OrCAD PSpice® Optimizer

User's Guide

Trademarks

OrCAD, OrCAD Layout, OrCAD Express, OrCAD Capture, OrCAD PSpice, and OrCAD PSpice A/D are registered trademarks of OrCAD, Inc. OrCAD Capture CIS, and OrCAD Express CIS are trademarks of OrCAD, Inc.

Microsoft, Visual Basic, Windows, Windows NT, and other names of Microsoft products referenced herein are trademarks or registered trademarks of Microsoft Corporation.

All other brand and product names mentioned herein are used for identification purposes only, and are trademarks or registered trademarks of their respective holders.

Part Number 60-30-637

First edition 30 November 1998

Technical Support	(503) 671-9400
Corporate offices	(503) 671-9500
OrCAD Japan K.K.	81-45-621-1911
OrCAD UK Ltd.	44-1256-381-400
Fax	(503) 671-9501
General email	info@orcad.com
Technical Support email	techsupport@orcad.com
World Wide Web	http://www.orcad.com
OrCAD Design Network (ODN)	http://www.orcad.com/odn

OrCAD®

9300 SW Nimbus Ave.
Beaverton, OR 97008 USA

Contents

Figures

Tables

Before you begin

Welcome to OrCAD®

Welcome to the OrCAD® family of products. Whichever programs you have purchased, we are confident that you will find that they meet your circuit design needs. They provide an easy-to-use, integrated environment for creating, simulating, and analyzing your circuit designs from start to finish.

OrCAD PSpice Optimizer overview

The OrCAD PSpice Optimizer is a circuit optimization program that improves the performance of analog and mixed analog/digital circuits. The PSpice Optimizer is fully integrated with other OrCAD programs. This means you can design your circuit with OrCAD Capture, simulate and analyze results with OrCAD PSpice A/D (or OrCAD PSpice) and optimize performance within the same environment.

How to use this guide

This guide is designed so you can quickly find the information you need to use the OrCAD PSpice Optimizer.

This guide assumes that you are familiar with Microsoft Windows (NT or 95), including how to use icons, menus, and dialog boxes. It also assumes you have a basic understanding about how Windows manages applications and files to perform routine tasks, such as starting applications and opening, and saving your work. If you are new to Windows, please review your *Microsoft Windows User's Guide.*

Typographical conventions

Before using the **OrCAD PSpice Optimizer**, it is important to understand the terms and typographical conventions used in this documentation.

This guide generally follows the conventions used in the *Microsoft Windows User's Guide*. Procedures for performing an operation are generally numbered with the following typographical conventions.

Notation	Examples	Description
Ctrl+R	Press Ctrl+R	A specific key or key stroke on the keyboard.
`monospacefont`	Type VAC...	Commands/text entered from the keyboard

Related documentation

Documentation for OrCAD products is available in both hard copy and online. To access an online manual instantly, you can select it from the Help menu in its respective program (for example, access the Capture User's Guide from the Help menu in Capture).

Note *The documentation you receive depends on the software configuration you have purchased.*

The following table provides a brief description of those manuals available in both hard copy and online.

This manual...	Provides information about how to use...
OrCAD Capture User's Guide	OrCAD Capture, which is a schematic capture front-end program with a direct interface to other OrCAD programs and options.
OrCAD Layout User's Guide	OrCAD Layout, which is a PCB layout editor that lets you specify printed circuit board structure, as well as the components, metal, and graphics required for fabrication.
OrCAD PSpice A/D & Basics+ User's Guide	PSpice A/D, the Stimulus Editor, and the Model Editor utility, which are circuit analysis programs that let you create, simulate, and test analog and digital circuit designs. It provides examples on how to specify simulation parameters, analyze simulation results, edit input signals, and create models.
OrCAD PSpice & Basics User's Guide	OrCAD PSpice & OrCAD PSpice Basics, which are circuit analysis programs that let you create, simulate, and test analog-only circuit designs.
OrCAD PSpice Optimizer User's Guide	OrCAD PSpice Optimizer, which is an analog performance optimization program that lets you fine tune your analog circuit designs.

The following table provides a brief description of those manuals available online *only*.

This online manual...	Provides this...
OrCAD PSpice A/D Online Reference Manual	Reference material for PSpice A/D. Also included: detailed descriptions of the simulation controls and analysis specifications, start-up option definitions, and a list of device types in the analog and digital model libraries. User interface commands are provided to instruct you on each of the screen commands.
OrCAD Application Notes Online Manual	A variety of articles that show you how a particular task can be accomplished using OrCAD's products, and examples that demonstrate a new or different approach to solving an engineering problem.
Online Library List	A complete list of the analog and digital parts in the model and symbol libraries.

Online help

Choosing Search for Help On from the Help menu brings up an extensive online help system.

The online help includes:

- step-by-step instructions on how to use OrCAD PSpice Optimizer features
- reference information about OrCAD PSpice Optimizer
- Technical Support information

If you are not familiar with Windows (NT or 95) Help System, select How to Use Help from the Help menu.

If you have the demo version

The demonstration version of the PSpice Optimizer has the following requirements and limitations:

- Requires OrCAD PSpice A/D with Capture demonstration package.
- Is limited to one goal, one parameter, and one constraint.

Things you need to know

1

Chapter overview

This chapter introduces the purpose and function of the PSpice Optimizer, the optimization process, and related terms.

- What is the PSpice Optimizer? on page 1-20 describes optimizer capabilities and the criteria designs must meet for successful optimization.
- Using the PSpice Optimizer with other OrCAD programs on page 1-22 presents the high-level design flow for optimization and how other OrCAD programs are integrated into each design phase.
- Terms you need to understand on page 1-23 defines the terms that are important for optimizing designs successfully.

What is the PSpice Optimizer?

The OrCAD PSpice Optimizer is a circuit optimization program that improves the performance of analog and mixed analog / digital circuits.

Run optimizations The PSpice Optimizer performs iterative simulations, while adjusting the values of design parameters until performance goals, subject to specified constraints, are nearly or exactly met. Constraints can include simple bounds on parameter values and nonlinear functions. The PSpice Optimizer also computes Lagrange multipliers that provide information on the cost of each constraint on the solution.

Explore performance tradeoffs When you enter new values for design parameters, the PSpice Optimizer provides graphical feedback showing performance. You can also tweak goal and constraint values to examine changes to parameter values.

Fit model parameters Given a parameterized model, a set of measured data points, and a good starting point for the parameter values, the PSpice Optimizer fits a more accurate model.

Designs that you can optimize

Optimization problems are not always solvable by a particular algorithm.

A design that you can optimize must meet the following criteria:

- It works; that is, it simulates with PSpice to completion and behaves as intended.
- One or more of its components have a variable value, and each value that is varied relates to an intended performance goal.
- An algorithm exists to measure its performance as a function of the variable value.

If you can visualize which factors should be adjusted to improve performance, and how you would manually step through the optimization process (even though the computations might seem unwieldy), then the design is a good candidate for the PSpice Optimizer.

Designs that you cannot optimize

You cannot use the PSpice Optimizer to:

- Create a working design. This especially applies when you begin with a design that is far from meeting specifications.
- Optimize a design in which the circuit has several states where a small change in a parameter value causes a change of state. For example: A flip-flop is *on* for some parameter value, and *off* for a slightly different value.

Using the PSpice Optimizer with other OrCAD programs

Because you can use Capture and PSpice A/D to design and simulate at the system, subcircuit, or component level, use the PSpice Optimizer to optimize at whatever level is most appropriate.

The PSpice Optimizer is fully integrated with other OrCAD programs. This means you can design your circuit with OrCAD Capture, simulate with OrCAD PSpice A/D (or OrCAD PSpice), analyze results with the PSpice A/D waveform viewer and optimize performance within the same environment. Figure 1 illustrates the typical design flow for circuit optimization.

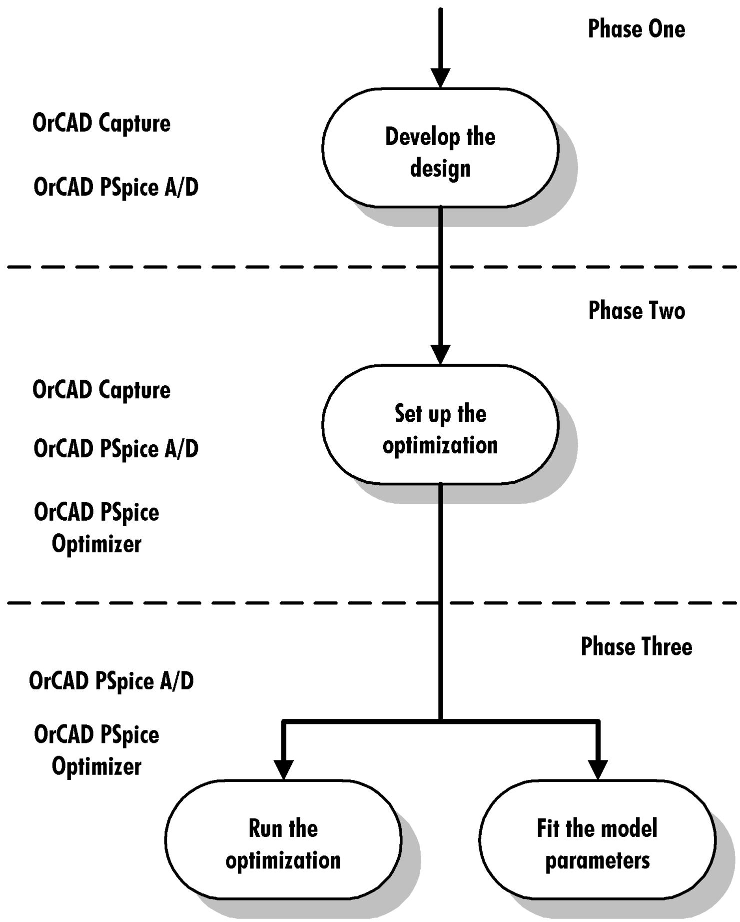


Figure 1 *Optimization design flow.*

See *Chapter 2, Primer: How to optimize a design* for a detailed description of each design phase.

Terms you need to understand

Optimization Optimization is the process of fine-tuning a design by varying design parameters between successive simulations until performance comes close to (or exactly meets) the ideal performance.

The PSpice Optimizer solves four types of optimization problems as described in Table 1.

Table 1 *Optimization problems*.*

Problem type	PSpice Optimizer action	Example
Unconstrained minimization	Reduces the value of a single goal	Minimize the propagation delay through a logic cell
Constrained minimization	Reduces the value of a single goal while satisfying one or more constraints	Minimize the propagation delay through a logic cell while keeping the power consumption of the cell less than a specified value
Unconstrained least squares**	Reduces the sum of the squares of the individual errors (difference between the ideal and the measured value) for a set of goals	Given a terminator design, minimize the sum of squares of the errors in output voltage and equivalent resistance
Constrained least squares	Reduces the sum of squares of the individual errors for a set of goals while satisfying one or more constraints	Minimize the sum of squares of the figures of merit for an amplifier design while keeping the open loop gain equal to a specified value

* All four cases allow simple bound constraints; that is, lower and upper bounds on all of the parameters. The PSpice Optimizer also handles nonlinear goals and constraints.

** Use unconstrained least squares when fitting model parameters to a set of measurements, or when minimizing more than one goal.

Parameter A parameter defines a property of the design for which the PSpice Optimizer attempts to determine the best value within specified limits. A parameter can:

- Represent component values (such as resistance, R, for a resistor).
- Represent other component property values (such as slider settings in a potentiometer).
- Participate in expressions used to define component values or other component property values.

The PSpice Optimizer can optimize designs with up to eight variable parameters.

See **Chapter 6,** Tutorial: Exploring design tradeoffs (active filter) for a working example showing parameterized slider values.

For example: A potentiometer part in a schematic uses the SET property to represent the slider position. You can assign a parameterized expression to this property to represent variable slider positions between 1 and 0. During optimization, the PSpice Optimizer varies the parameterized value of the SET property.

For more information, see *Goal* and Constraint on page 1-26.

Specification A specification describes the ideal behavior of a design in terms of goals and constraints.

For example: For a given design, the gain shall be 20 dB ±1 dB; for a given design, the 3 dB bandwidth shall be 1 kHz; for a given design, the rise time must be less than 1 usec.

A design can have up to eight goals and constraints in any combination, but there must *always* be at least one goal. You can easily change a goal to a constraint and vice-versa.

The PSpice Optimizer accepts specifications in two formats: internal and external.

Internal specifications

An internal specification is composed of goals and constraints defined in terms of target values and ranges, which are entered into the PSpice Optimizer through dialog boxes.

External specifications

An external specification is composed of measurement data, which are defined in an external data file that is read by the PSpice Optimizer.

Target value A target value is the ideal operating value for a characteristic of the design as defined by a goal or constraint specification.

Goal A goal defines the performance level that the design *should* attempt to meet (for instance, minimum power consumption). A goal specification includes:

- The name of the goal.
- A target value and an acceptable range.
- A circuit file to simulate.
- An evaluation for measuring performance.
- An analysis type used for simulation-based evaluations.

The goal specification can also include:

- The name of the file containing the PSpice A/D goal function definitions (.PRB file).
- When using an external specification, the name of the file containing measured data and the columns of data to be used as reference.

Note *Typically, the PSpice Optimizer measures performance using an evaluation that requires a simulation, and therefore, you must specify the circuit file for the simulation. However, when measuring performance using PSpice Optimizer expressions that do not require a simulation, you do not need to specify a circuit file.*

Constraint A constraint defines the performance level that the design *must* fulfill in which the target value exceeds, falls below, or equals a specified value (for instance, an output voltage that must be greater than a specific level). The constraint specification includes:

- The name of the constraint.
- A target value and an acceptable range.
- A circuit file to simulate. (See note on previous page.)
- An evaluation for measuring performance.
- An analysis type used for simulation-based evaluations.
- An allowed relationship between measured values and the target value, which can be one of the following:

<=	measured value must be less than or equal to the target value
=	measured value must equal the target value
>=	measured value must be greater than or equal to the target value

The constraint specification can also include the name of the file containing the PSpice A/D goal function definitions (.PRB file).

Constraints are often nonlinear functions of the parameters in the design.

For example: Bandwidth can vary as the square root of a bias current and as the reciprocal of a transistor dimension.

See Optimization on page 1-23 for more on least-squares and minimization algorithms.

Performance The performance of a design is a measure of how closely its specifications' calculated values approach their target values for a given set of parameter values. When there are multiple specifications (at least one of which is a goal), the PSpice Optimizer uses the sum of the squares of their deviations from target to measure closeness. For a single specification (goal), the PSpice

Optimizer uses either the goal's value, or the square of its deviation from target.

Each aspect of a design's performance is found by either:

- First performing the appropriate simulation, then running PSpice A/D to measure characteristics of the resulting waveform(s)

or

- Evaluating PSpice Optimizer expressions

In many cases (particularly if there are multiple conflicting specifications), it is possible that the PSpice Optimizer will not meet all of the goals and constraints. In these cases, optimum performance is the best *compromise* solution—that is, the solution that comes closest to satisfying each of the goals and constraints, even though it may not completely satisfy any single one.

Evaluation An evaluation is an algorithm that computes a single numerical value, which is used as the *measure of performance* with respect to a design specification.

The PSpice Optimizer accepts evaluations in one of these three forms:

- Single-point PSpice A/D trace function
- PSpice A/D goal function
- PSpice Optimizer expression

Given evaluation results, the PSpice Optimizer determines whether or not the changes in parameter values are improving performance, and determines how to select the parameters for the next iteration.

Trace function A trace function defines how to evaluate a design characteristic when running a *single-point analysis* (such as a DC sweep with a fixed voltage input of 5 V). For example: V(out) to measure the output voltage; I(d1) to measure the current through a component.

Refer to the online *OrCAD PSpice A/D Reference Manual* for the variable formats and mathematical functions you can use to specify a trace function.

Refer to the Goal Function Wizard in PSpice A/D and your *PSpice A/D User's Guide* for information on how to develop and specify goal functions.

Here are some quick tips. In PSpice A/D:

- To test the value returned by a specified goal function, choose Eval Goal Function from the Trace menu.
- To see the waveforms and marked points used to evaluate a goal function, select Display Evaluation in the Options dialog box (from the Tools menu, choose Options to display this dialog box).

PSpice A/D goal function A goal function defines how to evaluate a design characteristic when running any kind of analysis other than a single-point sweep analysis. A goal function computes a single number from a waveform. This can be done by finding a characteristic point (e.g., time of a zero-crossing) or by some other operation (e.g., RMS value of the waveform).

For example, you can use PSpice A/D goal functions to:

- Find maxima and minima in a trace.
- Find distance between two characteristic points (such as peaks).
- Measure slope of a line segment.
- Derive aspects of the circuit's performance which are mathematically described (such as 3 dB bandwidth, power consumption, and gain and phase margin).

See Gain on page 7-133 for an example of the YatX goal function definition.

To write effective goal functions, determine what you are attempting to measure, then define what is mathematically special about that point (or set of points).

Note *Be sure that the goal functions accurately measure what they are intended to measure. Optimization results highly depend on how well the goal functions behave. Discontinuities in goal functions (i.e., sudden jumps for small parameter changes) can cause the optimization process to fail.*

PSpice Optimizer expression A PSpice Optimizer expression defines a design characteristic. The expression is composed of optimizer parameter values, constants, and the operators and functions shown in Table 2.

For example: To measure the sum of resistor values for two resistors with parameterized values named R1val and R2val, respectively, use the PSpice Optimizer expression *R1val + R2val*.

Table 2 *Valid Operators and Functions for PSpice Optimizer Expressions.*

Operator	Meaning
+	addition
-	subtraction
*	multiplication
/	division
**	exponentiation
exp	e^x
log	ln(x)
log10	$\log_{10}(x)$
sin	sine
cos	cosine
tan	tangent
atan	arctangent

Note *Unlike trace functions and goal functions, PSpice Optimizer expressions are evaluated without using a simulation.*

Derivative A derivative defines mathematically how a specification value changes with a small change in parameter value.

See Derivatives on page 4-95 for a detailed discussion.

For a given design, the PSpice Optimizer calculates derivatives for each specification with respect to each parameter. Within an applicable range, the optimizer uses the derivatives to estimate new values for the goals and constraints.

Primer: How to optimize a design

2

Chapter overview

This chapter guides you through the basic steps needed to setup and run an optimization using a simple diode biasing design.

- Optimizing a diode biasing circuit—the objective on page 2-33 describes the sample circuit and its ideal operating characteristics.
- Why use optimization? on page 2-34 explains why fine-tuning your design using the PSpice Optimizer saves time.
- Phase One: Developing the design on page 2-35 walks you through the steps needed to create a working design.
- Phase Two: Setting up the optimization on page 2-37 walks you through the steps needed to define the parameters, goals, and constraints that describe the optimization.

- Phase Three: Running an optimization on page 2-42 walks you through the steps needed to optimize and finalize the design.

Optimizing a diode biasing circuit—the objective

Assume that you want to design a circuit that drives a current of 1mA (±5 µA) through a diode (D1N914) using a 5Vvoltage source and a series resistor to control the current through the diode. A circuit such as this is shown in Figure 2.

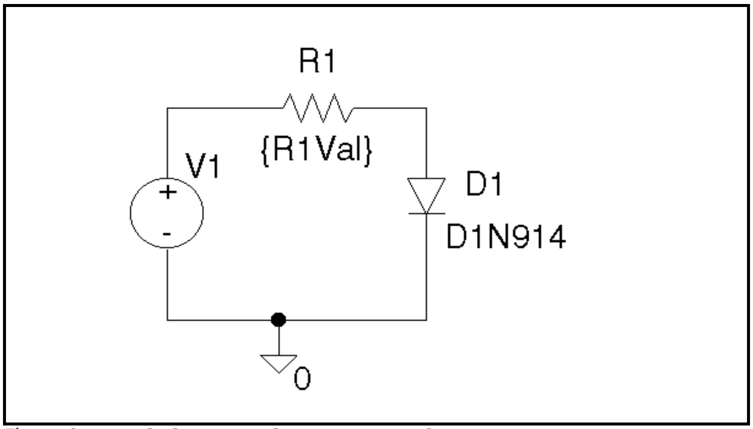


Figure 2 *Diode biasing design example.*

Your objective is to find a value for resistor R1 so that current through diode D1 falls in the range 0.995 mA to 1.005 mA.

Why use optimization?

When solving complex problems, the manual approach can be too unwieldy to consider. For example:

- When your design has multiple parameters or complicated parameter interactions, you may find it's nearly impossible to know which parameters to change, and how best to change them.
- When solving for multiple specifications, the solution often depends on the order in which goals and constraints are optimized. This sequential approach can miss possible solutions since it is impractical to repeat the process starting with a different goal or constraint each time.

Because the PSpice Optimizer solves for all specifications at once, and simultaneously adjusts all parameters between iterations, you end up examining fewer possible solutions.

To solve the problem manually, you could assign an arbitrary value to R1, manually calculate the current, then make an *educated guess* to adjust the values until a satisfactory solution is found. Or, you might use a simulation to sweep the value for R1 with a DC Sweep analysis, carefully analyzing the results to find the best solution.

These manual methods have two major disadvantages:

- Because the diode is a *non-linear* device, manual calculations can be time-consuming.
- Sweeping a parameterized value can take a large number of simulations, depending on the range and increment selected.

The PSpice Optimizer automates these processes by handling calculations for you and intelligently directing the series of simulations. Given results of the previous simulations, the optimizer automatically adjusts the parameterized value of R1 for the next run, thus eliminating unnecessary iterations, which in turn, provides a solution more quickly and with less effort.

Once the PSpice Optimizer settles on the best solution, you can still explore available tradeoffs. When done manually, this iterative process can be difficult and frustrating. With the optimizer, you can tweak the parameter(s) and immediately determine whether the design still meets specifications. You can also change the value of the specification(s) and immediately determine how parameter values change. If you are dissatisfied with the result after any change, you can always return to the last set of values.

Phase One: Developing the design

Before optimizing, you must have a *working* circuit. This means first drawing the design, then iteratively simulating with PSpice A/D and adjusting the design until the circuit operates with the intended behavior.

Phase 1 is also the time to investigate:

- The effects of individual components by replacing component values with parameters or parameterized expressions.
- Using PSpice A/D to perform a DC, AC, or parametric sweep of each parameterized value.

