured frequency-domain data
- Convolution is the key
    - Inverse Fourier transform of frequency-domain data produces the impulse response h(t)
    - The impulse response is convolved with time-domain signal

Time and Frequency Range

- Impulse response is calculated from the inverse Fourier transform of frequency-domain response frequency is uniformly sampled from 0 to some upper value
- Upper frequency sets the time-domain spacing of the impulse response
- Frequency spacing sets the length of the impulse response

Adaptive Impulse Response Calculation

- System bandwidth estimation is made from source frequencies and rise times (initial guess at fmax)
- Build a trial impulse response with 32 timepoints (very coarse frequency spacing)
- Build a second impulse response with 64 timepoints (less coarse frequency spacing)
- Double the number of timepoints until a good impulse response is obtained (increase fmax, decrease Df)
- y11 and y12 may be sampled with different fmax and Df
- Adaptive calculation is only done if ImpDeltaFreq is not set (set ImpDeltaFreq only when required)

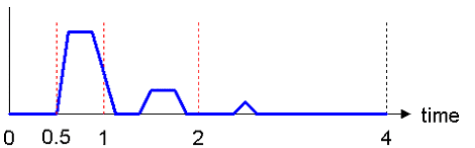
Good Impulse Responses

- Compare impulse responses with N and 2N points. The second impulse response is twice as long in time domain and has one-half the frequency spacing.
- An impulse is considered *good* when no appreciable energy is present in the second half of the impulse response
- If energy is present in the second half, the impulse is not long enough or it is noncausal

- If not good, the controller keeps doubling the length

  and

  The controller also tries doubling the maximum frequency, giving smaller impulse timesteps.

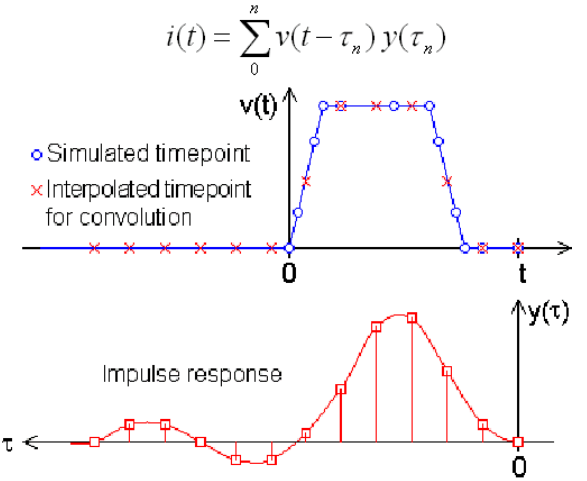


Interpolation

- The impulse response is sampled with a uniform timestep, but is not guaranteed to match the simulation timestep. The simulation may be using a variable timestep.
- Interpolate the signal v(t) to match the timepoints in the impulse response
- Do not interpolate the impulse response because the Fourier transform of the interpolated impulse response would no longer match the original frequency response

Impulse Evaluation

- Signal response at time zero extends back to minus infinity
- Evaluate the integral as a sum



Solving an Invalid Impulse Response

This problem is common during convolution. It does not necessarily imply noncausality but means that significant energy is present in the second half of the impulse response. In addition, simulation results may or may not be valid.

- Set ImpMaxFreq or ImpDeltaFreq: set ImpMaxFreq first, typically for measured data only.
- For every component that generates this message, fix each component one at a time to simplify the design.

Viewing an Impulse Response

- In an S-parameter simulation, analyze over the given frequency spacing and maximum frequency: inverse Fourier transform the response by plotting ts(x)
- In the time domain, apply an impulse and simulate: plot the transient result the pulse risetime is used to set fmax and thus can influence the impulse response

Setting ImpMaxFreq and ImpDeltaFreq

Generally a good impulse response can be found without manually setting ImpMaxFreq and ImpDeltaFreq.