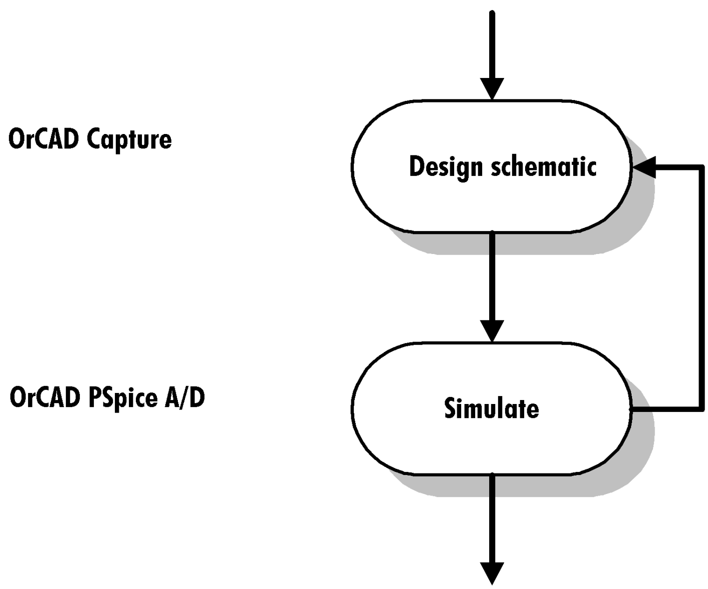


Figure 3 *Design flow for developing the design.*

To draw the schematic page for the diode biasing design

1. In Capture's Project Manager, choose New under the File menu, then choose Project.
2. Enter `MYDIODE` as the name of the new project.
3. Select Analog or Mixed-signal Circuit Wizard to make this a design that can be simulated with PSpice.
4. Click OK, then click Finish.

   A blank schematic page will appear.
5. From the Place menu, choose Part to select and place the following parts on the schematic page:

Note *When you initially place resistor R1, its value is 1 k. Later, when you set up the optimization, you will parameterize R1's value as shown in Figure 2.*

R	resistor R1
D1N914	diode D1
VSRC	voltage source V1

6 Choose Place Ground to select and place the following simulaton parts on the schematic page:

0	analog ground 0

7 From the Place menu, choose Wire to connect the parts as shown in Figure 2.

8 Click on the VSRC part (V1) to select it.

9 From the Edit menu, choose Properties, then User Properties to set V1's DC property to 5V.

10 From the File menu, choose Save.

The PSpice optimizer advantage

To determine a value for R1 manually, you can set up a parametric analysis of a DC sweep where:

- The value of R1 steps from 4 k to 5 k in increments of 0.1 k
- The DC sweep analysis is a single-point voltage analysis at 5 V

Such an analysis requires *eleven* PSpice simulations. Using Probe, the resistor value giving rise to 1 mA current through D1 is 4.14 k.

The remainder of this chapter shows how to use the PSpice Optimizer to determine the same solution automatically using *fewer* simulations.

Phase Two: Setting up the optimization

Now that preliminary design development is complete, you are ready to define the optimization parameters, goals, and constraints.

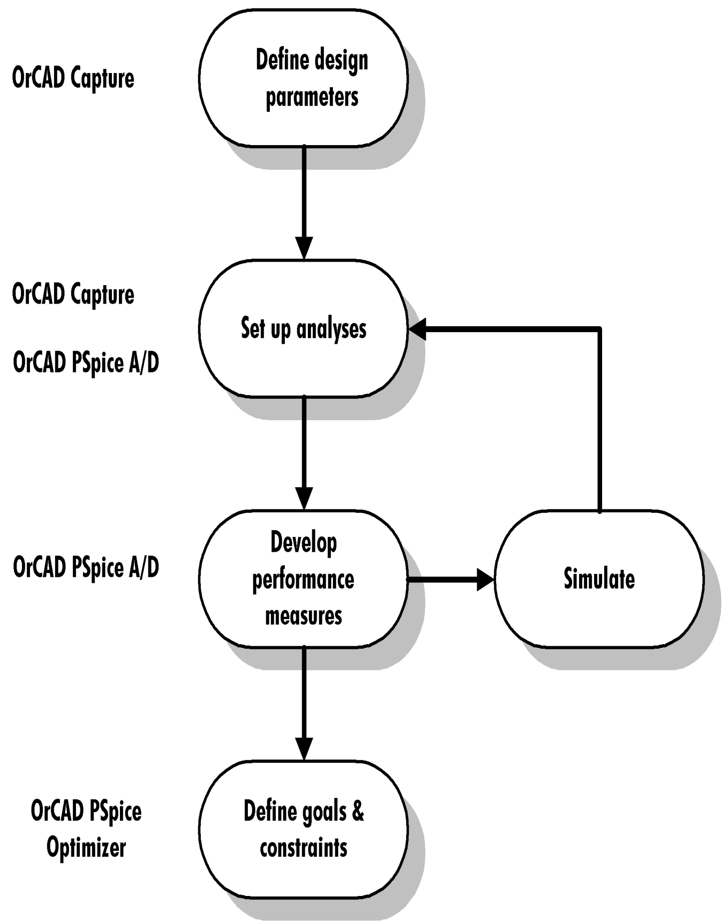


Figure 4 *Design flow for setting up the optimization.*

Defining design parameters

To define parameters for optimization, you must:

- Identify the parameters to adjust for optimization and assign a unique name to each one.
- Set up each parameter as a global optimization parameter using Capture.
- Select which components in the design are affected by the parameter and, for each component, replace its value (e.g., the value of its VALUE property) with an expression that includes the parameter name.

You can also define optimization parameters in the PSpice Optimizer by selecting Parameters from the Edit Menu. See Adding and editing parameters on page 3-63 for more information.

To prepare the diode design example for optimization, you need to parameterize the value of R1 and specify its optimization properties.

To set up the value of R1 as a parameter named R1Val

1. From Capture's PSpice menu, choose Place Optimizer Parameters to place an instance of the OPTPARAM part (from SPECIAL.OLB).
2. Double-click on the OPTPARAM part, then click the User Properties button.
3. Set R1val properties as shown in the Optimizer Parameters dialog box.

The parameter settings are:

Name= R1Val

Initial Value= 5k

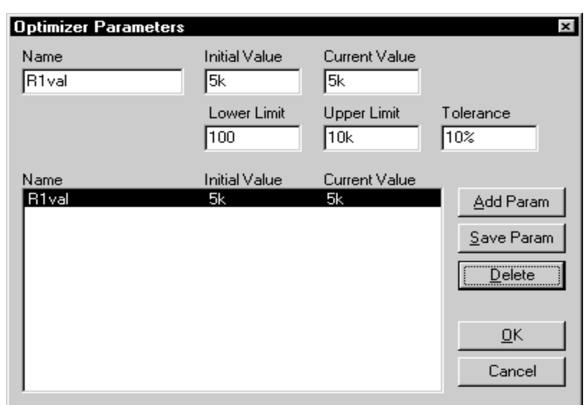
Current Value= 5k

Lower Limit= 100

Upper Limit= 10k

Tolerance = 10%

Later, in Capture, when you select Run Optimizer from the PSpice menu, the parameters specified with the OPTPARAM part are loaded into the PSpice Optimizer and displayed in its main window.



4. Click OK.

5 Double-click the 1 k label for R1 and enter `{R1val}` to parameterize the value of R1.

6 Click OK.

Setting up goals and constraints

Before you can evaluate and improve the circuit's performance, answer these questions:

- What operating characteristics do I want to measure?
- How do the parameters affect the operating characteristics?

After you've answered these questions, you are ready to:

- Set up the analyses needed to evaluate the performance measures.
- Develop the performance measure algorithms.
- Fully define the goals and constraints in terms of these performance measures and analyses.

Setting up analyses for each goal and constraint

For each specification, you must define an analysis type: AC, DC, or transient. This is the analysis that PSpice will run in order to generate results that will be used by the PSpice Optimizer to measure performance.

For the diode design example, you want to monitor the value of I(D1) at a fixed input voltage of 5 V while the optimization parameter, R1val, is varied. This means setting up a single-point voltage sweep.

To set up a single-point voltage sweep analysis at 5 volts

1 From Capture's PSpice menu, choose New Simulation Profile, then enter a name (`DC Sweep`) for the profile.

The Simulation Settings dialog box appears.

2 Under Analysis type, select DC Sweep.

3 To fix the voltage of V1, fill in the DC Sweep dialog box as shown.

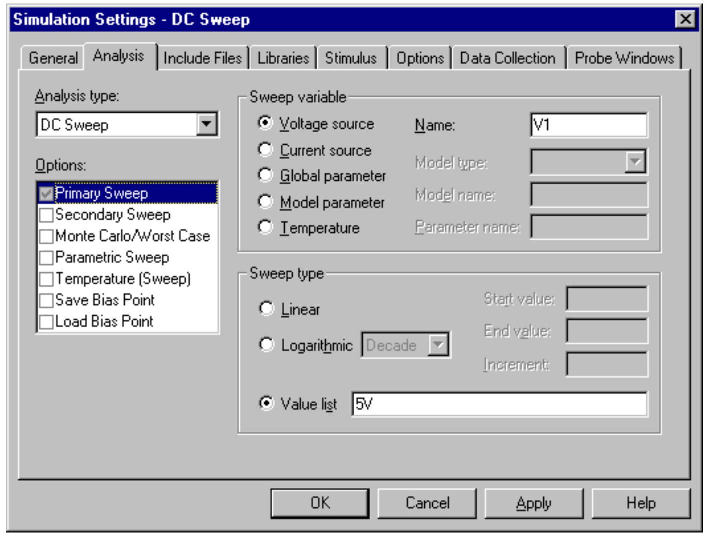
The DC Sweep settings are:

Swept Var Type = Voltage Source

Sweep Type= Value List

Name = V1

Values= 5v



4 Click OK to save the profile.

See Evaluation on page 1-27 and the sections that follow for definitions of trace function, goal function, and PSpice Optimizer expression.

Developing performance measures

To measure performance you must define an evaluation algorithm for each specification. There are three alternatives:

- Trace function (for single-point simulations)
- Goal function
- PSpice Optimizer expression

OrCAD supplies standard goal functions for AC, DC, and transient analyses in the file PSPICE.PRB. (This file resides in the OrCAD root/PSpice directory.) You can add goal functions to this file, or create a local .PRB file for use with a specific design.

See **Chapter 7,** Tutorial: Using constrained optimization (MOS amplifier) for examples of Probe goal functions used to evaluate performance.

Refer to your *PSpice A/D User’s Guide* for instructions on creating goal functions, and for a description of the global and local .PRB files.

When the evaluation is anything other than a single-point simulation or PSpice Optimizer expression, you must develop goal functions to derive values from the simulation results. Developing goal functions is an iterative process that involves writing the goal function, simulating the design, and testing the goal functions against actual results to make sure you are measuring the waveform characteristics you intended.

A goal function is not required for the diode design example. You are examining the trace of R1Val versus

I(D1) which shows the relationship between the value of R1 and the diode forward current. Because only a single point on the curve is of interest, the trace function, I(D1), is the appropriate evaluation.

Defining specifications: goals and constraints

Now that you've completed the preliminary groundwork, you are ready to define the properties for goals and constraints. So far, you have performed all steps in Capture. To finalize setup, you must specify the goal for the diode design example using the PSpice Optimizer.

To define the design goal, Id1, for the diode design example

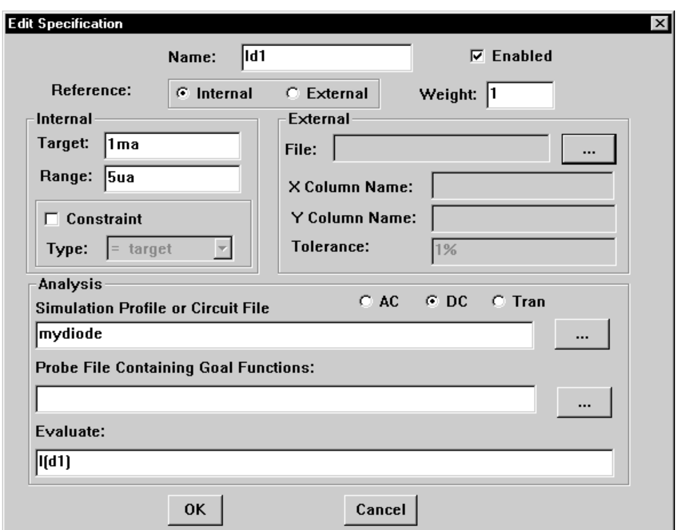
1 From Capture's PSpice menu, choose Run Optimizer to start the PSpice Optimizer.

  The PSpice Optimizer window appears showing the parameter R1val that you defined using the OPTPARAM part in the your design.

2 From PSpice Optimizer's Edit menu, choose Specifications.

3 In the Specifications dialog box, click Add.

4 Enter Id1 properties, as shown below.



The goal specification settings are:

Name = Id1

Target = 1ma

Range = 5ua

Analysis = DC

Circuit File = mydiode

Evaluate = I(d1)

The PSpice Optimizer appropriately defaults to the internal specification setting shown in the Reference control.

Phase Three: Running an optimization

Now that you have defined the parameters, specifications, and evaluations for the design, you are ready to optimize, adjust, and finalize your design.

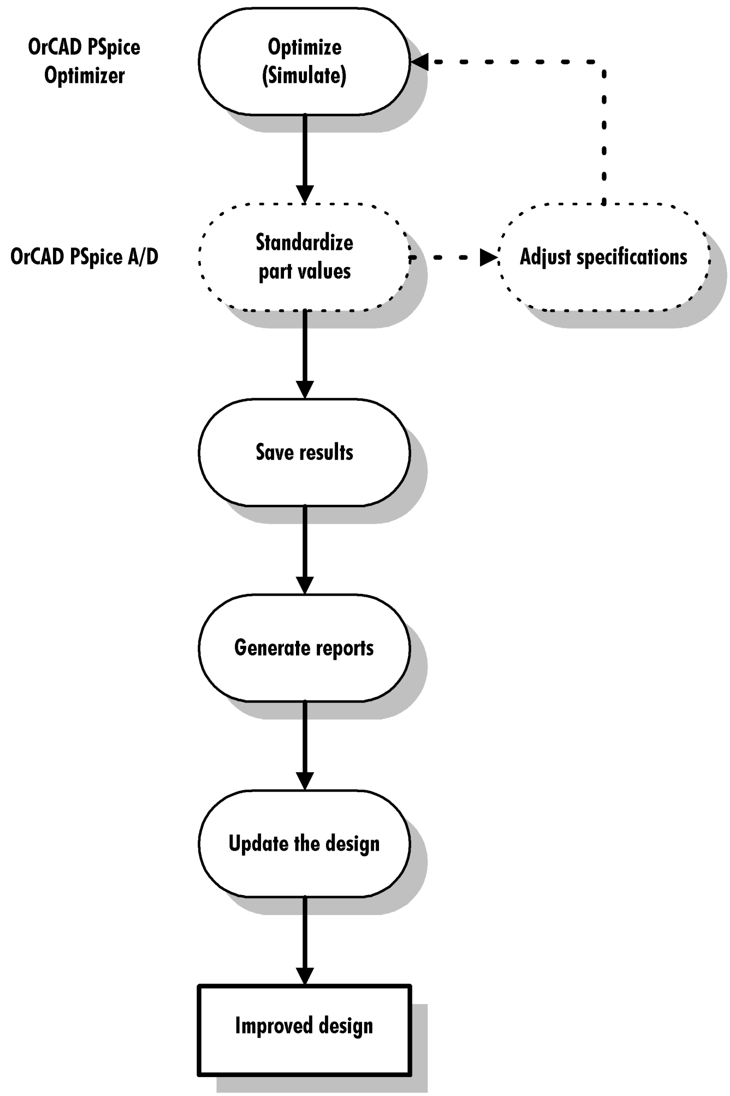


Figure 5 *Design flow for running an optimization.*

Running the PSpice Optimizer

You can use the PSpice Optimizer to:

- Optimize the circuit to completion (from the Tune menu, select Auto).
- Evaluate performance for a single set of parameter values (from the Tune menu, select Update Performance).

  This is useful when initially validating the circuit or when restarting optimization after adjusting parameters or specifications.
- Compute derivatives of each specification with respect to each parameter (from the Tune menu, select Update Derivatives).

  This is useful when exploring design tradeoffs by tweaking parameter and specification values.

When you select Auto from the Tune menu, the PSpice Optimizer automatically computes the derivatives for each specification with respect to each parameter (Figure 6). Using the derivatives, the optimizer determines the direction in which to vary the parameters, and changes parameter values accordingly until it achieves a reduction in the overall error. After updating the parameters, the optimizer computes new derivatives and repeats the process until one of the following occurs:

- Specifications are met (success).
- No more progress can be made (failure).
- You manually interrupt the process.

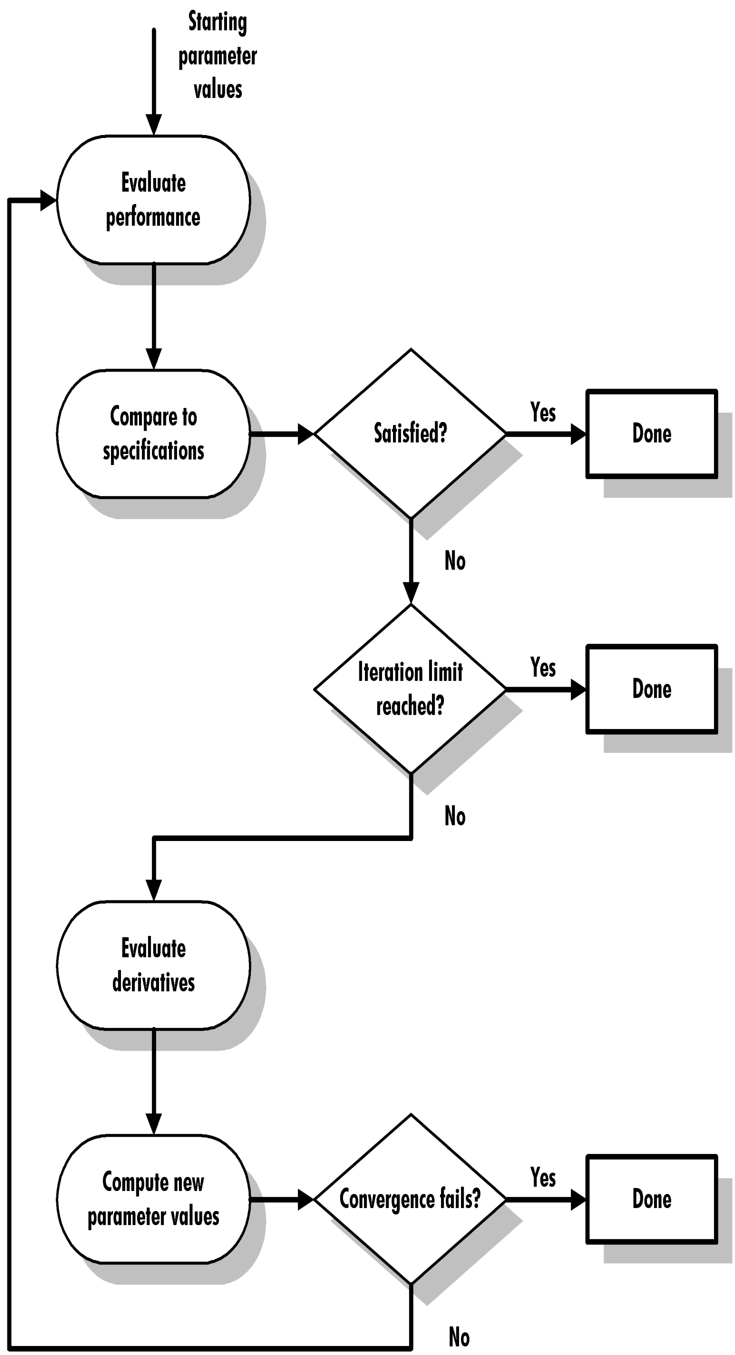


Figure 6 *PSpice Optimizer automatic optimization process.*

To start optimizing the diode design

1 From the Tune menu, select Auto and click Start.

The PSpice Optimizer performs several simulations. For each iteration of the parameter values, the optimizer calculates overall performance and graphically displays the results. The optimizer also calculates the value of the trace function, I(d1), and displays the new value in the specifications area of the PSpice Optimizer window. After three iterations, the optimizer should converge on a solution of 4.131 k, as shown in Figure 7.

In the diode example, the derivative at R1=5 k is –1.62 x 10 $^{-7}$, or

$$\frac{\partial}{\partial R1}(Id) = -1.62\times10^{-7}$$

This indicates that a 1 ohm increase in R1 will produce a decrease of 0.162 uA in Id1. To verify this, try reducing the value of R1 by 100 ohms (to 4.9 kohm) and simulate. This increases the diode current by 16.2 uA and agrees with the derivative information.

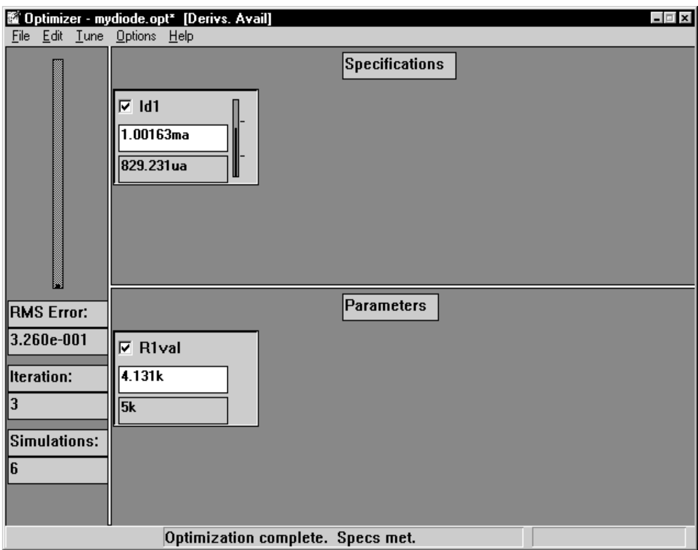


Figure 7 *Optimization results for the diode design example.*

See The PSpice Optimizer Window on page 3-58 for a complete description of the window elements and how you can interact with them.

Adding a constraint and rerunning the PSpice Optimizer

So far, you have optimized for a single goal, Id1. Now suppose you want to add a condition, or *constraint*, on the power dissipated in resistor R1.

Constraints are defined like goals (using the Edit Specification dialog box) with two additions. In the Internal frame, you must:

- Select the Constraint check box.
- Choose the constraint type (>= target, = target, or <= target).

  The constraint type specifies the required relation between what is evaluated (as defined in the Evaluate text box) and the target value (defined in the Target text box).

To define the constraint for power dissipation in R1

The power dissipated in R1 must be less than or equal to 4mW±400 uW. Define the constraint by doing the following:

1. From PSpice Optimizer’s Edit menu, choose Specifications.
2. In the Specifications dialog box, click Add.
3. Enter the power (Pc) constraint properties, as shown below.

The constraint specification settings are:

Name = Pc

Target = 4mW

Range = 400uW

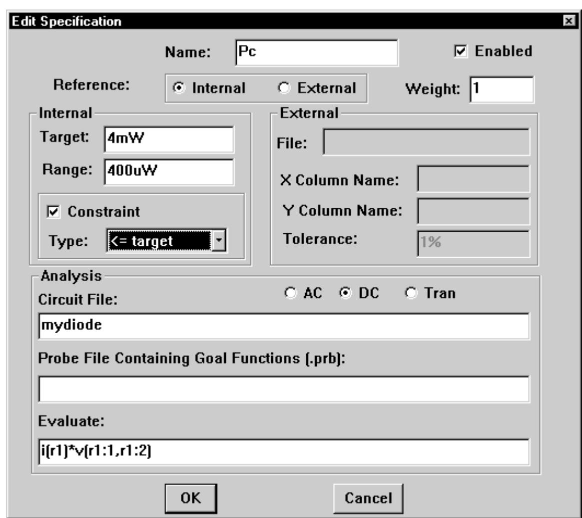
Constraint selected

Type = <=target

Analysis = DC

Circuit File = mydiode

Evaluate = i(r1)*v(r1:1,r1:2)



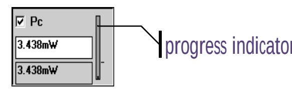
The Evaluate text box contains the expression for measuring dissipated power. For each iteration, PSpice A/D will compute dissipated power by taking the product of the voltage across the resistance and the current through it.

4 To calculate the performance of the design for initial parameter and specification values only (one iteration):

a From the Edit menu, choose Reset Values.

b From the Tune menu, choose Update Performance.

Note the appearance of the progress indicator in the Pc box. Since Pc is a *less than or equal to* constraint, the progress indicator has a tick mark 1/4 of the way up. The vertical bar within the indicator is below the tick mark; this means that the constraint is currently satisfied.



See Progress indicator on page 3-60 for more on the different kinds of progress indicators and how to interpret them.

5 From the Tune menu, choose Auto and click Start to start optimization.

After a number of iterations, the optimization ends *without* satisfying the goal.

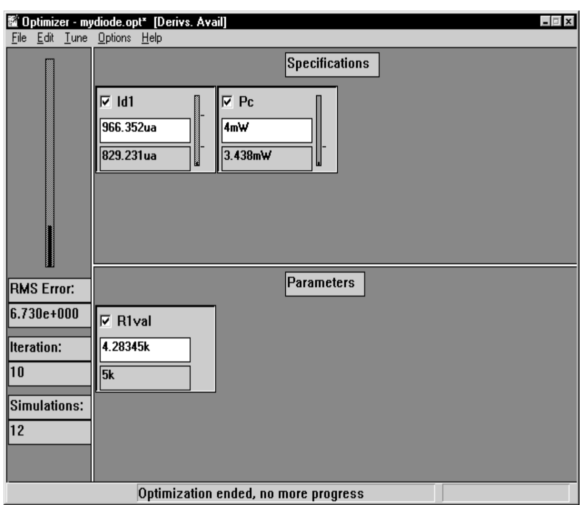


Figure 8 *Results after adding the power constraint.*

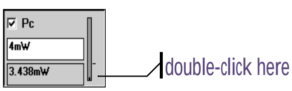
Note that the power dissipated in R1 is exactly equal to the target value of the constraint (4 mW). In this example there is no feasible solution to the problem.

However, the PSpice Optimizer found the lowest value for Id1 which does not violate the constraint.

Changing the constraint and rerunning the PSpice Optimizer

You can examine the effect the Pc constraint has on performance by changing its constraint type so the power dissipation in the resistor must be *greater than or equal* to 4mW.

To change the Pc constraint type to “greater than or equal”



1. In PSpice Optimizer, double-click the lower right-hand corner of the Pc box.
2. In the Edit Specifications dialog box, change Type to >= target.
3. Click OK.

To run the optimization with the modified constraint

1. Test performance with the updated constraint:
    a. From the Edit menu, choose Reset Values.
    b. From the Tune menu, choose Update Performance.

    The Pc constraint is initially violated because the power dissipation is less than 4 mW.
2. From the Tune menu, choose Auto and click Start to start optimization.

    The PSpice Optimizer finds a solution which satisfies both the goal (current of 1 mA) and the constraint (power dissipated in the resistor greater than 4 mW). Figure 9 shows the results.

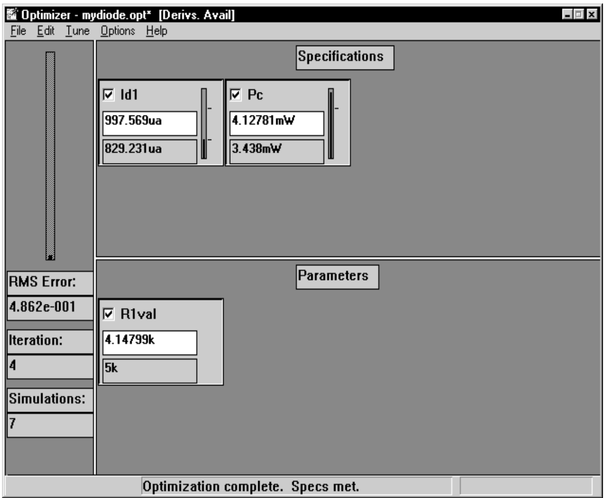


Figure 9 *Results after changing the constraint type.*

Using standard component values

See Using standard part values on page 3-82 for more information, including how the PSpice Optimizer uses tolerances and limits when standardizing values.

When an optimization completes successfully, the optimizer displays the new parameter values in the PSpice Optimizer window. However, each calculated value might not correspond to an actual value that is available with off-the-shelf components. For example, resistors are not readily available in all possible values.

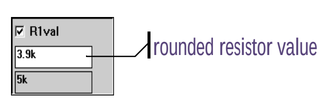
You can use the PSpice Optimizer to select standard component values. The optimizer either:

- rounds to the nearest value, or
- computes values based on the most recent optimization run.

To round to the nearest standard component value

1 From the Edit menu, select Round Nearest.

R1val's current value changes to 3.9 k in accordance with the specified 10% tolerance.



Producing reports

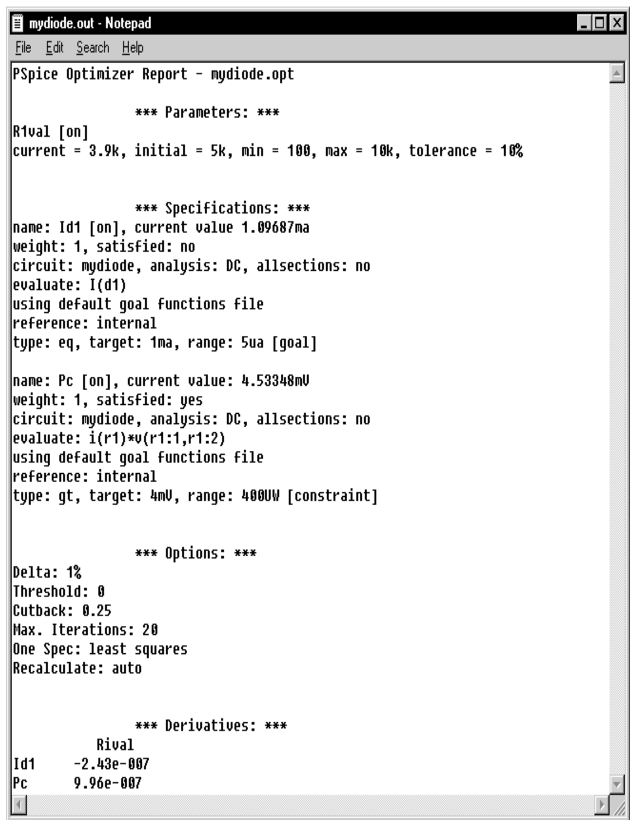
You can use the PSpice Optimizer to generate a report summarizing:

- current settings for parameter, specification, and program options.
- calculated derivatives and Lagrange multipliers.

To generate a summary report

1 From the File menu, choose Report.

The PSpice Optimizer saves the final results to MYDIODE.OPT as shown in Figure 10.




```
PSpice Optimizer Report - mydiode.opt

                *** Parameters: ***
R1val [on]
current = 3.9k, initial = 5k, min = 100, max = 10k, tolerance = 10%


                *** Specifications: ***
name: Id1 [on], current value 1.09687ma
weight: 1, satisfied: no
circuit: mydiode, analysis: DC, allsections: no
evaluate: I(d1)
using default goal functions file
reference: internal
type: eq, target: 1ma, range: 5ua [goal]

name: Pc [on], current value: 4.53348mV
weight: 1, satisfied: yes
circuit: mydiode, analysis: DC, allsections: no
evaluate: i(r1)*v(r1:1,r1:2)
using default goal functions file
reference: internal
type: gt, target: 4mV, range: 400UW [constraint]


                *** Options: ***
Delta: 1%
Threshold: 0
Cutback: 0.25
Max. Iterations: 20
One Spec: least squares
Recalculate: auto


                *** Derivatives: ***
           Rival
Id1     -2.43e-007
Pc      9.96e-007
```

Figure 10 *Report summary for the diode optimization.*

Saving results

When you have finished optimizing, you can save all of the optimizer data, including the current values for all parameters and specifications.

To save the optimizer data for the diode design

1 From the File menu, choose Save.

The PSpice Optimizer updates the MYDIODE.OPT file. All of the options settings are also saved.

Updating the schematic

Having completed the optimization, you can update the data in the schematic file to include the optimized parameter values.

To update the diode schematic with the current parameter values

1 From the Edit menu, select Update Schematic.

Recall that R1Val was initially set at 5.0 k in the schematic file. When you select Update Schematic, the PSpice Optimizer sends a message to Capture to update the design file. Capture writes the new parameter value of 3.9 k to the OPTPARAM part on the schematic (as the current value).

Figure 11 shows the updated schematic for the diode design.

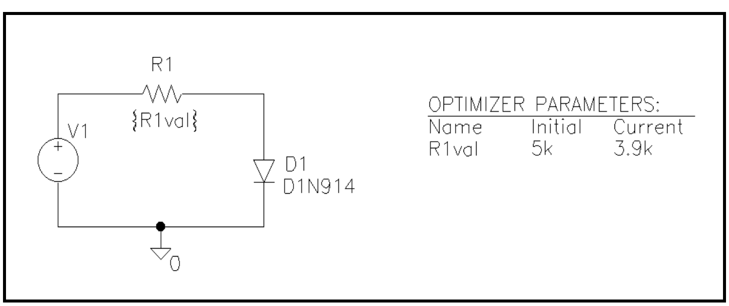


Figure 11 *Updated diode schematic.*

Using the PSpice Optimizer

3

Chapter overview

This chapter describes in general terms how to complete any task using the PSpice Optimizer, including:

- How to activate the PSpice Optimizer and load a design, *page* **3-55**.
- How to interact with the PSpice Optimizer window, *page* **3-58**.
- How to add and edit optimization parameters, *page* **3-63**.
- How to add and edit goals and constraints, *page* **3-66**.
- How to measure and optimize performance, *page* **3-71**.
- How to explore design tradeoffs, *page* **3-74**.
- How to generate result summaries including Lagrange multipliers and derivative values, *page* **3-79**.

- How to finalize the design: standardize component values, save results, and back-annotate the schematic, *page* 3-79.

Starting and loading the PSpice Optimizer

This section describes how to:

- Start the PSpice Optimizer.
- Set special startup options.
- Load a design.

Starting the PSpice Optimizer

Start the PSpice Optimizer program either from:

- Capture

or

- The Windows Start menu

From Capture

To start the PSpice Optimizer from within Capture

1 From Capture's PSpice menu, choose Run Optimizer.

If you have an active schematic loaded in the schematic editor when you choose Run Optimizer, any optimization parameters defined with the OPTPARAM part and any existing setup information contained in the corresponding optimization file (.OPT) are automatically loaded into the PSpice Optimizer.

If no design is active, the PSpice Optimizer starts without an optimization setup. Instead, you must load an optimization file directly into the optimizer as described in Loading a different optimization file on page 3-57.

From the Windows Start menu

From the Start menu, there is a program folder which contains Windows shortcuts for all installed OrCAD programs, including the PSpice Optimizer.

To start the PSpice Optimizer from the Windows Start menu

From the Windows Start menu, select the OrCAD program folder and then the PSpice Optimizer shortcut to start optimizer.

The optimizer starts without an optimization setup. See Loading a different optimization file on page 3-57 for further instructions.

Changing startup options

Because you can start the PSpice Optimizer from either Capture or from the Windows Start menu, we recommend that you change the command line definition as follows:

- In Windows Explorer, change the Target text box in the Optimizer's Properties dialog box (click once on the PSpice Optimizer shortcut, then, from the File menu, choose Properties and click the Shortcut tab).

The PSpice Optimizer supports two command line options, which are used to:

- Start the optimizer with an initialization file other than the default (PSPICE.INI).
- Automatically load an optimization file (.OPT) after startup.

You can add one or both options to the command line.

To change the initialization file used by the PSpice Optimizer

In the Optimizer command line, use the -i option as follows:

`OPTIMIZE -i` *initialization_file_name*

To automatically load an optimization file after startup

In the PSpice Optimizer command line, add the name of the optimization file as follows:

`OPTIMIZE` *optimization_file_name*

The following command line example shows how to start the optimizer *at all times* with an initialization file named MYINIT.INI and an optimization file named MYDESIGN.OPT:

```
OPTIMIZE mydesign.opt -i myinit.ini
```

Loading a different optimization file

Once you have started the PSpice Optimizer, you can change to a new or different optimization file at any time.

To start a new optimization

1. From the File menu, choose New.

To load an existing optimization setup

1. From the File menu, choose Open.
2. Locate and select the appropriate optimization file.

The PSpice Optimizer Window

The PSpice Optimizer window contains three areas:

- Specifications area
- Parameters area
- Error gauge area

Figure 12 illustrates their position in the window.

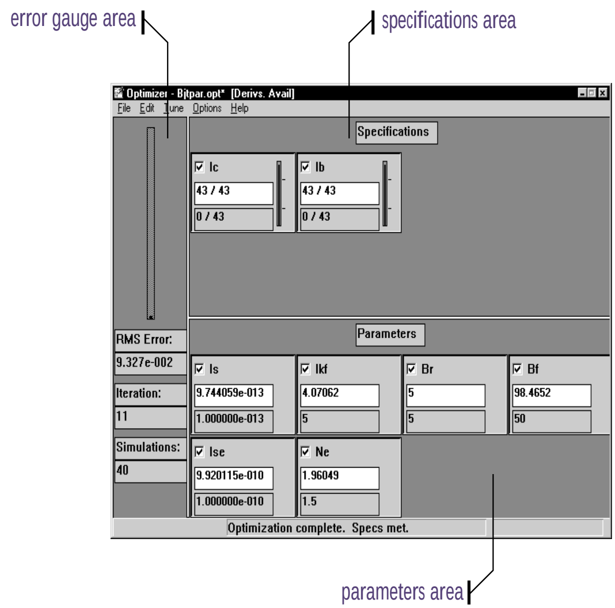


Figure 12 *The PSpice Optimizer window.*

Specifications area

The specifications area can show up to eight specification boxes where each box represents either a goal or a constraint. Figure 13 illustrates the fields contained in each box.

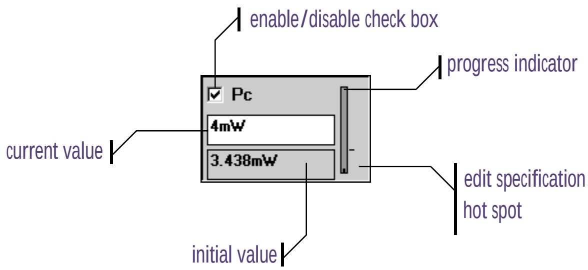


Figure 13 *Example of a specification box.*

The contents of the box varies depending on the source for the specifications—either internal or external. The following sections describe the differences in the initial and current value fields, and the progress indicator.

For more information on the enable/disable check box, see Excluding parameters and specifications from optimization on page 3-78. For more information on the edit hot spot, see Selecting a specification to edit on page 3-71.

Internal specifications

Initial value The initial value field displays a single performance measure that the PSpice Optimizer sets when you start an optimization. The optimizer derives this value from the initial optimization parameter values you defined in the schematic (or the optimizer).

Current value The current value field displays the performance measure that corresponds to the current parameter values. Current values are updated each time an optimization iteration makes progress. When the current value satisfies the specification (that is, the current value is within the allowed range of the target) then the progress indicator turns from red to green, and the PSpice Optimizer considers the specification satisfied.

If derivative data is available, you can change the value in this field to explore how parameter values might change. See Testing performance when changing current values on page 3-74 for more information.

Progress indicator As mentioned above, the progress indicator shows red while the specification is violated, and changes to green when the specification is satisfied.

You can monitor progress as the optimization runs by watching the progress indicator and observing the height of the vertical black bar relative to the tick mark(s) to the right. The number and relative position of the tick mark(s) varies depending on the type of specification:

- Two tick marks for a goal or equality constraint denote the acceptable range around the target value.
- A single tick mark one-quarter of the way up denotes a *less than or equal to* constraint.
- A single mark three-quarters of the way up denotes a *greater than or equal to* constraint.

External specifications

Initial value The initial value field is used in a different way from an internal specification: the field contains a pair of numbers separated by a '/' character. The first number is the number of subgoals in the external specification that are satisfied. The second number is the total number of subgoals in the external specification.

Current value The current value field is used in a different way from an internal specification: the field contains a pair of numbers as described above for initial value.

Progress indicator The progress indicator shows the fraction of subgoals that are satisfied. The indicator turns from red to green when all of the subgoals are satisfied.

For example: If an external specification has 20 subgoals and five are satisfied, the current value field reads `5/20` and the bar in the progress indicator rises to the one-quarter mark.

Parameters area

The parameters area can show up to eight parameter boxes. Figure 13 illustrates the fields contained in each box.

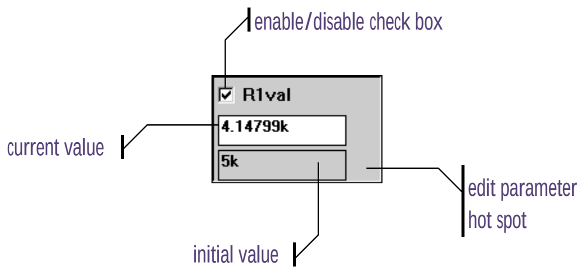


Figure 14 *Example of a parameter box.*

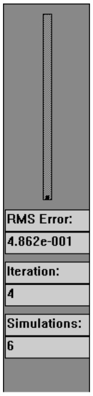
The following sections explain the initial and current value fields in the parameter box.

For more information on the enable/disable check box, see Excluding parameters and specifications from optimization on page 3-78. For more information on the edit hot spot, see Selecting a parameter to edit on page 3-65.

Initial value The initial value field displays a single value taken from the parameter specification that you set up either in Capture (using the OPTPARAM part) or in the PSpice Optimizer (from the Edit menu, choose Parameters).

Current value Initially, the current value is taken from the parameter specification you set up in either the schematic or the PSpice Optimizer. With each iteration of an optimization, the optimizer updates the current value field with the new set of parameter values.

If derivative data is available, you can change the value in this field to explore how specification values might change. See Testing performance when changing current values on page 3-74 for more information.



See Target value scaling on page 4-97 for a discussion of normalization and scaling.

Error gauge area

The error gauge area has an error indicator and shows information reflecting the overall condition of the optimization.

The error gauge shows how far the goals are from their target values. Initially, the indicator displays 100%. As the optimization proceeds, the error falls as the goals approach their target values. The PSpice Optimizer displays the RMS of the normalized value of the goals in the RMS Error field.

If all of the specifications meet the target values exactly, the error is zero. If the optimization succeeds by finding a solution that works within the set ranges, but does not meet the target values exactly, the error represents the combined error of all of the specifications.

Note *The RMS error reflects only those goals that are enabled. The error does not include contributions from constraints, nor does it reflect any specifications that are disabled.*

Adding and editing parameters

This section describes how to create and change optimization parameters using the PSpice Optimizer. This means you can also edit the properties of parameters that you defined in a schematic.

Note *You are limited to eight parameters per optimization file. This limit includes parameters that you have defined but disabled for a given run.*

Adding a parameter

There are two ways to add a parameter using the PSpice Optimizer:

- Create the parameter from scratch.
- Copy an existing parameter and change its name.

You can also define parameters in Capture using the OPTPARAM part. See Defining design parameters on page 2-38 for an example.

To create a parameter from scratch

1 From PSpice Optimizer's Edit menu, choose Parameters.

Any optimization parameters that you have already established in the schematic appear in the selection list.

2 In the Parameters dialog box, either:

- click Add

or

- double-click the blank line following the parameter list

3 In the Edit Parameter dialog box, set the controls as described in **Table 3**.

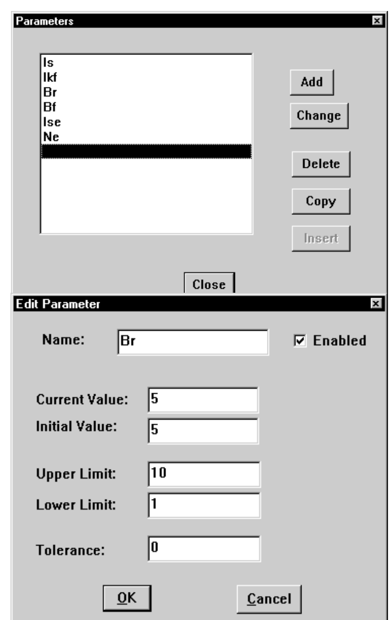


Table 3 *Edit parameter dialog box controls.*

Control Name	Meaning
Name	Parameter name; for a new parameter, double-click <<new>> and enter a text string that is unique to the current optimization file.
Current Value	Latest parameter value; for a new parameter, set Current Value equal to Initial Value.
Initial Value	Starting value for the parameter.
Upper Limit	Highest allowable value for the parameter.
Lower Limit	Lowest allowable value for the parameter.
Tolerance	Tolerance level (1%, 5%, etc.) to use when standardizing component values.
Enabled	When selected, includes the parameter in the next optimization run. If cleared, excludes the parameter.

To create a parameter based on an existing parameter

1. From PSpice Optimizer's Edit menu, choose Parameters.
2. Select the parameter that you want to copy and click Copy.

   Notice that the Insert button is now enabled (no longer grayed out)
3. Click Insert to add a copy of the selected parameter to the parameter list.