- If ImpMaxFreq is set, the adaptive algorithm tries different lengths but does not modify fmax

- If ImpDeltaFreq is set, the adaptive algorithm is disabled and the impulse is calculated from ImpDeltaFreq and ImpMaxFreq
- Set ImpMaxFreq on the component, then set ImpDeltaFreq on component if necessary, and finally, set ImpMaxFreq on the transient controller if necessary
- For transmission lines, set ImpMaxFreq to at least n/td, where td is the delay time and n is a small integer (2-3)
- For lowpass and bandpass filters, set ImpMaxFreq to at least twice the upper passband edge

Measured Data with S2P Component

- The algorithm that calculates the impulse response has no special knowledge of the component it's working on and assumes data is available at any desired frequency. It has no knowledge of flow and fhigh or frequency spacing of measured data
- S2P interpolates and extrapolates data as needed
- Supply good data to prevent dangerous extrapolation extending down to DC and up to fmax
- Set ImpMaxFreq on S2P component to match frequency limits in datafile (avoid extrapolation)
- Typically there is not enough frequency-domain data in the S2P file for use in the simulation
  - Given a pulse with a risetime of tr, the equivalent bandwidth is 2.2/tr (0.1 ns risetime represents a 22 GHz bandwidth)
  - Package models typically must be measured up to 10x higher than the signal frequency to represent transmission line effects well

Solving a Noncausal Impulse Response

This is the second most common problem during convolution. The time-domain simulation starts at time zero and moves forward in time, calculating the value of next timepoint from all previous timepoints. And the Controller overcomes this by introducing a delay to force causality:

- Length of delay set to ImpNoncausalLength (default=32) with timestep set by default ImpMaxFreq

Simulation results will not be accurate because of the added delay, especially if the delay is added in a critical timing or phase path.

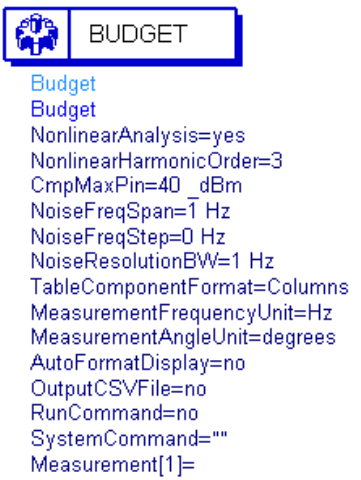
All physically realizable devices are causal (the output is dependent only on past states and not any future states) while noncausal devices are nonphysical. Some ADS components, user-defined data, or equations may be noncausal.

- Frequency-dependent real part with constant imaginary part, for example resistance as a function of frequency without any reactance
- Constant real and constant non-zero imaginary part
- Negative time delays
- INDQ, CAPQ, PLCQ, SLCQ have problems in some modes

RF Budget Controller

Use the Budget controller for budget analysis of an RF system. This analysis enables you to determine the linear and nonlinear characteristics of an RF system comprising a cascade of 2-port, 2-pin linear or nonlinear components.

The Budget controller provides built-in budget measurements and improved budget noise measurements.



Use the Budget controller to

- Modify simulations using tuning, parameter sweeps, optimization, yield analysis, etc.
- Include AGC loops to control gain and set power levels at specific points in the RF system.
- Select alternate budget paths.

Optimization and Statistical Design Controllers

Optimization and statistical design controllers are used in conjunction with RF/Analog and signal processing simulation controllers to:

- Characterize and improve an unknown process such as the response of a design
- Identify variables that contribute significantly to variations in performance
- Vary parameter values to identify combinations that deliver the desired yields

Design applications include:

- Optimizing gain and matching
- Filter response optimization
- Pulse risetime tuning
- Carrier lock time and residual loop error optimization
- Fixed-point bit-width optimization
- Maximize manufacturing yield

Nominal Optimization Controller

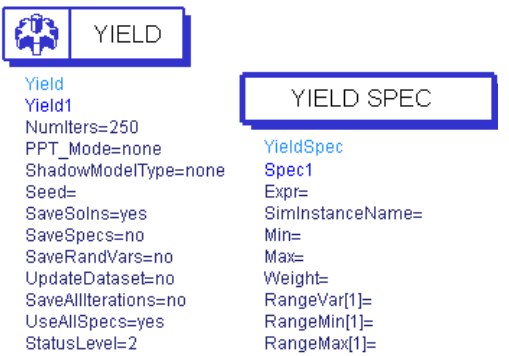
Use the nominal optimization (Optim) controller in combination with a Goal component to satisfy performance goals. Optimizers that compare calculated and desired responses and modify design parameter nominal values to bring the calculated response closer to that desired can be selected in the Optim setup.