   The optimizer inserts the new parameter immediately above the highlighted parameter.
4. Select the new parameter in the list and click Change.
5. Change the Name text box to a unique name, and change any other controls as needed.

6 When finished, click OK to return to the Parameters dialog box.

Note *You cannot copy a parameter description from one optimization file to another. Instead, save an entire optimization file under a new name, then edit the new version as needed.*

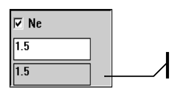
Selecting a parameter to edit

You can change the properties of an existing parameter at any time using the Edit Parameter dialog box.

To display the Edit Parameter dialog box for a parameter

1 Do one of the following:

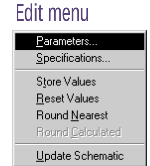
- In the parameters area of the PSpice Optimizer window, double-click the lower right-hand corner of the box for the parameter you want to edit.



or

- From the Edit menu, choose Parameters, then select the parameter you want to edit, and click Change.

Edit menu



Adding and editing specifications

This section describes how to create and change specifications—that is, goals and constraints—using the PSpice Optimizer.

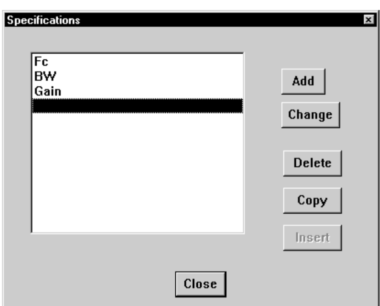
Note *You are limited to eight specifications per optimization file. This limit includes specifications that you have defined but disabled for a given run.*

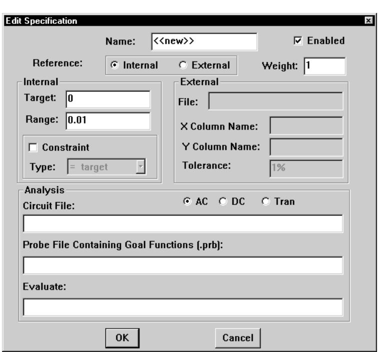
Adding a specification

There are two ways to add a specification using the PSpice Optimizer:

- Create the specification from scratch.
- Copy an existing specification and change its name.

To create a specification from scratch





1 From PSpice Optimizer's Edit menu, choose Specifications.

2 In the Specifications dialog box, either:

- click Add

or

- double-click the blank line following the specification list

3 In the Edit Specification dialog box, set the controls as described in **Table 4**.

Table 4 *Edit specification dialog box controls.*

Control name	Meaning
Name	Specification name; for a new specification, double-click <<new>> and enter a text string that is unique to the current optimization file.
Reference	• Select Internal when defining the specification's target value and range in this dialog box. • Select External when defining the specification's measurement data using a file (e.g., for curve fitting).
Weight	Relative weight of the specification. To give a specification more weight, assign a number that is higher than that for the other specifications.
Enabled	When enabled, includes the specification in the next run. If cleared, excludes the specification.
Internal specification	
Target	Ideal value for the specification.
Range	Delta applied to Target, defining the acceptable range of values; i.e., Target ± Range. For example: If Target=10 and Range=2, then the PSpice Optimizer accepts values from 9 to 11.
Constraint	When enabled, defines the specification as a constraint (not a goal).
Type	Defines whether constraint values must equal the target value, be less than or equal to the target value, or be greater than or equal to the target value.

External Specification

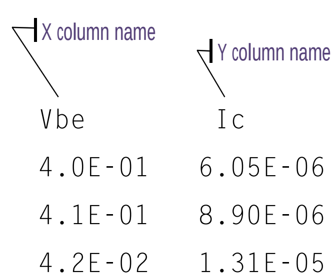


Figure 15 *Sample format for an external specification.*

Table 4 *Edit specification dialog box controls. (continued)*

Control name	Meaning
File	Name of the file that contains the measured data.
X Column Name	Heading for the data column in File containing the independent (X) values.
Y Column Name	Heading for the data column in File containing the dependent (Y) values
Tolerance	Tolerance value used when standardizing component values. Syntax: < 0 ≤ *integer value* ≤ 100 >%
Analysis Settings	
(Analysis Type)	Kind of analysis used for simulation-based evaluations. • Select AC for AC sweep analysis. • Select DC for a DC sweep analysis. • Select Tran for a transient analysis.
Circuit File	Set to either: • the name of the circuit file PSpice uses for simulation, or • leave blank if the Evaluate text box contains a PSpice Optimizer expression.
Evaluate*	Trace function, Probe goal function, or PSpice Optimizer expression used to measure performance.

* See Defining an evaluation for an external specification on page 3-70 for the purpose and use of the '!' part in the Evaluate edit control.

To create a specification based on an existing specification

1. From PSpice Optimizer's Edit menu, choose Specifications.
2. Select the specification that you want to copy and click Copy.

   Notice that the Insert button is now enabled (no longer grayed out)
3. Click Insert to add a copy of the selected specification to the parameter list.

   The optimizer inserts the new specifications immediately above the highlighted specification.
4. Select the new specification in the list and click Change.
5. Change the Name text box to a unique name, and change any other controls as needed.
6. When finished, click OK to return to the Specifications dialog box.

Note *You cannot copy a specification description from one optimization file to another. Instead, save an entire optimization file under a new name, then edit the new version as needed.*

Defining an evaluation for an external specification

After substitution, the PSpice Optimizer does one of the following depending on the kind of evaluation:

- For a waveform analysis goal function, the optimizer sends the substituted goal function to PSpice A/D for evaluation.
- For a PSpice Optimizer expression, the optimizer evaluates the expression directly.

See **Chapter 8, Tutorial: Fitting model data (bipolar transistor)** for another example using this technique.

When defining evaluations for external specifications, use the '!' part within the evaluation as a placeholder for data in the external file. For each of the subgoals in the external file, the PSpice Optimizer first replaces the '!' character with the X column data value (defined in the X Column Name text box in the Edit Specification dialog box), and then proceeds with evaluation. This approach allows a waveform analysis goal function or PSpice Optimizer expression to *track* the independent data value.

To set up an evaluation for an external specification

1. Create the goal function or PSpice Optimizer expression.
2. In the Edit Specification dialog box, enter the evaluation in the Evaluate text box as follows:

   Substitute the '!' character for every X value argument; i.e., wherever a measured subgoal value should appear.

   For example: Suppose that you want to fit a set of data measured at different values of Vtest (2.0, 2.5, and 3.0 volts), using the goal function:

   `YatX(V(out),` *`x_value`*`)`

   to measure the output of the design. To use this with an external specification, replace *x_value* with the '!' character and enter the evaluation:

   ```
   YatX(V(out), !)
   ```

   into the Evaluate text box for the specification.

   As the fitting process proceeds, the PSpice Optimizer passes the following goal functions (one for each measured data point) to PSpice A/D:

   ```
   YatX(V(out), 2.0)
   YatX(V(out), 2.5)
   YatX(V(out), 3.0)
   ```

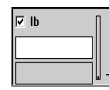
Selecting a specification to edit

You can change the properties of an existing specification at any time using the Edit Specification dialog box.

To display the Edit Specification dialog box for a goal or constraint

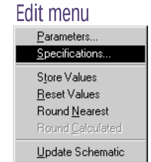
1 Do one of the following:

- In the specifications area of the PSpice Optimizer window, double-click the lower right-hand corner of the box for the specification you want to edit.
- From the Edit menu, choose Specifications, then select the specification you want to edit, and click Change.



double-click here

Edit menu



Measuring and Optimizing Performance

This section describes how to:

- Optimize your design once all of the parameters and specifications are defined.
- Monitor progress using the waveform viewer in PSpice A/D.

Optimizing Your Design

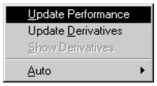
Optimization is a two-stage process:

1 Running one evaluation to ensure that the circuit is valid and that it simulates.

2 Starting the optimization process.

To optimize your design

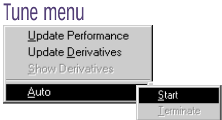
Tune menu



1 From the Tune menu, choose Update Performance.

The PSpice Optimizer measures the design's performance using both the initial and current values for each of the parameters. The optimizer updates the initial and current fields, respectively, for each specification, and sets the progress indicators showing how closely these initial measures match each of the target values.

Tune menu



2 From the Tune menu, choose Auto and click Start.

See Viewing the optimization log on page 3-81 for information on the audit trail the PSpice Optimizer generates when running an optimization.

The PSpice Optimizer computes the derivatives for each specification with respect to each parameter, and uses this information to determine the direction in which to vary the parameters. With each iteration, the optimizer tries parameter changes along the chosen direction and measures performance until it achieves a reduction in the overall error. The optimizer then updates the parameters, calculates new derivatives, and repeats the process until one of the following occurs:

- Specifications are met (success).
- No more progress can be made (failure).
- You manually interrupt the process.

To cancel the optimization run

1 From the Tune menu, choose Auto.

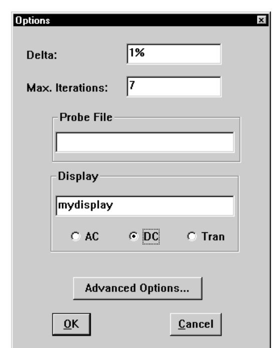
2 Click Terminate.

Graphically monitoring progress

To see how well specifications are approaching the optimization requirements, use the waveform viewer in PSpice A/D to monitor the simulation results from each of the iterations.

To monitor optimization progress

1. From Capture's PSpice menu, choose Simulate Active to simulate the circuit.
2. Create a display configuration.
   a. In PSpice A/D, add the traces (from the Trace menu, choose Add Trace) and modify the axes (from the Plot menu, choose Axis Settings) to appear as you want them to when monitoring intermediate results.
   b. From the Window menu, choose Display Control.
   c. In the New Name text box, enter a name for the waveform display and click Save.
   d. Click Close.
3. Define the waveform display to use when optimizing.
   a. From PSpice Optimizer's Options menu, choose Defaults.
   b. In the Display text box, enter the name of the (Probe) waveform display.
   c. Select the analysis type corresponding to the (Probe) waveform display.
   d. Click OK.



Exploring the effect of parameter and specification changes

The PSpice Optimizer provides three easy ways to examine tradeoffs between goals, constraints, and parameters. You can:

- Tweak parameter values to explore performance effects, or change specification values to see how parameters change, by entering new current values directly into the PSpice Optimizer window.
- Exclude specifications and/or parameters that you previously defined.
- Add new or edit existing parameter definitions and specification descriptions.

Testing performance when changing current values

In the PSpice Optimizer window, you can quickly examine the effects of a small change to either a parameter's current value or a specification's current value.

Recalculation modes When changing current values, you can choose between automatic and manual recalculation of performance.

- With automatic recalculation *selected*, you can change the current value (upper text box) for *one* parameter or a specification value, and see the impact on other values almost immediately.
- With manual recalculation *selected*, you can change the current value for *several* parameters or specifications, and initiate recalculation when you are ready.

Derivative calculations When tweaking values, the PSpice Optimizer does not perform any simulations. Instead, the optimizer requires derivative data and uses this to estimate what will happen when you change a parameter or specification.

See Derivatives on page 4-95 for more information on how the PSpice Optimizer computes derivative data. See Viewing derivatives on page 3-81 for instructions on how to display the latest derivative data.

The PSpice Optimizer calculates derivatives either:

- once, when you choose Update Derivatives from the Tune menu, or
- for each iteration of an optimization run, when you choose Auto from the Tune menu.

Use the first method (Update Derivatives) when you are exploring design tradeoffs.

Note *Because the performance of the design usually depends on the parameters in a highly nonlinear way, the results are typically reliable only for small changes in values. See "Ensuring reliable results when tweaking values" on page **3-77** for the steps you can take for best results.*

Automatically recalculating performance

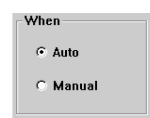
The PSpice Optimizer automatically recalculates performance after each change provided that:

- Automatic recalculation is selected (the default).
- Derivatives are available.

To test performance using automatic recalculation

1. Do the following to enable automatic recalculation:
   a. From the Options menu, choose Recalculate.
   b. In the When frame, choose Auto.
   c. Click Close.

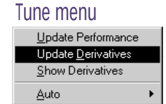
When frame in the Recalculate dialog box



2. From the Tune menu, choose Update Derivatives.

   When derivative calculations are complete, the message `Derivs. Avail` appears in the title bar.

Tune menu



The PSpice Optimizer accepts numerical entries in any format supported by PSpice. Refer to the online *OrCAD PSpice A/D Reference Manual* for a complete list of supported numeric forms.

3 Change the current value for the parameter or specification you want to investigate.

   a Double-click in the current value (upper) field for a parameter or specification.

   b Enter the new value.

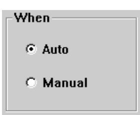
   c Press Enter↵.

   The PSpice Optimizer immediately recalculates the values. When you change parameter values, a small “e” appears next to the progress indicator for each recalculated specification to show that the value is an *estimate* based on derivative data.

Manually recalculating performance

When you want to change multiple values before recalculating performance, disable recalculation. As with automatic recalculation, derivative data is required.

To test performance using manual recalculation



When frame in the Recalculate dialog box

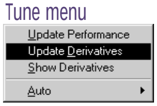
1 Disable automatic recalculation.

   a From the Options menu, choose Recalculate.

   b In the When frame, choose Manual.

   c Click Close.

Tune menu



2 From the Tune menu, choose Update Derivatives.

   When derivative data is calculated, the message `Derivs. Avail` appears in the title bar.

The PSpice Optimizer accepts numerical entries in any format supported by PSpice. Refer to the online *OrCAD PSpice A/D Reference Manual* for a complete list of supported numeric forms.

3 Change the current value for the parameter or specification you want to investigate.

   a Double-click in the current value (upper) field for a parameter or specification.

   b Enter the new value.

   c Press Enter↵.

4 From the Options menu, choose Recalculate to recalculate performance.

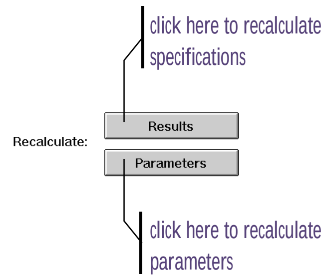
5 Select the kind of recalculation as follows:

- If you entered a new *parameter* value, click Results.

  The PSpice Optimizer recalculates performance. A small 'e' appears next to the progress indicator for each recalculated specification to show that the value is an *estimate* based on derivative data.

- If you entered a new *specification* value, click Parameters.

  The PSpice Optimizer recalculates values for all of the parameters based on the current specification values.



Ensuring reliable results when tweaking values

Because the PSpice Optimizer uses derivative data to estimate what will happen when a parameter or a specification is changed, and because the design usually depends on the parameters in a highly nonlinear way, results are typically reliable only for small changes in values. Once you have significantly changed values, resimulate and recompute the derivatives before adjusting values any further.

To ensure that results are reliable after significant tweaking

1 From the Tune menu, choose Update Performance.

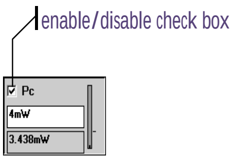
The PSpice Optimizer will run the appropriate simulations and update the specifications.

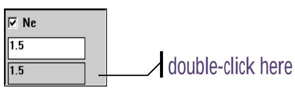
2 From the Tune menu, choose Update Derivatives.

You are now ready to continue exploring the design.

Excluding parameters and specifications from optimization

Every specification and parameter has a check box that you can select to exclude that specification or parameter from the next optimization run.





To exclude a parameter or specification from an optimization

1 Do one of the following:

- In the PSpice Optimizer window, clear the check box (check box should be empty) to the left of the parameter or specification name.
- Double-click the lower right-hand corner of the box for the parameter or specification you want to exclude, clear the Enabled check box (check box should be empty) in the dialog box, and click OK.

Note *When you exclude a specification from optimization, the PSpice Optimizer still re-evaluates its performance when you update the parameters or derivatives. It is also included in the matrix of partial derivatives.*

Testing performance when adding or changing parameters or specifications

Even after running an optimization, you can add new parameters and specifications, or change the properties of existing definitions to see their effect on performance.

To test performance after changing the parameters or specifications

1 From the Edit menu, choose Reset Values.

2 From the Tune menu, choose Update Performance.

3 If you want to run a complete optimization:

a From the Tune menu, choose Auto.

b Click Start.

Saving intermediate values

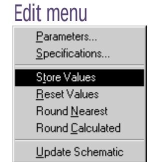
You can save a set of parameter and specification values, and then continue to investigate performance.

To save intermediate values

1 From the Edit menu, choose Store Values.

The PSpice Optimizer copies the current values to the initial values for all specifications and parameters.

Edit menu

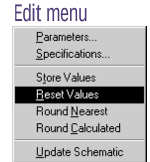


To restore the previous settings

1 From the Edit menu, choose Reset Values.

The PSpice Optimizer copies the initial values to the current value fields.

Edit menu



Viewing result summaries

This section describes how to:

- Generate a report that summarizes the latest run.
- View the current derivative calculations.

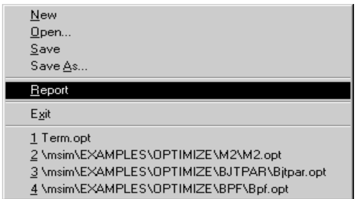
Producing optimization reports

You can produce a report containing:

- Settings for each of the parameters, specifications, and options

- Performance results
- Partial derivatives
- Lagrange multipliers

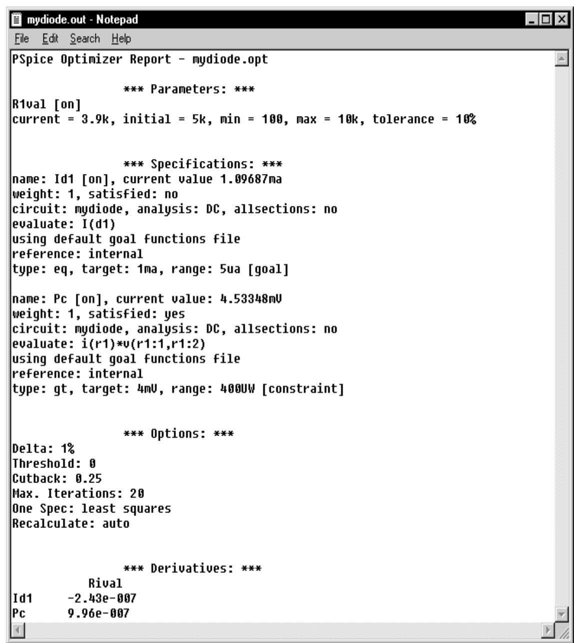
File menu



To generate an optimization report

1 From the File menu, choose Report.

The PSpice Optimizer generates the report and saves it to an ASCII file named *design_name*.OOT. The optimizer also displays the report in the text editor configured for your installation.




```
PSpice Optimizer Report - mydiode.opt

                *** Parameters: ***
R1val [on]
current = 3.9k, initial = 5k, min = 100, max = 10k, tolerance = 10%


                *** Specifications: ***
name: Id1 [on], current value 1.09687ma
weight: 1, satisfied: no
circuit: mydiode, analysis: DC, allsections: no
evaluate: I(d1)
using default goal functions file
reference: internal
type: eq, target: 1ma, range: 5ua [goal]

name: Pc [on], current value: 4.53348mV
weight: 1, satisfied: yes
circuit: mydiode, analysis: DC, allsections: no
evaluate: i(r1)*v(r1:1,r1:2)
using default goal functions file
reference: internal
type: gt, target: 4mV, range: 400UW [constraint]


                *** Options: ***
Delta: 1%
Threshold: 0
Cutback: 0.25
Max. Iterations: 20
One Spec: least squares
Recalculate: auto


                *** Derivatives: ***
           Rival
Id1     -2.43e-007
Pc      9.96e-007
```

Figure 16 *Sample excerpt from a report.*

To print an optimization report

1 Open the report file in a standard text editor (such as Notepad).

2 Use the Print command to print the report.

Viewing the optimization log

The PSpice Optimizer automatically creates an audit trail of optimization progress and saves the information to a file named *design_name*.OLG. You can use this file as a debugging tool when an optimization fails to converge.

To view the optimization log file

1. Activate Notepad or another text editor.
2. Open the log file for browsing.

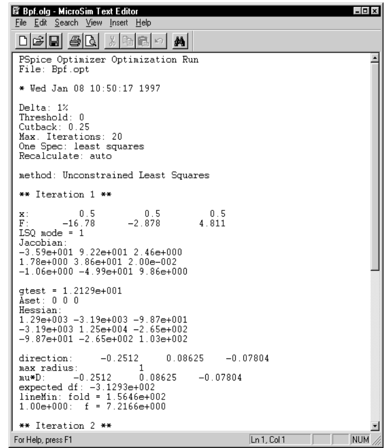


Figure 17 *Sample excerpt from a Log file.*

Viewing derivatives

You can display a matrix showing the most recent derivative data the PSpice Optimizer uses to calculate performance.

To display the derivative data

1. From the Tune menu, choose Show Derivatives.

   Each entry in the matrix represents the partial derivative of one specification (the row label) with respect to one active parameter (the column heading). If a specification is completely independent of a parameter, the derivative is zero.

2. When finished, click Close.

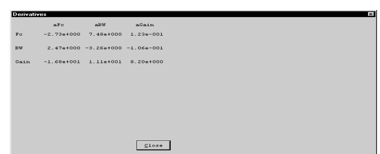


Figure 18 *Sample derivative data.*

Finalizing the design

This section describes how to use the PSpice Optimizer to:

- Standardize part values once the optimization is complete.
- Save results.
- Back-annotate the schematic with the final part and parameter values.

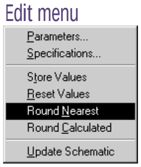
Using standard part values

When optimization parameters directly correspond to part values, you can use the PSpice Optimizer to select standard part values by either:

- rounding to the nearest values, or
- computing values based on the most recent optimization run.

The PSpice Optimizer considers the tolerance specified for the parameters using tables of preferred values for 1%, 5% and 10% tolerance components. Other tolerance values cause the optimizer to use the nearest calculated value for that tolerance. Parameters with zero tolerance are not changed.

Edit menu

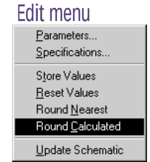


To round component values to the nearest standard values

1 From the Edit menu, choose Round Nearest.

To compute standard component values based on the most recent optimization

Edit menu



From the Edit menu, select Round Calculated.