Monte Carlo Controller

The MonteCarlo analysis controller is used to randomly vary network statistical parameter values according to statistical distributions to determine the overall performance variation. This process involves simulating the design over a given number of trials in which the statistical variables have values that vary randomly about their nominal values with specified probability distribution functions.

Yield Analysis Controller

The Yield analysis controller is used in combination with a YieldSpec component to vary a set of statistical parameter values (using specified probability distributions) to determine how many possible combinations result in satisfying predetermined performance requirements. This process involves simulating the design over a given number of trials in which the statistical variables have values that vary randomly about their nominal values with specified probability distribution functions. The numbers of passing and failing trials are recorded and these numbers are used to calculate an estimate of the yield.



Yield Optimization Controller

Use the yield optimization (YieldOptim) controller in combination with a YieldSpec component to perform multiple yield analyses with the goal of adjusting the nominal values of the statistical variables to maximize the yield estimate. During yield optimization, each yield improvement is referred to as a design iteration.

Design of Experiments Controller

Use the design of experiments (DOE) controller in combination with DoeGoal components to perform an experiment and collect response data. You can then analyze the data using statistical methods. Sequential application of this methodology can be used to improve the statistical performance of a given circuit or system. Because of an inherent compromise between statistical performance prediction accuracy and the number of input variables, a *screening* experiment is used to identify variables that contribute significantly to performance variation. A *refining* experiment can then be used to *hone in* on the target statistical response.

Chapter 5: Data Display

ADS uses datasets to store the simulation information you generate when analyzing designs. You can display this information for analysis using the Data Display window. A Data Display window can also be used to display data imported from other sources.

In a Data Display window you can:

- Display data in a variety of plots and formats
- Use markers to read specific data points on traces
- Use equations to perform operations on data
- Annotate results using text and illustrations

Once a simulation is complete, data is automatically displayed if you:

- Specified a dataset and display before simulation
- Used a schematic template for an Analog/RF simulation
- Specified Rectangular in the Plot parameter in a sink for a Signal Processing simulation

Creating Data Displays

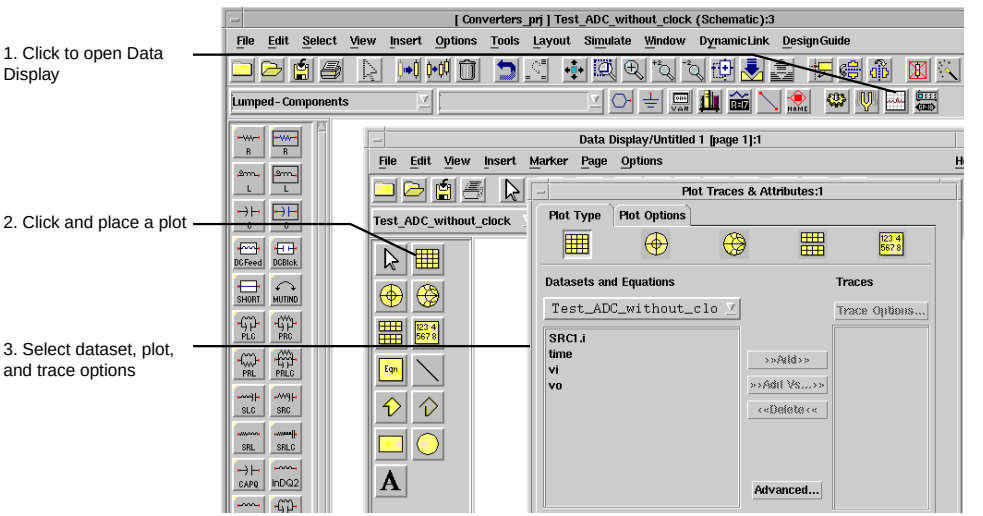
To create a data display:

1. Choose a plot type for the display
2. Choose the dataset that contains the data you want to display
3. Select the data variable to be displayed
4. Choose a trace type for the display

To enhance the display you can also add:

- Markers to identify specific data points
- Annotations using text and illustrations
- Legends to help identify specific traces

If you used a template to create the design you have simulated, the initial setup and configuration to create displays for data analysis is done for you.

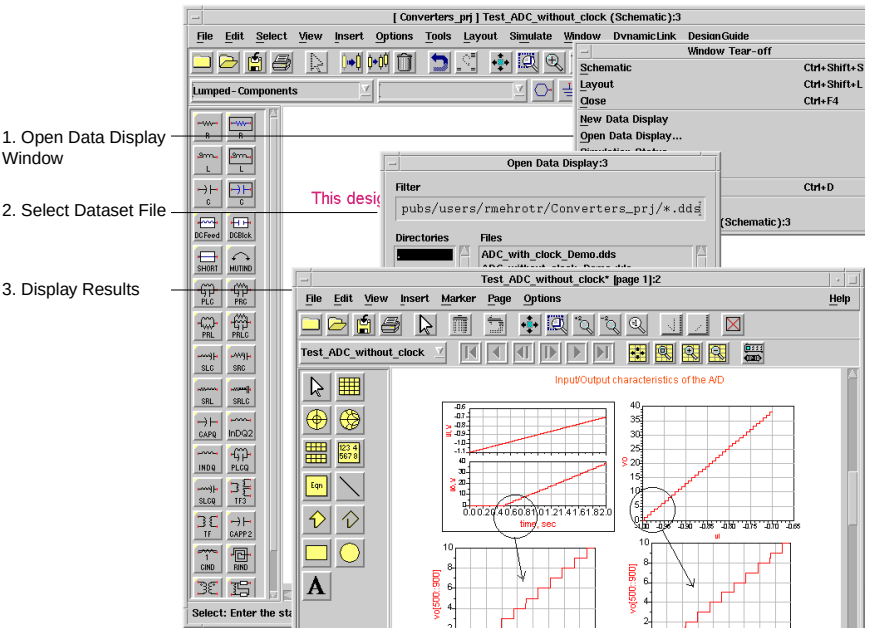


Viewing Results

To view simulation results from the Main, Schematic, or Layout window choose **Window > Open Data Display** and use the dialog box to locate and open the results.

To display a list of data display files in the File Browser pane of the Main window choose **View > Show All Files**.

To display simulation results:



Display Options

The following plot, trace, and data options can be used to display data for analysis:

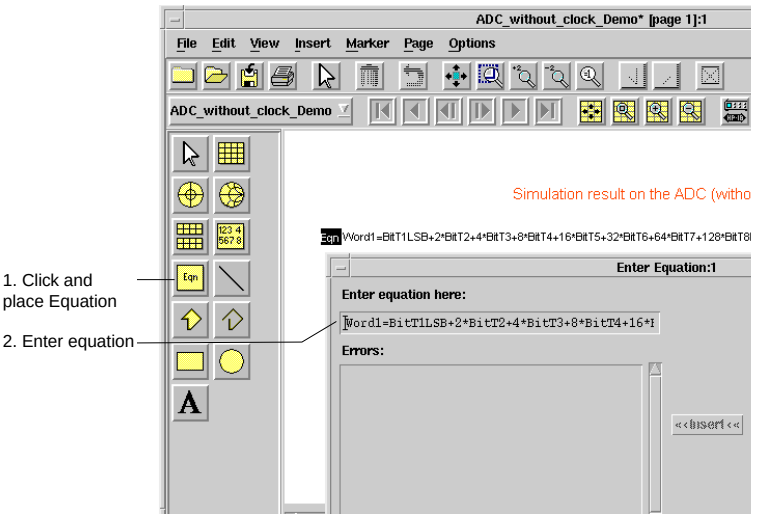
Plot Type	Trace Type	Data Source
Rectangular plot Stacked plot Smith chart Polar plot	Auto, Bus, Linear, Scatter, Spectral, Histogram, Digital, Sampled	Simulation dataset File Formats: Touchstone MDIF Citifile ICCAP
List		

Using Functions

Measurement Equations can be used to perform operations on data generated during a simulation. These equations are created using functions that are based on AEL, the Application Extension Language.

Note Data from a marker can also be used as part of an equation. To insert a marker, choose **Marker** > **New** and click the trace where you want to insert it.

To create and insert a function:



Chapter 6: Importing and Exporting

To import or export a design (schematic or layout):

- Choose **File** > **Import** (**Export**) from a Schematic or Layout window
- Choose a file type
- Enter a file name

To export ADS Ptolemy designs:

Choose **Tools** > **Export ADS Ptolemy Design** > **As WTB Model to RFDE**

or

Choose **Tools** > **Export ADS Ptolemy Design** > **As AMS Model to AMSD-ADE**

To import or export data:

- Choose **Tools** > **Data File Tool** from a Schematic window
  (for Touchstone, MDIF, CITI, and IC-CAP files)
- Choose **Tools** > **Connection Manager Client** from a Schematic window
  (for data from connected instruments)
- Choose **Tools** > **Instrument Server** (Windows only)
  (read and write data from various legacy instrument sources in a variety of file formats)

Formats for Design Exchange

Format	Import	Export
DXF (.dxf)		Layout
EGS Archive Format (_a)	Layout	Layout
EGS Generate Format (_g)	Layout Schematic	Layout
GDSII Stream Format (.gds)	Layout	Layout
Gerber (.gbr)		Layout
Gerber Viewer (.msk, .gbr)		Layout
HPGL/2 (.hpg)	Layout Schematic	Layout
HP IFF (.iff)	Layout Schematic	Layout Schematic
IGES (.igs)	Layout	Layout
Mask File (.msk)	Layout Schematic	Layout
MGC/PCB (.iff)		Layout
Spice (.cir, .cki, .iff, .net)	Schematic	

Drawing Exchange Format (DXF)

This format was developed by Autodesk for its AutoCAD product to transfer geometric data between systems. Like the mask file format, it provides a simple geometric representation of data. DXF files can be transferred between PC-based or UNIX-based systems.

Engineering Graphics System (EGS)

This format is a general graphics format used for capturing manually entered designs. EGS has been applied to ICs, Micro-circuits, Hybrids, and PC Board design applications. Using this format, you can easily exchange data with other programs using EGS formats. In addition, EGS facilitates better artwork translation with ADS.

- The Generate format is a flattened list of EGS primitives specified in the user-defined unit space.
- The Archive Format is a hierarchically organized list of EGS primitives specified in the user-defined unit space. Information such as drawing shapes, layout units, database precision, and grid spacing is included.

GDSII Stream Format (Calma)

This format is an industry standard for translating final mask data to foundries. ADS reads GDSII versions 4.0 through 6.0 and writes GDSII version 6.0. Unlike other data formats, GDSII stream format is binary. You cannot easily view or edit a stream format file using a text editor. This format is easily translated between different CAD systems because it represents a highly restrictive data type.

Gerber

This format refers to various data input formats that Gerber Scientific uses to drive its photoplotters. The Gerber format is used by photoplotters produced by other manufacturers also. The program supports various types of Gerber output via mask files to either the Gerber or DXF translator.