The PSpice Optimizer replaces the parameter values with the standardized values only if the new values remain within the specified limits. If so, the optimizer automatically calculates new performance values (based on derivative data) using the new parameter values, and displays an ‘e’ in the upper right-hand corner of each specifications area to indicate that the performance measure is an *estimate*.

Saving results

Performance results are not written to the optimization file until you deliberately initiate a save operation.

To save optimization results to the current optimization file

1 From the File menu, choose Save.

By default, the PSpice Optimizer saves the results and the latest optimization settings to a file named *design_name*.OPT. If you started your design from a schematic, this file already exists. If not, the optimizer displays the Save As dialog box and you must enter the name of a new file.

To save optimization results to a new or different file

1 From the File menu, choose Save As.

2 Enter a new file name.

3 Click OK.

If an optimization file with this name already exists, the PSpice Optimizer requests confirmation to overwrite the file.

Updating the design

Once the optimization is complete and you have optionally standardized component values, you can update the underlying schematic with the final parameter values.

Edit menu

To back-annotate the schematic with the latest parameter values

1 From PSpice Optimizer's Edit menu, choose Update Schematic.

The optimizer writes the optimized parameter values to the schematic file. On the schematic, the new values appear in the Current column of the OPTPARAM part.

To use the new parameter values in subsequent simulations run from Capture

1 From Capture's PSpice menu, choose Use Optimized Values.

Understanding optimization principles and options

4

Chapter overview

This chapter explains optimization concepts and how you can influence the outcome of an optimization.

The concepts covered in this chapter include: constrained optimization, function characteristics, convergence and the effect of starting points, and how the PSpice Optimizer computes derivatives and scales target values.

- Default options on page 4-98 describes the options you can set to control derivative calculations, maximum simulation iterations, and the Probe display.
- Advanced options on page 4-101 describes the options you can set to control cutback, threshold, and the method (least squares / minimization) that the PSpice Optimizer uses.

Goals versus constraints

Goals and constraints represent the ideal behavior of a design. In practice, this behavior is often unattainable.

If there is more than one goal, the PSpice Optimizer combines the errors by summing the squares of the normalized values.

For example: A gate cannot achieve zero propagation delay, but the goal of the optimization process might be to come as close as possible to that target value (that is, to reduce the error as much as possible).

When solving problems involving both goals and constraints, the PSpice Optimizer trades off meeting the target values for the goals against violation of the constraints. This means that the error indicator does not always reduce in value for a given iteration.

When setting up an optimization, you must decide which specifications are goals and which are constraints. In many cases, there are several legitimate ways to describe the design.

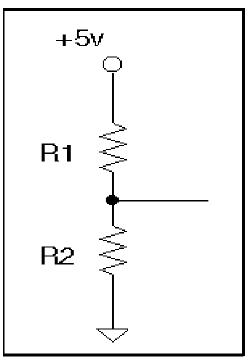


Figure 19 *Resistive terminator circuit.*

For example: Assume you want to design a resistive terminator that produces an output voltage of 3.75 V (± 0.1 V) at the junction of the two resistors, and the Thevenin equivalent resistance of the resistor combination must equal 100 Ω (±1 Ω).

Your objective is to find the best resistor values to meet these two specifications:

- output voltage of 3.75 V (V_e)
- equivalent resistance of 100 Ω (R_e)

You can manually solve this problem using the following simultaneous equations:

$$\frac{5R_2}{R_1 + R_2} = 3.75 \quad \text{and} \quad \frac{R_1 \cdot R_2}{R_1 + R_2} = 100$$

These equations solve to R1 = 133.3 Ω and R2 = 400 Ω, an exact solution.

When using the PSpice Optimizer, you can set up this problem in one of three ways:

- Consider V_e and R_e as equally important; set up both as goals.
- Consider V_e as the most important requirement to meet, even at the expense of R_e; set up V_e as a constraint and R_e as a goal.
- Consider R_e as the most important requirement to meet, even at the expense of V_e; set up R_e as a constraint and V_e as a goal.

Note *Because at least one optimization goal is necessary, the case where both V_e and R_e are constraints is excluded.*

If the problem, like this one, has a solution, the PSpice Optimizer might arrive at the same answer for all three methods. However, most problems do not have a single, exact solution as this one does. For most designs, the result is a compromise that minimizes the goals while not violating the constraints.

Constrained optimization

Many problems in analog circuit optimization are naturally expressed as the minimization of a function representing a goal (e.g., power consumption) which is subject to one or more constraints (e.g., bandwidth). Constraints are typically complicated nonlinear functions of the parameters of the problem, so manual optimization is a difficult task.

Most other analog circuit optimizers implement only unconstrained optimization of a single goal or a sum-of-squares of several goals. To tackle a problem like the problem outlined above, other optimizers must combine the functions for the goals and constraints and then optimize the combination. Unfortunately, this scheme does not differentiate between reduction of the goals and violation of the constraints. In general, constraints are given much greater weight than the goals.

This approach has a number of pitfalls. In particular:

- If a very large value is used for the weight of the constraints, numerical problems occur.
- If a more reasonable value is used, the result is not a true solution of the original problem.
- Using a sequence of weights, and performing a series of minimizations can lead to the true solution, but at the expense of a large increase in optimization time (because of all of the extra evaluations required to solve the intermediate problems).

The PSpice Optimizer implements both constrained and unconstrained minimization algorithms. This means that the optimizer:

- Tackles constrained problems directly and efficiently.
- Calculates Lagrange multipliers for the solution, which provide valuable insight to design tradeoffs.

Types of constraints

Constraints are restrictions placed on potential solutions to optimization problems. The simplest constraints are *bound constraints*—simple limits on the ranges of the parameters (e.g., a resistor whose value has to be at least 100 Ω).

More challenging constraints that frequently arise in analog circuit optimization have dependencies on other characteristics of the design.

Example: Consider optimizing a MOS amplifier cell which must satisfy these specifications:

- Reduce power consumption.
- Make sure gain-bandwidth product of the cell is greater than or equal to some minimum value.

The power consumption of the cell is the *goal* (the characteristic to be minimized) and the gain-bandwidth product is the *inequality constraint*.

The PSpice Optimizer can also handle *equality constraints* where a performance measure is required to be equal to some defined value.

To continue the example, consider the dependence of power consumption and gain-bandwidth product on bias current in one of the amplifier stages. Power consumption is proportional to the bias current, while the gain-bandwidth product is proportional to the square root of the bias current. Bias current must be reduced in order to reduce power consumption. Below some critical bias current the minimum gain-bandwidth requirement will be violated. This critical value is the *constrained minimum* for this problem.

Feasible and infeasible points

The starting point for an optimization can satisfy all the constraints (a *feasible* point) or it can violate one or more of the constraints (an *infeasible* point). Depending on the feasibility of the starting point, the PSpice Optimizer does the following:

- From an infeasible point, it attempts to reduce the goals *and* to reduce the amount by which the constraints are violated.
- From a feasible point, it attempts to reduce the goals *while* keeping the constraints satisfied.

Note *Because the PSpice Optimizer sometimes trades off reduction of the goals against violation of the constraints to make progress, an iteration can produce an infeasible point even though the initial starting point was feasible.*

Active and inactive constraints

An *active* constraint is one which affects the solution of the optimization problem—that is, the solution would probably be different if the constraint were removed. Equality constraints are always active (e.g., a constraint of the form V_{out} = 3.75 V is always active). Inequality constraints are considered active if the solution violates the constraint or if it is equal to the constraint (e.g., a constraint of the form R_{eq} >= 100 is active if R_{eq} is less than or equal to 100 at the solution).

Lagrange multipliers

To view Lagrange multipliers for your design, generate a report by selecting Reports from the File menu and browse the *design_name*.oot report file. See Producing optimization reports on page 3-79 for instructions.

The result of a constrained optimization is typically a compromise between further reduction of the goals and violation of one or more constraints. A set of numbers—the *Lagrange multipliers*—provide valuable information about the tradeoffs between the goals and the constraints.

The PSpice Optimizer calculates Lagrange multipliers only for those constraints which are active at the solution. A constraint which is inactive can be removed from the problem without affecting the solution, which means there is no tradeoff between the goals and the constraint.

Think of Lagrange multipliers as the incremental *cost* of each active constraint on the solution.

Example: Consider optimizing propagation delay through a gate subject to a constraint on gate width and a constraint on bias current. Suppose that at the optimum, both constraints are active. There are two Lagrange multipliers for this problem:

- Incremental cost of propagation delay versus gate width.
- Incremental cost of propagation delay versus bias current.

For this problem, the units of the Lagrange multipliers are seconds / meter and seconds / ampere, respectively.

In a problem with several goals which have been combined as a sum-of-squares, the target value is dimensionless. In this case, the units of the Lagrange multipliers are the reciprocal of the units of their associated constraint.

Example: A multiplier associated with a width constraint has units of meter^{-1}.

Characteristics of functions

The success of an optimization depends highly on the behavior of the functions related to each specification.

Generally, the PSpice Optimizer obtains values (measures performance) for each of the specifications by *evaluating* a trace which results from a simulation with varying parameter values. The optimizer can experience difficulties if the accuracy of the measurement is decreased by:

- Discontinuities in the simulation results.
- An error in the goal function definition.
- Inaccuracies in the simulation results on which the evaluation is based.

Generally, simulations and measurements using AC and DC analyses behave better than simulations using transient analyses. This is particularly true if the evaluations are set up to measure the value of a single point (e.g., the time when a trace crosses a specified level). This kind of measurement tends to behave discontinuously as the parameters change, creating difficulties for the optimizer.

To review, the PSpice Optimizer measures performance in one of three ways: by taking the value of the single-point trace (trace function) in PSpice A/D, by applying a goal function to the trace, or by evaluating a PSpice Optimizer expression. Trace functions and goal functions require a simulation; PSpice Optimizer expressions do not.

To improve measurement accuracy, consider any of the following techniques:

- Use several points rather than a single point for the Probe goal function. That is, specify several points on a waveform instead of a single point.
- Reduce the step ceiling in Transient analysis to produce a more finely sampled set of data.
- Increase simulator accuracy (e.g., reduce RELTOL).

The first proposed technique is preferred because it does not affect the time required to run each simulation (usually the determining factor in how long an optimization run takes).

Global and local minima

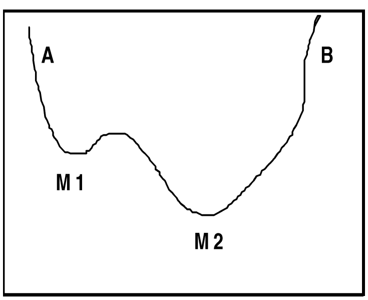


Figure 20 *Global and local minima of a function.*

The curve in Figure 20 shows a 1-dimensional function with 2 minima. Point M1 is a local minimum; it satisfies the conditions for a minimum, but there is another minimum which is smaller. Point M2 is the global minimum for the function. There are no points within the range of the function which are smaller.

All practical optimization techniques find local minima, including the algorithms used by the PSpice Optimizer. This may or may not present a problem. The application may not have any local minima within the domain of interest. If local minima do exist, the global minimum may be the nearest solution to the starting point. This is discussed further in Starting points on page 4-93.

Starting points

It is important to begin with a good estimate of the starting point. There are two reasons for this:

- The process may converge to the *wrong* solution (a local minimum) rather than to the *right* solution (the global minimum).

  For example: Consider Figure 20. If point A is chosen, the PSpice Optimizer will most likely find local minimum M1. If point B is chosen, the optimizer will most likely find the global minimum M2.

- The PSpice Optimizer may require a large number of simulations to find a region close to a solution. It is usually more efficient to find the approximate location of the desired solution (perhaps by performing a number of analyses sweeping out ranges of the parameters) before starting the optimization process.

Convergence

When running an optimization, the PSpice Optimizer varies the parameter values and measures the resulting performance. For each subsequent iteration, the optimizer chooses each parameter step to reduce the error between the design's measured and specified target performance.

If the optimizer finds a solution where all of the specifications are met, then the process has *converged*. There are several common reasons why the process may fail to converge:

- There is no solution to the problem as specified.
- The simulation and/or evaluations are not accurate enough to allow the solution to be found.

- A limit on the number of simulations or elapsed time is encountered.
- The optimizer finds a spurious numerical minimum which is not the desired solution.

To improve convergence, consider the following techniques:

- In the second case, use more accurate measurement techniques (possibly with the aid of *help* circuitry).
- In the last case, restart from a different starting point which might lead to a different solution.

Parameter bounds

The PSpice Optimizer performs *bound-constrained* minimization. This means it will solve problems where one or more of the parameters are limited by the specified upper or lower bound for that parameter. In other words, the optimizer finds a solution (if one exists) even if one or more parameters are at their limit.

However, solving this kind of problem is intrinsically more difficult than performing *unconstrained* minimization.

If one or more parameters appear to be limited during the optimization run, and you don't expect the final solution to have limited parameters, you could save time by using one or both of the following techniques:

- Use a starting point that is further from the parameter limits.
- Loosen the limits on the parameter(s) in question.

Derivatives

To perform optimization, the PSpice Optimizer computes the matrix of partial derivatives—the *Jacobian*.

The PSpice Optimizer calculates derivatives either:

- once when you select Update Derivatives from the Tune menu, or
- automatically for each iteration when you start optimization by selecting Auto from the Tune menu.

See Exploring the effect of parameter and specification changes on page 3-74 for more information.

How the PSpice Optimizer estimates derivatives

The PSpice Optimizer approximates derivatives using a finite difference approach. In one-dimensional terms, this method computes an approximation to the first derivative of a function *f(x)* by:

$$f'(x) \cong \frac{f(x+h) - f(x)}{h}$$

where *h* is a small perturbation.

The optimizer organizes the simulations and evaluations to compute the Jacobian in the most efficient way possible.

Example: If there are two specifications, each of which requires a DC analysis of the same circuit file, the optimizer will run a single simulation for each parameter, then load the data file (.DAT) into PSpice A/D and evaluate the perturbed values of *f*. If there are M specifications (all using the same analysis type and the same circuit file) and N parameters, then forming the Jacobian takes:

- N simulations, and
- M Probe goal function evaluations per simulation.

Note *The time needed to simulate is usually much greater than the time needed to evaluate the goal functions. This means that the time taken to optimize a design depends heavily on the number of variable parameters.*

Limitations of derivative data

See Controlling finite differencing when calculating derivatives (Delta option) on page 4-98 for more information on Delta.

A derivative analysis calculates a linear relationship between a parameter and a specification. It assumes that the function is linear near the initial value within a region defined by the value set for the Delta option. If the data is well-behaved in this region, then this is a valid assumption; the PSpice Optimizer can use the derivative to approximate specification values based on the linear relationship.

However, when the function is not well behaved in the region around the initial value, the approximation may not be valid.

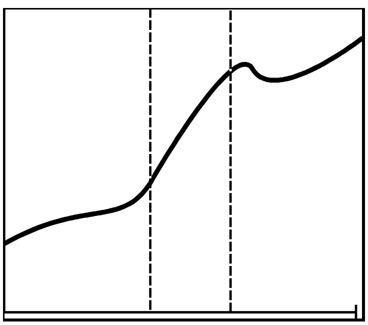


Figure 21 *Hypothetical function.*

For example: Assume that the function shown in Figure 21 is the plot of a specification's behavior vs. a parameter value. Note that the function is approximately linear between the dashed lines, but not necessarily linear outside of that region.

If you pick an initial value for the parameter which is between the two dashed lines, then subsequently compute the derivatives, the derivative data provides a reasonable approximation for any other parameter value between those lines. However, if you try to use the same derivative data to estimate new specification values for parameter values outside of those lines, the estimates are not reliable.

When, for a given parameter, the difference between its initial value and the value of interest is large (that is, the relative difference is much bigger than Delta), modify its initial value and restart the simulation.

Note *When you modify the current value for a parameter to recalculate specification values (by editing the value appearing in the PSpice Optimizer window), the PSpice Optimizer uses the derivative data rather than a resimulation to determine the new values. Therefore, you should periodically verify the results with another simulation (see "Ensuring reliable results when tweaking values" on page **3-77**).*

Target value scaling

When there is more than one goal, or a combination of goals and constraints, the PSpice Optimizer needs to scale the raw measurements before combining them.

Example: Consider a least-squares optimization with two specifications:

- Collector-base capacitance
- Collector resistance

The first of these can have values of several picofarads; the second of tens of Kohms. Clearly, adding the squares of the errors for these specifications will lose the significance of the capacitance term. Instead, the optimizer scales the values as follows:

$$T_{Scaled} = \frac{T_{measured} - T_{target}}{Range}$$

Suppose the specified target and measured values are as shown below.

Table 4-1

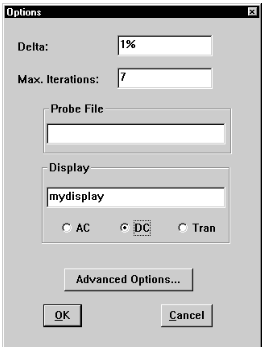
Specification	Target value	Allowed range	Measured value
collector-base capacitance	10 pF	±1 pF	8 pF
collector resistance	10 k	±1 k	12 k

Then the scaled values are:

$$\frac{(8\text{pF} - 10\text{pF})}{1\text{pF}} = -2 \quad \text{and} \quad \frac{25k - 10k}{1k} = 15 ,$$

respectively.

Since these numbers are close enough in magnitude, the PSpice Optimizer can combine them without losing numerical significance.



The PSpice Optimizer saves option settings to the .OPT file so that they remain with the optimization's parameter and specification settings.

To generate a summary of the option settings for a given design, choose Report from the File menu.

Default options

This section describes the basic configuration options for the PSpice Optimizer.

To display the Options dialog box

1 From the Options menu, choose Defaults.

Controlling finite differencing when calculating derivatives (Delta option)

The Delta option specifies the relative amount (as a percentage of current parameter value) by which the PSpice Optimizer perturbs each parameter from its initial value when calculating the derivatives.

The optimizer uses gradient-based optimization algorithms that use a finite difference method to approximate the gradients (gradients are not known analytically). To implement finite differencing, the optimizer:

1. Perturbs each parameter in turn from its current value by an amount *h*.
2. Evaluates the function at the perturbed value.
3. Subtracts the old function value from the new.
4. Divides the result by *h*.

Note *There is a tradeoff. If h is too small, the difference in function values is unreliable due to numerical inaccuracies. However if h is too large, the result is a poor approximation to the true gradient.*

To control parameter perturbation when calculating derivatives

1 Enter a value in the Delta text box that defines a fraction of the parameter's total range.

Example: If a parameter has a current value of 10^{-8}, and Delta is set to 1% (the default), then the PSpice Optimizer perturbs the parameter by 10^{-10}.

The 1% default value is suitable for the accuracy of typical simulations.

2 If the accuracy of your simulation is very different from typical (perhaps because of the use of a non-default value for either RELTOL or the time step ceiling for a Transient analysis), then change the value of Delta as follows:

- If simulation accuracy is better, decrease Delta by an appropriate amount.
- If simulation accuracy is worse, increase Delta by an appropriate amount.

Note *The optimum value of Delta varies as the square root of the relative accuracy of the simulation. For example, if your simulation is 100 times more accurate than typical, you should reduce Delta by a factor of 10.*

Limiting simulation iterations (Max. Iterations option)

The Max. Iterations option defines how many attempts, at most, the PSpice Optimizer can make before *giving up* on the solution, even if the optimizer is making progress.

To limit the simulation iterations

1 In the Max. Iterations text box, enter an integer value that defines the maximum number of attempts you will allow the PSpice Optimizer to make.

Specifying a waveform display (Waveform Data File and Display options)

The Display option defines the name of the waveform display the PSpice Optimizer uses to display simulation results in PSpice A/D. The Probe File option specifies a nondefault .PRB file in which the Probe display information has been stored.

To use a specific Probe display when optimizing

Refer to online Help in PSpice A/D for information on using PSpice A/D Display Control options.

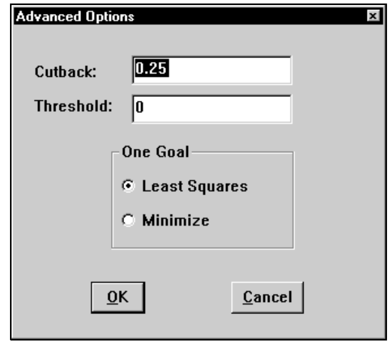
1. In PSpice A/D, configure the plots as you want to view them.
2. Save and name the display:
   a. From the Window menu, choose Display Control.
   b. Enter the name for the display.
   c. Click Save.
3. In the PSpice Optimizer, specify the waveform display.
   a. From the Options menu, choose Defaults to display the Defaults dialog box.
   b. In the Display frame, type the (Probe) waveform display name in the text box.
   c. Select the appropriate analysis type (AC, DC, or Tran).
   d. If you saved the waveform display to a nondefault .PRB file, enter the name of the file in the Probe File text box.

Advanced options

This section describes the advanced configuration options for the PSpice Optimizer.

To display the Advanced options dialog box

1. From the Options menu, choose Defaults.
2. Click the Advanced Options button.



The PSpice Optimizer saves option settings to the .opt file so that they remain with the optimization's parameter and specification settings.

To generate a summary of the option settings for a given design, select Report from the File menu.

Controlling cutback (Cutback option)

The Cutback option defines the minimum fraction by which an internal step is reduced while the PSpice Optimizer searches for a reduction in the goal's target value.

To set cutback

1. In the Cutback text box, enter a decimal fraction that defines the minimal percent reduction in internal step size.

   If the data is noisy, consider increasing the Cutback value from its default of 0.25.

Controlling parameter value changes between iterations (Threshold option)

The Threshold option defines the minimum step size the PSpice Optimizer uses to adjust the optimization parameters during the optimization process.

The optimizer assumes that the values measured for the specifications change continuously as the parameters are varied. In practice, this assumption is not justified. For some analyses, especially transient analyses, the goal function values show discontinuous behavior for small parameter changes. This can be caused by accumulation of errors in iterative simulation algorithms.

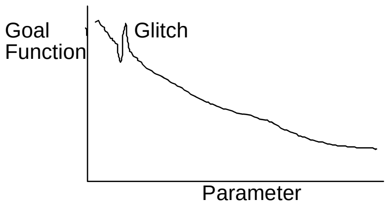


Figure 22 *Hypothetical data glitch*

Figure 22 demonstrates a typical case. The effect of the glitch is serious—the optimizer can get stuck in the spurious local minimum represented by the glitch.

The optimizer's threshold mechanism limits the effect of unreliable data.

To control parameter perturbation between iterations

1. In the Threshold text box, enter a value that defines a fraction of the current parameter value.

   For example: A Threshold value of 0.01 means that when the PSpice Optimizer changes a parameter value, the value will change by at least 1% of its current value.

   By default, Threshold is set to 0 so that small changes in parameter values are not arbitrarily rejected. To obtain good results, however, you may need to adjust the Threshold values. When making adjustments, consider the following:

   - If data quality is good, and Threshold is greater than zero, reduce the Threshold value to find more accurate parameter values.
   - If data quality is suspect (has potential for spurious peaks or glitches), increase the Threshold value to ensure that the optimizer will not get stuck during the run.

Choosing an optimization method for single goal problems (Least Squares/Minimization options)

The PSpice Optimizer implements two general classes of algorithm to measure design performance: least squares and minimization. These algorithms are applicable to both unconstrained and constrained problems.

Least squares

A reliable measure of performance for a design with multiple targets is to take the deviation of each output from its target, square all deviations (so each term is positive) and sum all of the squares. The PSpice Optimizer then tries to reduce this sum to zero.

This technique is known as *least squares*. Note that the sum of the squares of the deviations becomes zero only if all of the goals are met.

Minimization

Another measure of design performance considers a single output and reduces it to the smallest value possible.

For example: Power or propagation delay, each of which is a positive number with ideal performance corresponding to zero.

Choosing the algorithm

When optimizing for more than one goal, the PSpice Optimizer always uses the least-squares algorithm. For a single goal, however, you must specify the algorithm for the optimizer.

To set the optimization method for a single goal

1 Do one of the following:

- Click the Least Squares button to minimize the square of the deviation between measured and target value.
- Click Minimize to reduce a value to the smallest possible value.

If your optimization problem is to *maximize* a single goal, then set up the specification to *minimize the negative of the value*.

For example: To maximize gain, set up the problem to minimize *–gain*.

Tutorial: Optimizing a design (passive terminator)

5

Tutorial overview

The following tutorial takes you through the steps needed to setup and run an optimization starting with the simple terminator example provided with your OrCAD programs.

In this tutorial, you will:

- Verify the design setup:
  - Check that part values are parameterized.
  - Check that optimization parameters are defined.
  - Check the analysis settings.
- Check the goal specification settings.
- Run the optimization.

For a complete hands-on tutorial in which you draw the schematic and set up the optimization from scratch, see **Chapter 2,**Primer: How to optimize a design.

The passive terminator design

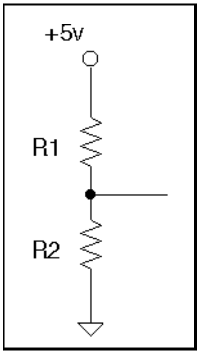


Figure 23 *Resistive terminator circuit.*

Figure 23 shows a simple terminator that you could use, for example, for one line of a backplane. The top end of R1 connects to a +5 v DC supply; the lower end of R2 connects to ground. The center point of the two resistors provides the line termination.

This design must meet two specifications:

- Voltage at the junction of the two resistors (V_{center}) must lie within a specified range.
- Thevenin equivalent resistance of the combination (R_{equiv}) must equal a specified value.

Assume that the DC supply has negligible internal resistance, and that:

- V_{center} must be 3.75±0.1 V.
- R_{equiv} must be 100±1 Ω.

Your objective is to find the best values for R1 and R2 that meet the stated specifications. You can manually solve this problem using the following simultaneous equations:

$$\frac{5R_2}{R_1 + R_2} = 3.75 \quad \text{and} \quad \frac{R_1 \cdot R_2}{R_1 + R_2} = 100$$

These equations solve to R1 = 133.3 Ω and R2 = 400 Ω, an exact solution. As you complete this tutorial, compare this solution to that produced by the PSpice Optimizer.

Loading the design into Capture

To begin, set up a design with:

- Parameters for the resistor values
- A way to measure the performance of the two specifications

Your OrCAD installation includes a schematic for the passive terminator design.

To load the passive terminator schematic

1 From the Windows Start menu, choose the OrCAD Design Desktop program folder and then the Capture shortcut to start Capture.

2 From the File menu, choose Open, then choose Project.

3 Move to the directory containing TERM.OPJ (*Program Files\OrCAD*\PSpice\Samples\Optimize\Term) and select the project file in the File Name list.

This design contains two sections. The first section connects an instance of the terminator to a DC source and to ground. The output is connected to a bubble port labeled Vc. The second section connects the top and bottom ends of an instance of the terminator to ground, and a 1 A current source to its output. This output is connected to a bubble port labeled Vr. The voltage at Vc gives the value of the V_{center} specification. The voltage at Vr gives the equivalent resistance of the network, the R_{equiv} specification.

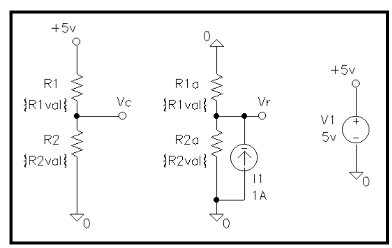


Figure 24 *Schematic for the terminator Example,* TERM.DSN.

Setting part values to expressions

The parameters varied between iterations must relate to the components you are optimizing. For this example, R1 and R2 resistor values are already parameterized.

To see how the resistor values are set up to use optimization parameters

1. Double-click the part for R1 or R2.

   The Edit Part dialog appears.

The curly braces are PSpice syntax for an expression. You can specify any expression that PSpice can evaluate.

2. Within the Edit Part dialog box, note the value of the VALUE property: {R1val} or {R2val}, depending on the part you selected.
3. Click OK to close the Edit Part dialog box.

Defining optimization parameters

Next, set up the optimization parameters that the PSpice Optimizer will vary between iterations; these are the same parameters used in the value expressions for R1 and R2. For this example, R1val and R2val are already defined using an OPTPARAM part as shown below.

OPTIMIZER PARAMETERS:

Name	Initial	Current
R1val	500	500
R2val	500	500

The optimization parameter settings are the same for R1val and R2val as follows:

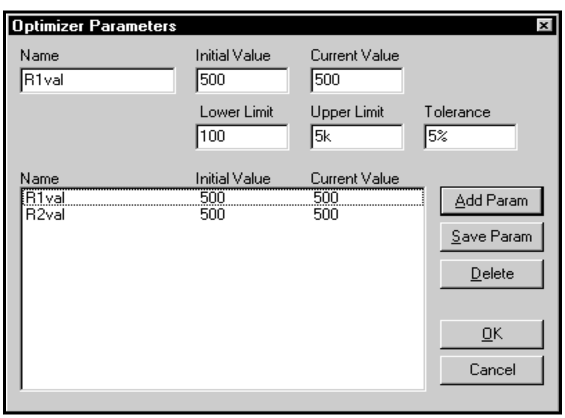
Setting	Value
Initial Value	500
Current Value	500
Lower Limit	100
Upper Limit	5k
Tolerance	5%

To see the settings for R1val and R2val

1 On the schematic, double-click OPTIMIZER PARAMETERS.

By default, the settings for R1val are displayed.

2 To view settings for R2val, click the R2val entry in the parameter list.



3 Click OK to close the Optimizer Parameters dialog box.

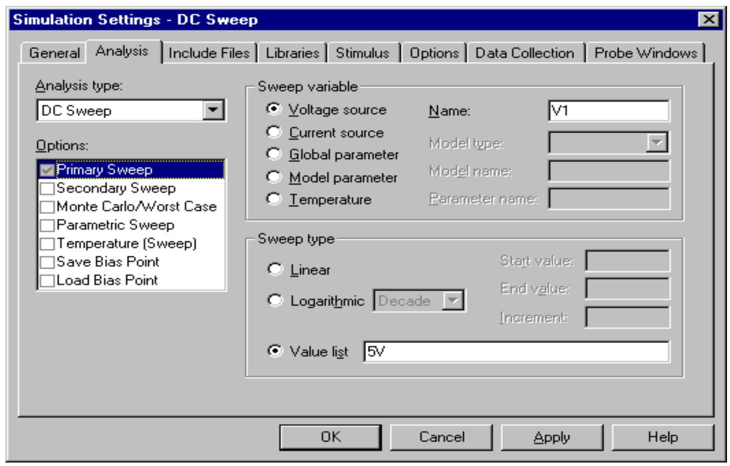
Defining the analysis type

For each specification, set up an analysis which PSpice will run for each iteration of the optimization. This example is already set up for a 1-point DC analysis, with the supply set to 5 V.

To see the analysis settings

1 From the PSpice menu, choose Edit Profile.

The Simulation Settings dialog box appears.



2 Under Analysis type, select DC Sweep and verify the settings.

Running an initial circuit analysis

Before optimizing, verify that the circuit works, and check that the voltages at Vc and Vr are as you expect.

To test the circuit setup

1 From Capture's PSpice menu, choose Run.

2 From PSpice A/D's Trace menu, choose Add Trace and click `V(Vc)` and `V(Vr)`.

3 Click OK.

4 Verify that the voltages at Vc and Vr are 2.5 V and 250 V respectively.

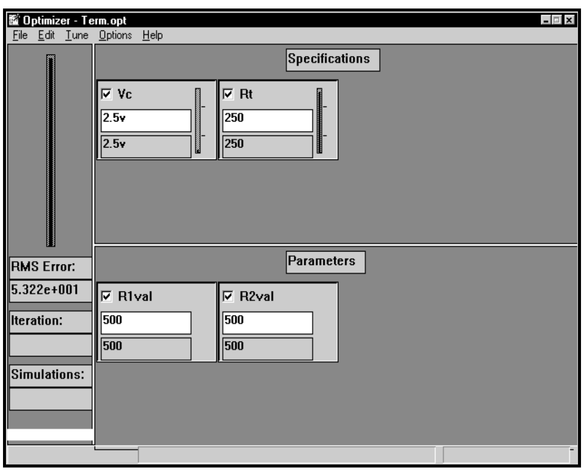
Starting the PSpice Optimizer

Assuming the circuit simulated successfully, you are now ready to start the PSpice Optimizer and complete optimization setup.

To start the PSpice Optimizer

1 From Capture's PSpice menu, choose Run Optimizer.

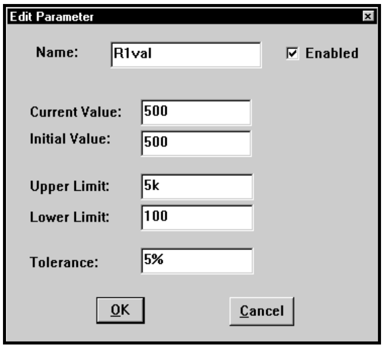
The PSpice Optimizer window displays the current optimization file, TERM.OPT, in the title bar.



The two parameters, R1val and R2val, appear in the parameters area. Because the specifications, Vc and Rt, are predefined for this example circuit, they also appear in the specifications area. Initially, the error gauge area is empty.

Viewing the parameter description

The PSpice Optimizer automatically loads the parameter descriptions defined earlier in Capture.



To verify the parameter descriptions

1. From the Edit menu, choose Parameters.
2. To see R1val settings.
   a. In the Parameters list, click R1val.
   b. Click Change to display the Edit Parameter dialog box.
3. Click Cancel to leave the parameter unchanged.
4. Optionally repeat steps 2 and 3 for R2val.
5. Click Close to exit the Edit Parameter dialog box.

Defining the goals and constraints

To review, the specifications for this example are:

- Voltage, V_{center}, at the junction must be 3.75±0.1 V.
- Thevenin equivalent resistance, R_{equiv}, must be 100±1 Ω.

One way to handle the optimization problem is to treat both specifications as a goal.

For each goal, you must define:

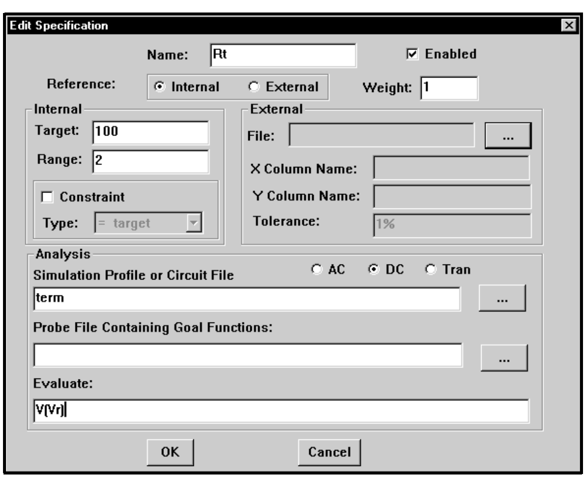
- its name
- its target value and range
- the analysis to use for simulation
- the circuit file containing the netlist description
- the way to measure performance

For this example, these are already defined.

To see the Vc and Rt goal settings

1 From the Edit menu, choose Specifications.

2 Click Vc, then click Change.



Notice that Vc is described as a goal (Constraint check box is cleared). The analysis type is DC and the evaluation to measure performance is the trace function V(Vc)—as in the preliminary test simulation.

3 Click Cancel to leave the specification unchanged.

4 To see the settings for the Rt goal, click Rt, then click Change.

Again, the analysis type is DC, and the evaluation to measure performance is the trace function V(Vr).

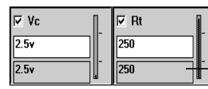
5 Click Cancel to leave the specification unchanged.

6 Click Close to exit the Edit Specification dialog box.

Checking that the design will simulate

Before running a full optimization, you should run a single iteration to make sure the design still simulates (by selecting Update Performance from the Tune menu). The PSpice Optimizer performs the simulations and trace function evaluations required to find the current value for each active specification, and updates their initial values.

For this example, an initial iteration has already been run. The Vc specification shows 2.5 and the Rt specification shows 250 in the initial value fields. Note that this example requires one PSpice simulation to update the performance.



initial value

Starting the optimization

You are now ready to run the full optimization to find resistor values which satisfy the two specifications (output voltage 3.75 V, equivalent resistance 100 Ω).

To start optimizing

1 From the Tune menu, choose Auto and click Start.

First, the PSpice Optimizer computes the derivative of each specification with respect to each parameter at the initial value. While this is in progress, the message `Updating Derivatives` appears in the status bar of the optimizer window. In this example, two simulations are performed—one for each parameter.

Using the derivative information, the optimizer selects a direction in which to vary the parameters. The optimizer tries parameter changes along this direction until it achieves a reduction in the overall error. The optimizer then updates the parameters, and computes new derivatives.

The optimizer repeats this process until it achieves success, failure, or you terminate the process (from the Tune menu, point to Auto and select Terminate).

For this example, let the process run to completion. This takes 5 more simulations.

As the optimization proceeds, the vertical gauge and numeric display fields in the error gauge area (far left in the PSpice Optimizer window) show updated values. The numeric field shows the total error (the sum of the squares of the normalized errors for the active specifications) and the gauge shows the error relative to the starting point. Final results are shown in Figure 25.

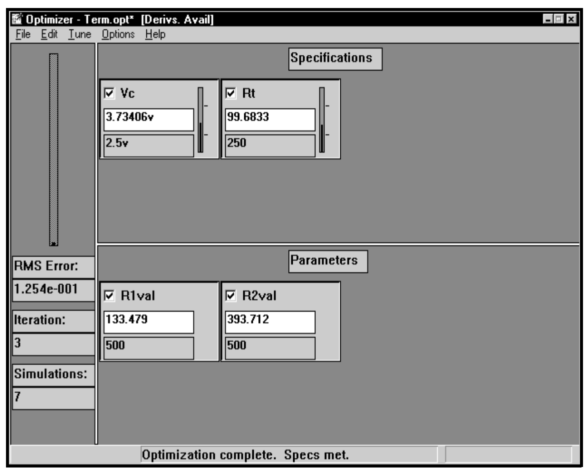


Figure 25 *Optimization results for the passive terminator example.*

Compare the parameter values of 133.5 for R1val and 393.7 for R2val with the calculated values on page 5-106. They are very close.

Changing a goal to a constraint

Try examining what happens when you change one of the specifications from a *goal* to a *constraint*.

For an example, see See Goals versus constraints on page 4-86 for more information.

To change a goal to a constraint

1. Edit a specification:
   a. In the PSpice Optimizer window, double-click the hot-spot (lower right-hand corner) in either the Vc or Rt specification box.
   b. Select the Constraint check box.
   c. In the Type list, select the type of constraint.
2. From the Edit menu, choose Reset Values to set the current values back to the original initial values.
3. From the Tune menu, choose Update Performance.
4. If you want to run a complete optimization:
   a. From the Tune menu, choose Auto.
   b. Click Start.

   For this example, the PSpice Optimizer should produce approximately the same result for each configuration of the two specifications (one goal and one constraint).

Saving results

At this point you would ordinarily choose Save from the File menu to save the results. However, to avoid modifying the tutorial files, either:

- From the File menu, choose Save As to store the results under another file name.

or

- Omit the save process entirely.

Tutorial: Exploring design tradeoffs (active filter)

6

Tutorial overview

The following tutorial shows you how to use the PSpice Optimizer to explore design tradeoffs. Using the simple active filter example provided with your OrCAD programs, you will:

- Review the optimization setup in Capture: parameter definitions and parameterized expression assignments.
- Review the goal and constraint setup in the PSpice Optimizer.
- Change a parameter value to see the effect on goal and constraint values.
- Change a goal value to see the effect on parameter values.

The active filter design

The filter has three adjustable resistors. These resistors adjust center frequency, bandwidth, and gain. In this case, the adjustments are interdependent.

Requirements for the filter are:

- Center frequency (Fc) must be 10 Hz with 1% accuracy.
- 3 dB bandwidth (BW) must be 1 Hz with 10% accuracy.
- Gain (Gain) must be 10 with 10% accuracy.

To load the active filter design

1. From the Windows Start menu, choose the OrCAD Design Desktop program folder and then the Capture shortcut to start Capture.

2. In the Project Manager, from the File menu, choose Open, then choose Project.
3. Move to the directory containing BPF.OPJ (*Program Files\OrCAD*\PSpice\Samples\Optimize\Bpf) and select the project file in the File Name list.

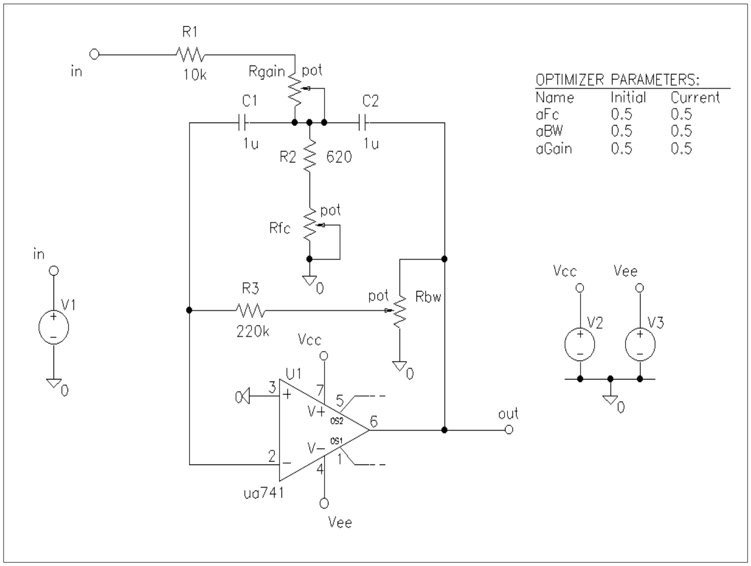


Figure 26 *Schematic for the active filter example,* BPF.DSN.

The parameters

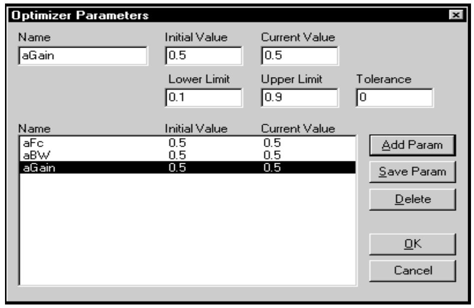
The three variable resistors—Rfc, Rbw, and Rgain—are implemented as potentiometers. The potentiometer part has an property called SET which describes the slider position—a value between 0 and 1.

To optimize slider position, the optimization parameters afc, aBW, and AGain are assigned to the SET property for the parts Rfc, Rbw, and Rgain, respectively.

To see the assignment for SET, double-click on one of the variable resistor parts to bring up its list of properties.

The parameters are shown in the OPTPARAM part on the schematic. Each parameter is set up to range from 0.1 to 0.9 with an initial value for each set to 0.5 (each potentiometer's center point). The settings for aGain are shown below.

To display the Optimizer Parameters dialog box, double-click the OPTPARAM part.



The goals

The example circuit is set up with predefined goals which you can view using the PSpice Optimizer.

To start the PSpice Optimizer

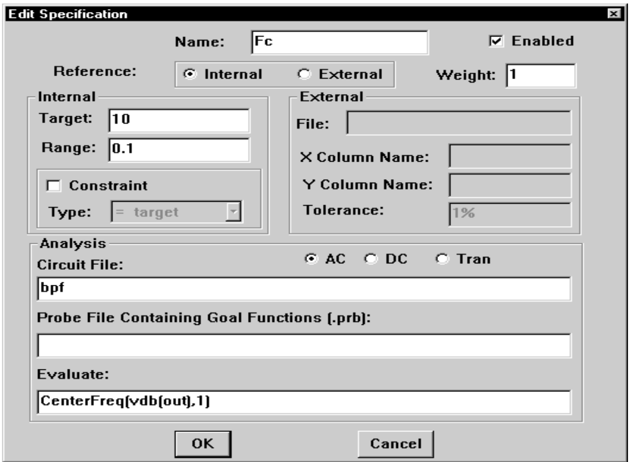
1 From Capture's PSpice menu, choose Run Optimizer.

The PSpice Optimizer window shows the three parameters (aFc, aBW and aGain) and the three corresponding goals (Fc, BW, and Gain). The goals are defined as follows:

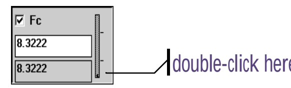
Table 6-1

Setting	Center frequency	Bandwidth	Gain
Name	Fc	BW	Gain
Target	10	1	10
Range	0.1	0.1	1
Analysis	AC	AC	AC
Circuit File	bpf	bpf	bpf
Evaluate	CenterFreq (vdb(out), 1)	Bandwidth (vdb(out), 3)	max(v(out))

The Fc settings as they appear in the Edit Specification dialog box are shown below.



To display the Edit Specifications dialog box, double-click the lower right-hand corner of the specification box of interest in the PSpice Optimizer window.



When finished browsing, click Cancel.

Note that, to measure performance, all three goals are evaluated using goal functions.

Testing performance

To remeasure performance, select Update Performance from the Tune menu.