Gerber Viewer

This format appears as an export file option. It is not a file format, but you can use it to view Gerber or mask files to help verify the correctness of your data if the files meet the following criteria:

- Use either absolute or incremental data coordinates
- Support apertures from D10-D999
- Have data formats from 0.1 to 4.5

HPGL/2

This format is a subset of the HPGL/2 printer/plotter language. When creating a graph or chart in another tool, you can write the graphics data to an HPGL/2 output file, then import the file into ADS. In ADS, the HPGL data is transformed into forms and shapes that can be edited and manipulated like any other drawing. Additional text, annotation, scaling or editing can be added.

Intermediate File Format (IFF)

This format is an ASCII file with a simple, line-oriented command structure and a fairly rich set of constructs. This format is machine- and application-independent, thus simplifying design data transfer. IFF files are used as the exchange mechanism

when transferring designs between ADS and third-party EDA tools such as Mentor Graphics Design Architect and Cadence Analog Artist.

Initial Graphics Exchange Specification (IGES)

This format is an approved ANSI standard that is used extensively throughout the computer-aided design and manufacturing world. It can represent both mechanical and electrical design data in two and three dimensions. The IGES standard for the transfer of electrical design data is known as CALS specification. ADS supports version 4.0 and 5.0 IGES formats. It reads and writes IGES CALS Level 1 (technical illustration) and Level 3 (electrical/electronic applications) files.

Mask

This format is a simple flat (non-hierarchical) geometric description. The format facilitates the transfer of simple geometric data for final mask processing. Only geometric forms are described in a mask file; simulation data, element parameters, substrate definitions, and hierarchy are not included.

MGC/PCB

These files are IFF files that are used exclusively for Mentor Graphics design transfers. MGC/PCB files write to a specific location each and every time. When you select this format, the filename and location of the IFF transport is determined automatically.

Spice

Simulation Program with Integrated Circuit Emphasis (Spice) has become a simulation tool used by engineers throughout the world for simulating circuits of all types. After its development at the University of California Berkeley, Spice has been commercialized and modified by a large number of vendors and also adopted and modified by electronics companies for their own in-house use.

Chapter 7: Additional Resources

Use the following ADS resources to optimize your design tasks.

Documentation

The Agilent EEsof Product Documentation website **http://www.agilent.com/find/eesof-docs/** includes information and links to complete documentation for all EEsof product.

For ADS it provides documentation for ADS 1.5 onward..

- **Release Notes** latest release information.
- **What's New** highlights of what is new in the current release.
- **Manuals** links to the complete documentation set.
- **Examples** descriptions of examples used to solve real-life design tasks.
- **DesignGuides** links to application-focussed DesignGuide documentation.
- **Search** global search of ADS documentation.
- **PDF Files** printable files for all ADS documentation.

The Agilent EEsof EDA website **http://www.agilent.com/find/eesof/** also includes information and links to the following resources.

- **EDA library** articles, papers, application notes, and other publications.
- **Foundry Partners** links to leading IC manufacturers who provide design libraries for ADS.
- **Success Stories** what real users, large and small, have to say about their successes with ADS.
- **Training Classes** current list of training courses being offered worldwide.
- **Applications** applications information including downloadable example files and publications.
- **Agilent EEsof Knowledge Center** database of information, discussions, and downloads.

Support Contacts

Agilent EEsof worldwide technical support is available Monday through Friday. The toll-free North America hotline is open 6:00 am to 5:00 pm PT. Throughout Europe, the localized Online Technical Support Centers are open 8:30 am to 5:30 pm, local time; throughout Asia, the localized Customer Response Centers are open 9:00 am to 6:00 pm, local time.

The e-mail addresses for the various regions are listed below. However, for both the regional e-mail addresses and local telephone numbers for more than 25 countries, please refer to the Agilent EEsof Support Web site at

http://www.agilent.com/find/eesof-supportcontact/

North America

Phone: 1 800 47 EEsof (473-3763) · Fax: 707-577-3511

e-mail: eesof_support@agilent.com

Europe: e-mail: eesof-europe_support@agilent.com

Japan: e-mail: eesof-japan_support@agilent.com

Korea: e-mail: eesof_korea@agilent.com

Asia: e-mail: eesof-asia_support@agilent.com

Index