Initial performance has already been measured for the active filter ensuring that the design simulated as expected, that current values for the goals were calculated, and that initial values were updated. Initial performance calculated to:

Table 6-2

Goal	Current / Initial value	Target value
Fc	8.322 Hz	10 Hz
BW	0.7122 Hz	1 Hz
Gain	14.81	10

Calculating derivatives

To estimate performance effects with small changes in parameters (or specifications), the PSpice Optimizer uses derivatives of each specification with respect to each of the parameters.

To calculate the derivatives

1 From the Tune menu, choose Update Derivatives.

The message `Updating Derivatives` appears in the status bar while this takes place.

Tweaking parameters

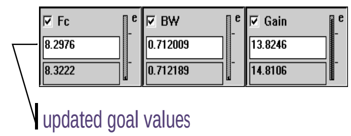
Once the derivatives are calculated, you can use the PSpice Optimize to explore how small changes in the parameters affect the performance of the design.

To examine the performance effect when changing aGain from 0.5 to 0.4

1 Highlight the current value for a Gain, and type 0.4.

2 Press Enter↵.

The value of the Gain specification changes from the initial value of 14.811 to a new value of 13.825. The target is 10.0, which is a change in the right direction.



Center frequency and bandwidth also change. In the case of center frequency, the change from its initial value of 8.322 to a new value of 8.298 is in the wrong direction.

This frequently occurs. You can try various strategies to get closer to the target values. Two common approaches are:

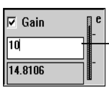
- Adjust a given parameter to get the best results, then vary the other parameters.

  Example: Continue to adjust aGain until you are satisfied with the performance. Then incrementally change Fc until you are satisfied. And finally, change BW.

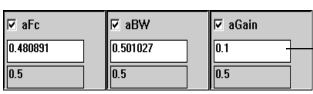
- Change every parameter in the set by a small amount at a time, and continue in this manner to get the best results.

  Example: Adjust aGain once by a small amount, then Fc, then BW. Continue with this pattern of adjustments until you are satisfied.

Tweaking goals and constraints



new goal value



updated parameter values

The PSpice Optimizer gives you the advantage of doing something not possible on the bench: changing the performance results and seeing what parameter values would produce these results.

To investigate the parameter changes when changing Gain to a new target value of 10.

1. Make sure that automatic recalculation is selected.
   a. From the Options menu, choose Recalculate.
   b. In the When frame, click Auto.
   c. Click OK.
2. Highlight the current value for Gain, and type 10.
3. Press Enter.

   The PSpice Optimizer automatically adjusts the parameters to satisfy the results, and then updates the results to match. The result changes from the value you specified to the nearest value which still satisfies all of the parameter limits.

   In this case, the lower limit of the aGain parameter (0.1) is violated, so the optimizer uses the smallest allowed parameter value. This gives a value of 10.8 for Gain.

Note *Because the PSpice Optimizer computes estimates using the previously calculated derivatives, results are typically reliable for only small changes in parameter values. After significant tweaking, you should resimulate and recompute the derivatives to see true performance.* *See "Ensuring reliable results when tweaking values" on page 3-77.*

Completing optimization

To finish exploring the active filter design, run an optimization.

To start optimizing

1 From the Tune menu, choose Auto and click Start.

The PSpice Optimizer finds a set of parameters for which the specifications are met. Figure 27 shows the results.

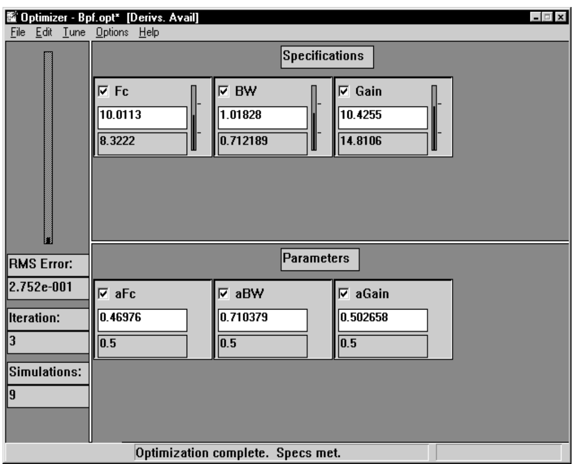


Figure 27 *Optimized values for the active filter example.*

Note *The number of simulations required to complete the optimization varies, depending on the starting values (in the current value fields) when you initiate optimization.*

Tutorial: Using constrained optimization (MOS amplifier)

7

Tutorial overview

The following tutorial shows you how to use the PSpice Optimizer to set up and run constrained optimization. Using the MOS amplifier example provided with your OrCAD programs, you will:

- Review the optimization setup in Capture: parameter definitions and parameterized expression assignments.
- Review the goal and constraint setup in the PSpice Optimizer including the evaluations used to measure performance.
- Review the optimization method selected for single-goal problems.
- Run the optimization.

The CMOS amplifier design

For this optimization, the CMOS amplifier requirements are:

- Minimize power consumption.
- Maintain gain at 20.
- Maintain 3 dB bandwidth at 1 MHz or greater.

To load the CMOS amplifier design

1 From the Windows Start menu, choose the OrCAD Design Desktop program folder and then the Capture shortcut to start Capture.

2 In the Project Manager, from the File menu, choose Open, then choose Project.

3 Move to the directory containing M2.OPJ (*Program Files\OrCAD*\PSpice\Samples \Optimize\M2) and select the project file in the File Name list.

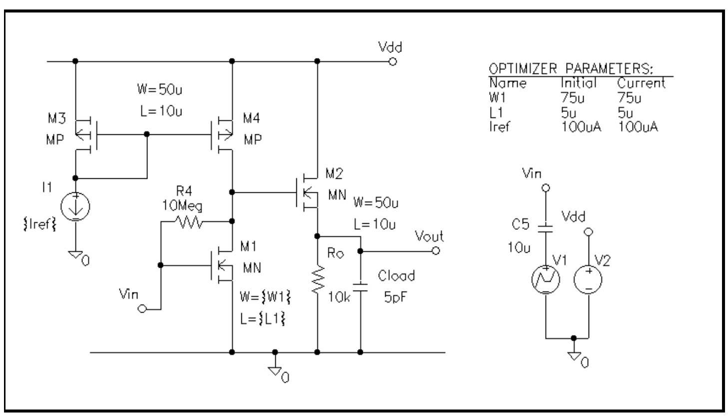


Figure 28 *Schematic for CMOS amplifier example,* M2.DSN.

The circuit consists of a common source stage (M1) with active load (M3 and M4) and a source follower (M2).

The parameters

For this example, there are three circuit values that you must optimize, and three corresponding parameters that the PSpice Optimizer will vary:

- Channel length for M1 (W1)
- Channel width for M1 (L1)
- Bias current for M3 and M4 (Iref)

Parameterized expression assignments In the schematic, you can see the parameterized component values for MOSFET M1 (`W={W1}`, `L={L1}`) and the current source I1 (`{Iref}`).

In preliminary tests, the parameter values W1 = 75 m, L1 = 5 m, and Iref = 100 mA produce the following performance characteristics:

gain	23.8
bandwidth	2.2 Mhz
power consumption	2.2 mW

These initial parameter values yield both excess gain and bandwidth, so a reduction in power consumption appears feasible. But because the gain, bandwidth, and power depend nonlinearly on circuit parameters such as transistor dimensions, manual optimization is impractical.

To display the Optimizer Parameters dialog box, double-click the OPTPARAM part in the design.

PSpice Optimizer parameters For optimization, the parameters for the amplifier are set up using the OPTPARAM symbol as follows:

Table 7-3

Property	Parameters		
	W1	L1	Iref
Initial Value	75u	5u	100uA
Current Value	75u	5u	100uA
Lower Limit	10u	2u	10uA
Upper Limit	150u	50u	500uA
Tolerance	0	0	0

The evaluations

The amplifier is set up with three performance characteristics: power, gain, and 3 dB bandwidth. To measure performance, the PSpice Optimizer needs to know how to calculate response for each of these.

To see the setup for the analyses:

1. From Capture's PSpice menu, choose Edit Simulation.
2. In the Analysis type box, select either DC Sweep or AC Sweep/Noise to examine the simulation settings.
3. Click Cancel to return.

Power The objective, to minimize power consumption, is a *single-point* goal. This means that a trace function appropriately measures the power response of the circuit.

The amplifier design is set up to measure power by performing a single-point DC analysis and then applying the trace function

```
-I(V2)*10V
```

to the simulation results.

Gain

The amplifier design is set up to measure the spot gain at 1 kHz (assuming that the bandwidth is much greater than this) by performing an AC analysis and then applying the waveform analysis goal function:

```
YatX(V(Vout), 1K)
```

to the simulation results.

YatX is provided with your OrCAD programs. The goal function definition is as follows:

```
; value at given x
YatX(1, where) = y1
{ 1 | sf xvalue(where) !1;}
```

`YatX(V(Vout), 1k)` gives the Y value on the V(Vout) trace for the X value corresponding to 1 k.

3 dB bandwidth

The amplifier design is set up to find the frequency where the output has fallen by 3 dB from its low-frequency value by performing an AC analysis (same as for Gain) and then applying the waveform analysis goal function:

```
LPBW(Vdb(Vout), 3)
```

LPBW is provided with your OrCAD programs. The goal function definition is as follows:

```
; bandwidth of low pass response
LPBW(1, db_level) = y1
{ 1 | sf level(max-db_level, n) !1;}
```

`LPBW(Vdb(Vout), 3)` gives the low pass cutoff value on the Vdb(Vout) trace where the output is 3 dB below the maximum value on the trace.

The goal functions, YatX and LPBW, are contained in the file, PSPICE.PRB. Every OrCAD PSpice A/D installation includes this file.

When loading a data file into PSpice A/D, PSpice automatically loads the global PSPICE.PRB file (located, by default, in your OrCAD root\common directory), and any local .PRB file (located in the working directory) corresponding to the current design.

You can freely update PSPICE.PRB or any other .PRB file with new goal function definitions.

Refer to online Help in PSpice for more information on the .PRB files.

The goals and constraints

The example circuit is set up with predefined goals and constraints which you can view using the PSpice Optimizer.

To activate the PSpice Optimizer

1. From Capture's PSpice menu, choose Run Optimizer.

   The PSpice Optimizer window shows the three parameters (W1, L1, and Iref) and the three specifications. Power is defined as a goal, and gain and bandwidth are defined as constraints as shown below.

Table 7-4

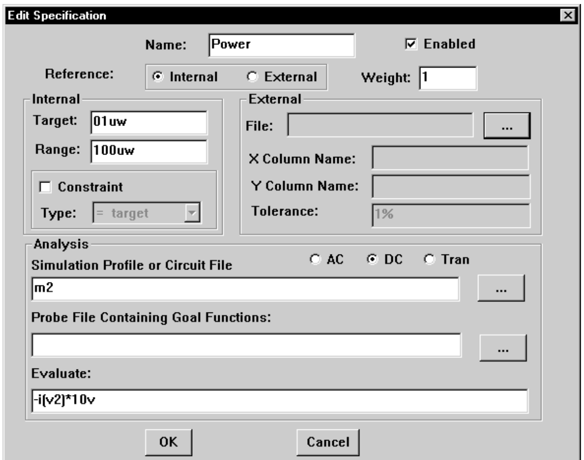
Setting	Power	Gain	Bandwidth
Name	Power	Gain	BW
Target	1 uW	20	1 MHz
Range	100 uW	2	100 kHz
Constraint Type	N/A*	= target**	>= target***
Analysis	DC	AC	AC
Circuit File	m2	m2	m2
Evaluate	-i(v2)*10v	YatX (v(vout), 1k)	LPBW (vdb(vout),3)

* Since Power is a goal, constraint type does not apply.

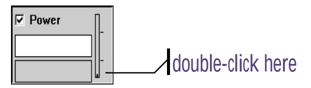
** Gain is defined as a constraint because circuit gain *must* remain at 20±1.

*** BW is defined as a constraint because the 3dB bandwidth *must* be at least 900 kHz or greater.

The Power settings as they appear in the Edit Specification dialog box are shown below.



To display the Edit Specifications dialog box, double-click the lower right-hand corner of the specification box of interest in the PSpice Optimizer window.



When finished browsing, click Cancel.

Setting the method for a single-goal optimization

The amplifier design is set up to use *minimization* to assess performance because this optimization has one goal: minimize the power consumption. The default method, *least squares*, is appropriate when minimizing the square of a function.

To see the Minimize setting, from the Options menu, choose Defaults and click Advanced Options.

Performing the optimization

You are now ready to run the optimization.

To optimize the amplifier design

1 From the Tune menu, choose Update Performance to calculate initial performance. The results are as shown in Figure 29.

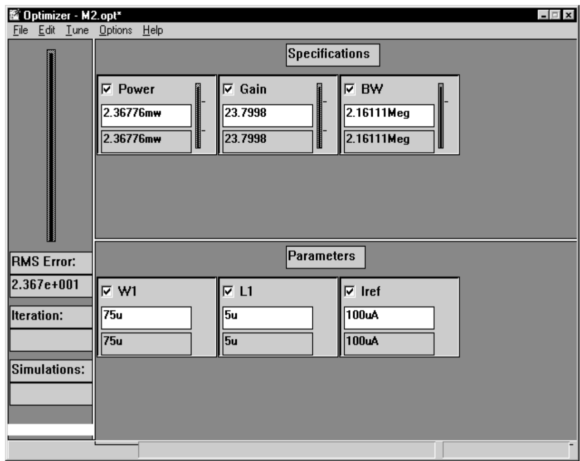


Figure 29 *Updated performance values for the amplifier example.*

2 From the Tune menu, choose Update Derivatives to compute the partial derivatives of each goal and constraint with respect to each parameter.

3 Display the derivatives:

a From the Tune menu, choose Show Derivatives.

b Click Close, when finished.

4 From the Tune menu, choose Auto and click Start to start the optimization.

Notice that the performance indicators change as the optimization proceeds. When complete, optimized values should appear as shown in Figure 30.

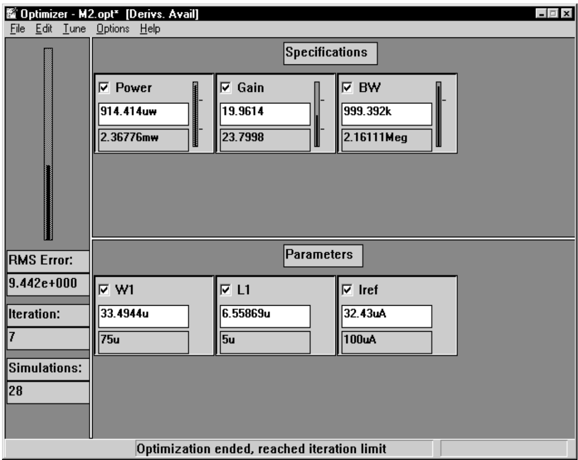


Figure 30 *Optimized values for the amplifier example.*

This optimization stops after reaching the maximum number of allowed iterations. Even so, power consumption is minimized (the objective) within the iteration limits.

Note *For a minimization problem like this one, you can still effectively use the PSpice Optimizer to improve the circuit's performance even though the goal is not attainable.*

Tutorial: Fitting model data (bipolar transistor)

8

Tutorial overview

The following tutorial shows you how to use the PSpice Optimizer to fit a parameterized model to a set of measured data points. Using the bipolar transistor example provided with your OrCAD programs, you will:

- Review the bipolar transistor test case: schematic, parameter definitions and parameterized expression assignments, analysis, and external file with measured data.
- Review the goal setup in the PSpice Optimizer.
- Set up PSpice A/D and the PSpice Optimizer to monitor intermediate waveform results.
- Run the fitting process.

Using the PSpice Optimizer to fit data to model parameters

The PSpice Optimizer can fit a parameterized model to one or more sets of data points. The source for this *external* data might be:

- Results from measuring a real device (e.g., using a semiconductor curve tracer).
- A manufacturer's data sheet.
- A PSpice simulation.

To fit a model using the PSpice Optimizer, you need a text file containing the measured values and the measurement points.

For example: For static I-V characteristic of a diode, the external data file would contain pairs of values where each pair consists of the voltage at which the measurement was made and the current through the device at that voltage.

The PSpice Optimizer performs a *least squares* estimation of the parameter values that result in the best fit between the external data and the values produced by the model. More formally, given a set of *N* data points $\{x_i,y_i\}$, a model that has *M* parameters $\{a_j\}$, and a model relationship of the form:

$$y(x) = y(x;a_1 \ldots a_M),$$

the optimizer chooses the parameters $\{a_j\}$ to minimize:

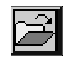
$$\sum_{i=1}^{N} [y_i - y(x_i;a_1 \ldots a_M)]^2$$

The bipolar transistor test case

This tutorial fits parameters to a bipolar transistor (BJT) model using measured data for Ic and Ib versus Vbe at constant Vce.

To load the BJT design

1. From the Windows Start menu, choose the OrCAD Design Desktop program folder and then the Capture shortcut to start Capture.
2. In the Project Manager, from the File menu, choose Open, then choose Project.



3. Move to the directory containing BJTPAR.OPJ (*Program Files\OrCAD*\\PSpice\\Samples\\Optimize\\Bjtpar) and select the project file in the File Name list.

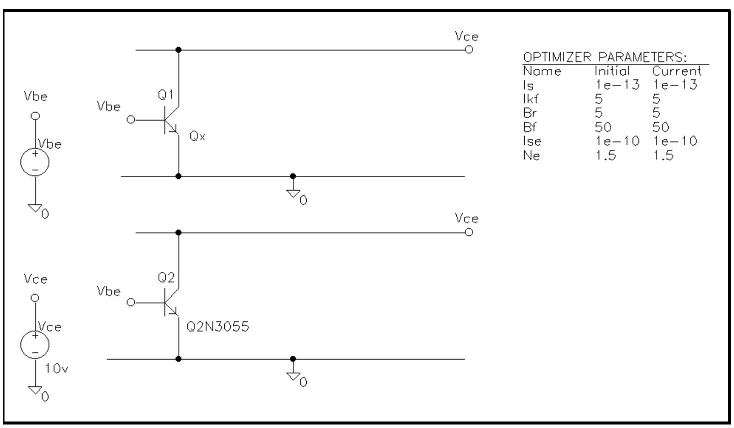


Figure 31 *Schematic for the BJT model fitting example.*

The circuit has voltage sources for Vbe and Vce and an instance, Q1, of a BJT. Q1 is a QbreakN device with its model reference set to Qx—the name of a BJT model.

The schematic also includes an instance of a 2N3055. This is the model that was used to produce the measured data. The 2N3055 in the schematic works as a reference during model fitting.

The QX model definition for this example is contained in the BJTPAR.INC include file. To check whether this file is included, from the PSpice menu, choose Edit Simulation. Then click the Include Files tab.

The parameters

Refer to Q devices in the online *OrCAD PSpice A/D Reference Manual* for more information on the parameters used for bipolar transistor models.

The model parameters to fit are: Is, Ikf, Br, Bf, Ise, and Ne. These parameters are a subset of the parameters PSpice uses to determine the BJT's DC characteristics. (To fit all of the BJT model parameters, you would need more measured data than is provided in this tutorial.)

To display the Optimizer Parameters dialog box, double-click the OPTPARAM part in the design.

PSpice Optimizer parameters

There are six PSpice Optimizer parameters, each corresponding to one of the BJT model parameters listed above. The initial and current values for each of these parameters are set up using the OPTPARAM part as follows:

Property	Parameters					
	Is	Ikf	Br	Bf	Ise	Ne
Initial Value	1e-13	5	5	50	1e-10	1.5
Current Value	1e-13	5	5	50	1e-10	1.5
Lower Limit	1e-14	1	1	20	1e-11	1.2
Upper Limit	1e-11	10	10	200	1e-8	2
Tolerance	0	0	0	0	0	0

To see the parameterized expression assignments for the model parameters:

1. Click Q1.
2. From the Edit menu, choose Model.
3. Click Edit Instance Model (Text).
4. When finished, click Cancel.

Parameterized expression assignments

The model definition is set up with parameterized expression assignments for the model parameters the PSpice Optimizer will vary. All other model parameters are left unchanged.

For example, using the Model Editor, the forward beta is specified as `bf = {bf}`.

The analysis

The BJT example is set up for a DC sweep of the voltage source which provides a range of values for Vbe. These values should match the measurement points contained in the external data file. Ic and Ib are measured at values of Vbe starting from 0.4 V, incrementing by 0.01 V to a maximum of 0.82 V.

The external file of measured data

There are two measured curves: Ic and Ib. Each of these has an associated PSpice Optimizer specification. When fitting model parameters, specifications are different from those used in other kinds of optimizations because they reference an external data file.

For the BJT example, both sets of measured data are contained in the 3055.MDP file. A portion of this file is shown below.

```
    Vbe          Ic           Ib
4.000E-01    6.047E-06    2.655E-06
4.100E-01    8.900E-06    3.253E-06
4.200E-01    1.310E-05    3.989E-06
4.300E-01    1.928E-05    4.897E-06
                 •
                 •
                 •
8.100E-01    8.002E+00    1.922E-01
8.200E-01    9.092E+00    2.371E-01
```

The file contains 3 columns of data: Vbe, Ic, and Ib. The first non-blank line in the file contains names for the columns of data. These map to the specification setup in the PSpice Optimizer.

The goals and constraints

The example circuit is set up with predefined goals which you can view using the PSpice Optimizer.

To start the PSpice Optimizer

1 From Capture's PSpice menu, choose Run Optimizer.

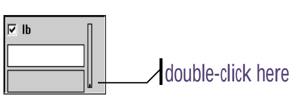
The PSpice Optimizer window shows the six parameters (Is, Ikf, Br, Bf, Ise, and Ne) and the two specifications (Ic and Ib). Ic and Ib are defined as a goals as shown below.

Table 8-5

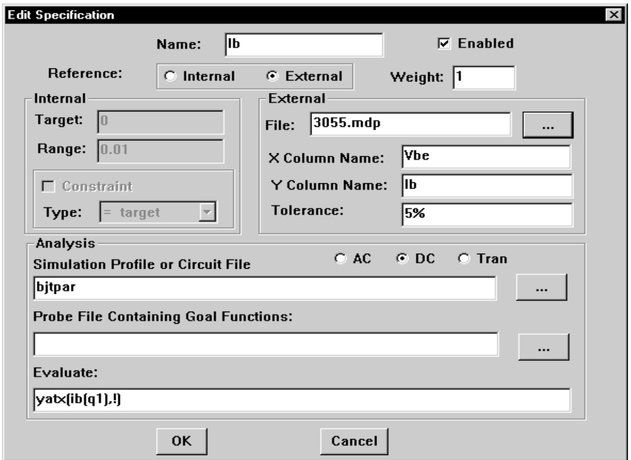
Setting	Ic	Ib
Name	Ic	Ib
Reference	External	External
File	3055.MDP	3055.MDP
X Column Name	Vbe	Vbe
Y Column Name	Ic	Ib
Tolerance	5%	5%
Analysis	DC	DC
Circuit File	bjtpar	bjtpar
Evaluate	YatX(ic(q1),!)	YatX(ib(q1),!)

The Ib settings as they appear in the Edit Specification dialog box are shown below.

To display the Edit Specifications dialog box, double-click the lower right-hand corner of the specification box of interest in the PSpice Optimizer window.



When finished browsing, click Cancel.



Mappings to the external data file Note the reference to the external data file, 3055.MDP, and the values assigned to X Column Name and Y Column Name. The column values map to the column names defined in 3055.MDP, such that the X column gives the data points at which the Y column values were measured. For the BJT example, the Y column contains measured collector current corresponding to the specified base-emitter voltages in the X column.

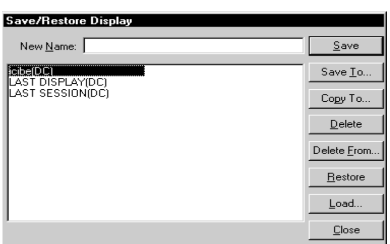
The evaluations To measure performance, both goals use the waveform analysis goal function, YatX (see The evaluations on page 7-132). Notice the '!' character appearing in the expression. This character allows PSpice A/D to make measurements at the multiple X column values.

When the PSpice Optimizer encounters this character, it replaces it with the current X column value before sending it to PSpice A/D for evaluation.

Example: If Evaluate is set to `YatX(Ic(Q1),!)` and if the X column of the external data file contains the values (`1v`, `2v`, `3v`), then the PSpice Optimizer will form the waveform analysis goal function expressions `YatX(Ic(Q1),1v)`, `YatX(Ic(Q1),2v)`, and `YatX(Ic(Q1),3v)`, and send each one to PSpice A/D for evaluation against the simulation results.

Monitoring progress with PSpice A/D

When optimizing, the PSpice Optimizer (with the help of PSpice) generates intermediate results which you can view in PSpice A/D. This is especially useful when fitting model parameters. For the BJT example, this means you can monitor how closely the Ic and Ib values match the target values.



Refer to online Help in PSpice A/D for more on waveform display control.

To set up PSpice A/D and the PSpice Optimizer to monitor the fitting process

1. From Capture's PSpice menu, choose Run to simulate the circuit.
2. Create or use an existing waveform display configuration. For the BJT example, a display named icibe is predefined. To see what the icibe display looks like:
   a. From PSpice A/D's Window menu, choose Display Control.
   b. In the display list, click `icibe(DC)`.
   c. Click Restore.
   d. Click Close.
   e. From the File menu, choose Close.

3 Define the display configuration to use when optimizing. For the BJT example, this is predefined. To verify the display name:

   a From PSpice Optimizer’s Options menu, choose Defaults.

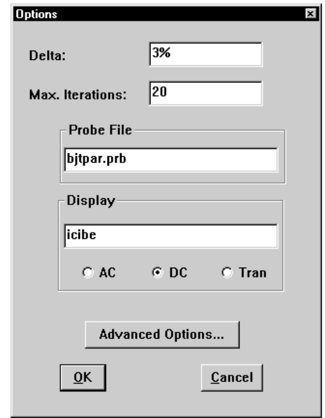
   The display name, icibe, appears in the Display text box and the DC analysis type is selected.

   b When finished, click Cancel.

4 Measure performance and re-display the traces in PSpice A/D:

   a From PSpice Optimizer’s Tune menu, choose Update Performance.

   The optimizer automatically updates the waveform display. When the iteration is complete, the display should look something like the one shown in Figure 32. This display compares the curves for Ic and Ib with curves for an instance of a 2N3055. The percentage error for Ic and Ib is also displayed in the uppermost plot.



In some regions of the device characteristic, the measured currents are relatively insensitive to changes in the parameters. Because of this, the value of Delta is increased to 3% from its default value of 1%. See Controlling finite differencing when calculating derivatives (Delta option) on page 4-98 for more information.

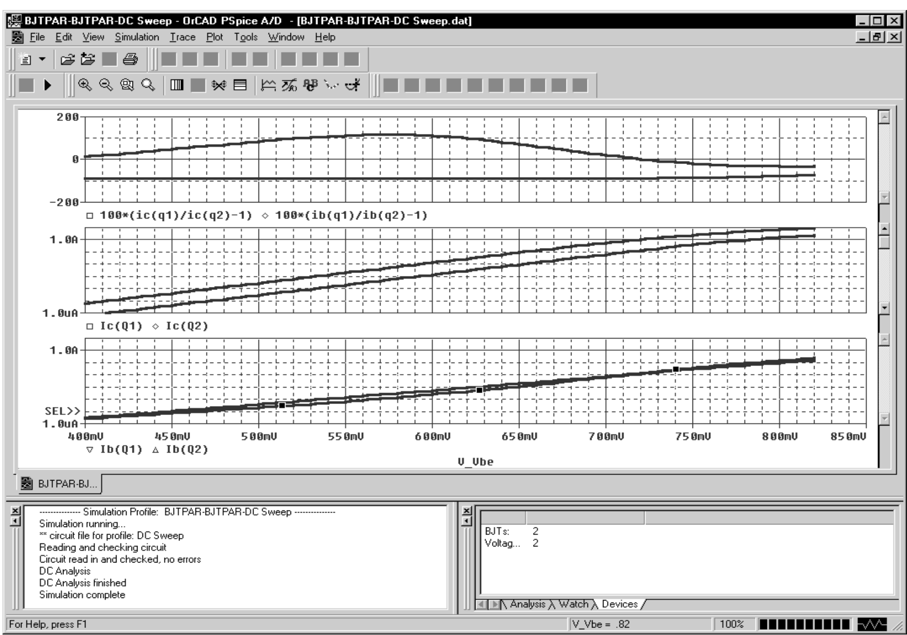


Figure 32 *Initial traces for the Ic and Ib parameters.*

Fitting the data

The PSpice Optimizer looks for a set of parameters which minimizes the total squared error between the measured and simulated curves. When using external data, the optimizer uses the relative error at each measured point. This means that all points have equal weight, regardless of the absolute value. This is essential for fitting semiconductor curves, where the currents may range over many orders of magnitude. Automatic normalization is discussed in detail in Target value scaling on page 4-97.

To fit the data to the model parameters

1 From the Tune menu, choose Auto and click Start.

The following table compares the model parameter values *estimated* by the PSpice Optimizer to the data values *measured* by PSpice for the 2N3055 transistor.

Table 8-6

Parameter	Estimated by the PSpice Optimizer	Measured from PSpice
Is	9.744e-13	9.744e-13
Ikf	4.071	4.029
Br	5	2.949
Bf	98.465	99.49
Ise	9.920e-10	9.025e-10
Ne	1.960	1.941

The estimated parameters compare well with the original values.

The results of the optimization are shown in Figure 33.

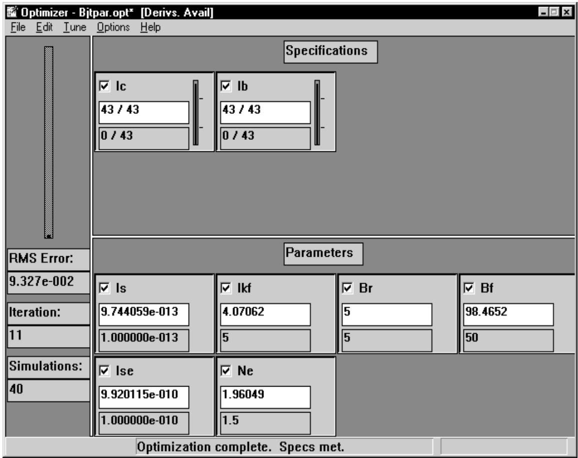


Figure 33 *Optimization results for the BJT model fitting example.*

The PSpice plot in Figure 34 compares fitted and measured curves, and shows the percent relative error in Ic and Ib.

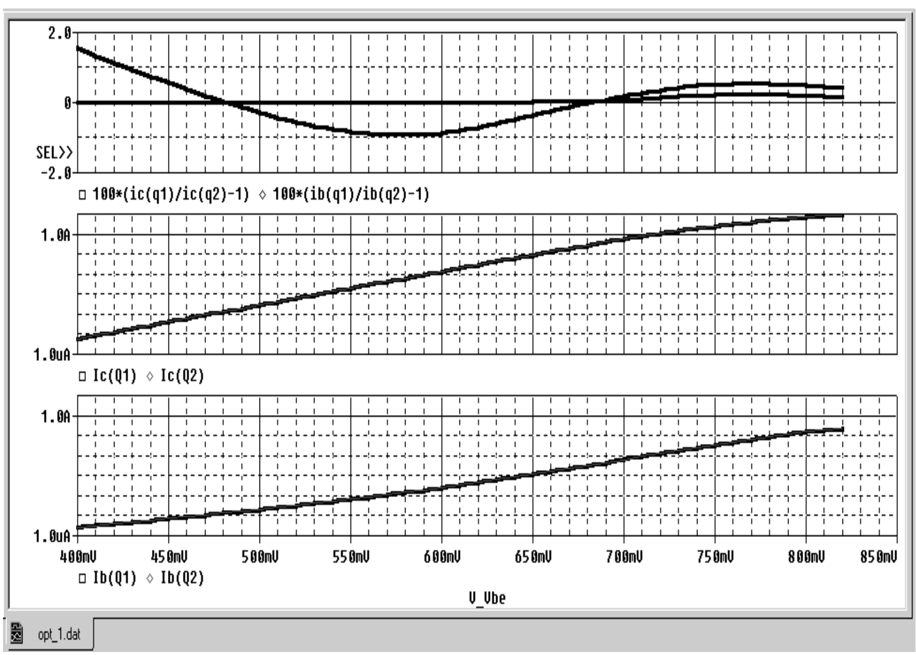


Figure 34 *PSpice A/D display after optimization is complete.*

Error messages

A

Appendix overview

This appendix lists and explains the error messages you might encounter while using the PSpice Optimizer, and, where appropriate, what action to take.

Error message descriptions

Table 7 *Error message descriptions.*

Error message	Description
[11002] No More Resets	No improvement was made by going back to the last good step and recalculating derivatives. Try refining the specifications.
[11003] Mismatched Parentheses	Check for mismatched parentheses in the evaluation expression.
[11004] Undefined Symbol	A symbol in a PSpice Optimizer expression was not recognized.
[11005] Divide by 0	A division by zero in a PSpice Optimizer expression occurred. Try adding a small offset, e.g., 1e-3, to the denominators of fractional terms in the expressions. You may need to change the lower or upper limit of a parameter.
[11006] 0 to the y	Try adding a small offset, e.g., 1e-3, to bases in PSpice Optimizer expressions. You may need to change the lower or upper limit of a parameter.
[11007] Unknown Operator	Valid operators and functions in PSpice Optimizer expressions are (+, -, *, /, **, exp, log, log10, sin, cos, tan, atan).
[11008] Invalid Number	PSpice Optimizer numbers are reals with an optional suffix (T, G, Meg, k, m, u, n, p, f, mil, %) and an optional symbol for units.
[11009] Log of Number <= 0	Log function (log or log10) can not have a zero argument. You may need to change the lower or upper limit of a parameter.

Error message	Description
[11012] .dat File In Use	You must unload the data file from PSpice A/D for the circuit being optimized. Click on the Probe window to make it active, then select Close from the File menu.
[11013] No Simulation Result	PSpice A/D could not open the `.dat` file for the circuit being optimized. The PSpice Optimizer uses two files for optimization (`opt_0.dat` and `opt_1.dat`). Try running a PSpice simulation to investigate the problem. If PSpice does not create a `.dat` file, check that a DC sweep, AC sweep, or transient analysis has been selected, and that there is adequate hard disk space.
[11014] G.F. Evaluation Failed	PSpice A/D could not find one or more of the marked points in the goal function (error message will contain goal function name). Try loading the `.dat` file into PSpice A/D, then from the Trace menu choose Eval Goal Function to test the goal function. Refer to online Help in PSpice A/D for more information on writing goal functions.
[11015] Aborted	Optimization was aborted because of an error or user termination.
[11016] Bad Numeric Field	PSpice Optimizer numbers are real numbers with an optional suffix (T, G, Meg, k, m, u, n, p, f, mil, %) and an optional symbol for units.
[11017] Name Required	Specifications must be given names. Type in an ASCII string in the Name text box in the Edit Specification dialog box.
[11018] MAX Must Be > MIN	The current and initial values for a parameter must be greater than the Lower Limit. These values can be changed on the OPTPARAM symbol in Capture, or select Parameters from the Edit menu in the PSpice Optimizer.
[11019] Must Be Between MIN & MAX	The Upper Limit of a parameter must be greater than the Lower Limit and less than the Upper Limit specified. These values can be changed on the OPTPARAM symbol in Capture, or select Parameters from the Edit menu in the PSpice Optimizer.

Error message	Description
[11020] Evaluation Field Required	The Evaluate text box in the Specification dialog box must contain a valid Probe output variable, Probe goal function, or PSpice Optimizer expression. Refer to application examples in the your PSpice user's guide.
[11021] Column Name Required	The first line of an external data file must contain names for each column of data. The first column should contain the X axis data, and each succeeding column should contain Y axis data for each external specification. The column names (on the first line) must be identical to those used in the Specification dialog box (X Column Name, Y Column Name).
[11022] Would Exceed Max #Params	There is a maximum of 8 parameters allowed for an optimization.
[11023] Would Exceed Max #Specs	There is a maximum of 8 specifications allowed for an optimization.
[11024] All Params Are Disabled	At least one parameter must be selected to perform an optimization.
[11025] Can't Open File	The `.opt` file could not be opened by the PSpice Optimizer. Make sure that this file is not locked by a word processor or other application.
[11026] Problem With External Spec	See [11021] for a description of the external specification data file.
[11027] Max #Iterations Exceeded	The maximum number of iterations can be increased by selecting Defaults from the Options menu. The default limit is 20 iterations. It is often more helpful to refine the specifications than to increase this limit.
[11028] Terminated	The optimization was terminated by the user.
[11029] Can't Make Progress	No improvement was made by going back to the last good step and recalculating derivatives. Try refining the specifications.
[11030] No External File	The external file typed in the Edit Specification dialog box could not be found. See [11021] for a description of the format of this file.
[11031] External Spec - Bad Format	A column could not be found in the external data file. See [11021] for a description of the format for this file.
[11032] Unrecognized Expression	The expression used in the Evaluate text box of the Edit Specification dialog box could not be parsed. See errors [11014] and [11016] for aids in debugging the problem. PSpice Optimizer expressions can only use the following operators: (+, -, *, / , **, exp, log, log10, sin, cos, tan, atan).

Error message	Description
[11033] Need At Least 1 Enabled Goal	At least one goal must be used in an optimization. An optimization can not be performed with constraints only.
[11034] Max External Items = 250	A maximum of 250 lines of data can be used in the external data file.
[11035] Missing External Column	The first line of an external data file must contain names for each column of data. The first column should contain the X axis data, and each succeeding column should contain Y axis data for each external specification. The column names (on the first line) must be identical to those used in the Specification dialogs (X Column Name, Y Column Name).
[11036] Eval Limits: 1 Goal, 1 Constraint	The evaluation version is limited to a maximum of 1 goal and 1 constraint.

File types used by the PSpice Optimizer

B

Appendix overview

This appendix describes how OrCAD programs, used to capture and optimize your design, interact and share information with each other.

- File and program relationships on page B-158 illustrates and describes the data flow among OrCAD programs.
- File type summary on page B-161 explains the contents of each file type.

File and program relationships

Typically, you enter your design using Capture and use this as the basis for optimization with the PSpice Optimizer. Each design produces a single netlist file (*design_name*.NET) which you create in one of the following ways:

- From Capture's PSpice menu, choose Create Netlist or Run.

  or

- From Capture's PSpice menu, choose Run Optimizer.

In the latter case, Capture also produces:

- An optimization file (*design_name*.OPT) which contains optimization settings describing parameters, specifications, and other options. The PSpice Optimizer adds to and keeps the information in this file up-to-date.
- A parameters file (*design_name*.PAR) which contains parameter values as of the last optimization iteration. The PSpice Optimizer keeps the information in this file up-to-date.

Note *The netlist file produced by Capture automatically references the .PAR file. If you are entering your design using a circuit file instead of a schematic, add a .inc ("include") command naming the .PAR file. See "***Chapter C,**Optimizing a netlist-based design*" for details.*

Measuring performance using information in the circuit file and .PRB file

Remember that for a simulation-based optimization, you must define how to evaluate performance. This means that, for each specification (goal or constraint), you must define the following.

What circuit file to use Typically, an optimization uses only one simulation profile or circuit file—based on the design that is active in Capture when you start the PSpice Optimizer. But you can set up the optimization to use a different circuit file for each specification.

Which analysis to run The circuit file contains the analysis directives for any PSpice simulations (e.g., DC sweep, transient, etc.) that you set up for the schematic. However, you control which analyses are actually run for an optimization when you define how to evaluate each specification in the Edit Specifications dialog box.

How to measure performance If, to evaluate performance, your optimization is based on a multi-point simulation, you must define goal functions. Goal functions reside in the [GOAL FUNCTIONS] section of a .PRB file. By default, a given design references the global .PRB file shipped with all OrCAD programs (PSPICE.PRB in the OrCAD root directory\PSpice\Common) and a local .PRB file (*design_name*.PRB in the working directory).

Defining specification criteria in the external data file

Vbe	Ic	Ib
4.000E-01	6.047E-06	2.655E-06
4.100E-01	8.900E-06	3.253E-06
4.200E-01	1.310E-05	3.989E-06
4.300E-01	1.928E-05	4.897E-06
	•	
	•	
	•	
8.100E-01	8.002E+00	1.922E-01
8.200E-01	9.092E+00	2.371E-01

Figure 35 *Sample external data file.*

The external data file contains the measured data used in the optimization (e.g., when fitting data to model parameters). Data is formatted as labeled columns of values.

Typically, the first column contains the independent values at which all other data was measured; this corresponds to the X Column Name in the optimizer's Edit Specification dialog box. All other columns contain the dependent measured values and correspond to the Y Column Name in the dialog box.

To set up an external file

Using a standard text editor:

1. On line 1, enter the column headings for the data separated by spaces or tabs.
2. On all lines following line 1:
   a. Enter data values that correspond to the column headings, separated by spaces or tabs.
   b. Sort the lines of data in ascending X Column Name order.
3. Save the file using any file extension that does not conflict with other OrCAD standard file extensions. We recommend using the naming convention *design_name*.MDP.

For an example of an optimization that uses an external data file, see **Chapter 8,** Tutorial: Fitting model data (bipolar transistor).

File type summary

Table 1 briefly describes all of the file types that the PSpice Optimizer uses directly.

Table 1 *Summary of PSpice Optimizer-related file types.*

File type	Source	Description
.NET .ALS	Generated by Capture or user-entered with a text editor	Netlist file, and alias file, respectively. Define components and connectivity, analysis directives, and simulation control directives for the circuit. Refer to your PSpice user's guide for more information.
.MDP*	User-entered with a text editor	External data file. Contains measured data to which the optimization must adhere.
.OLG	Generated by the PSpice Optimizer when running an optimization	Optimization log file. Contains an optimization audit trail that is useful as a debugging aid when the optimization doesn't converge. Have this file available when contacting Technical Support.
.OOT	Generated by the PSpice Optimizer when generating a report	Optimization report file. Provides a formatted description of the optimization specifications and parameters, and results including derivatives and Lagrange multipliers.
.OPT	Generated by Capture when activating the PSpice Optimizer, or by the PSpice Optimizer when saving results	Optimization file. Defines the optimization settings for the circuit. When an optimization is complete, the PSpice Optimizer saves results (on command) to this file. Capture also reads this file (on command) to back-annotate the schematic.
opt_0.DAT opt_1.DAT	Generated by the PSpice Optimizer for each simulation in a run	Interim data files. Contain simulation results which are read by Probe to evaluate performance.
.PAR	Generated by Capture when activating the PSpice Optimizer and updated by the optimizer when running an optimization	Parameters file. Contains the latest optimization parameter values.

* You can use any file extension for an external data file.

Optimizing a netlist-based design

C

Appendix overview

This appendix describes how to set up and run an optimization for a design defined in a circuit file.

- Optimizing without a schematic on page C-164 gives an overview on optimizing a netlist-based design.
- Setting up the circuit file on page C-165 describes the steps that you must complete using a text editor to parameterize the circuit file for optimization.
- Setting up and running the PSpice Optimizer on page C-166 describes the steps that you must complete using the PSpice Optimizer to define parameters and specifications, run the optimization, and save the results.
- Example: Parameterizing the circuit file on page C-168 steps through parameterizing a simple diode biasing circuit file.

Optimizing without a schematic

Although the PSpice Optimizer is designed to optimize a Capture-based design, you can optimize a design for which a netlist, but no schematic, is available.

To optimize your design without using Capture:

1. Implement your design as a circuit file in PSpice-compatible format.
2. Create a separate file containing the optimization parameter definitions.
3. Using the PSpice Optimizer, complete the setup and run the optimization as usual.

Refer to your PSpice A/D user's guide for any questions you might have on circuit-file syntax. The remaining sections explain what you need to do once your design is defined as a circuit file.

Setting up the circuit file

To use the PSpice Optimizer to optimize a design defined as a netlist:

1 Decide on the design parameters you want the optimizer to vary.

2 Place parameter definitions (.PARAM) in a separate parameters file (.PAR).

3 Include the parameters file in the circuit file using the .INC command.

4 Replace component values that are dependent on the parameters with expressions.

All of these steps use standard PSpice A/D syntax; see your PSpice A/D user's guide for details. The following section explains why you need to set up a separate parameters file, and how to do it (steps 2 and 3 above).

The parameters file (.PAR) When optimizing, the PSpice Optimizer uses a separate parameters file to track changes to parameter values with each iteration. This file contains nothing but PSpice A/D parameter definitions reflecting the last-used values.

The optimizer automatically looks for a file with the name *circuit_file_name*.PAR.

Example: If the design is contained in MYAMP.CIR, the optimizer writes updated parameter definitions to MYAMP.PAR.

To create a parameters file for a design:

1 Start any text editor.

2 For each parameter in the circuit, enter a line using the syntax:

.PARAM *parameter_name* = *starting_value*

3 Save the file as *circuit_file_name*.PAR.

To make the parameter definition in the .PAR file available to PSpice A/D:

1. Insert a .INC command anywhere after the first line of the circuit file using the syntax:

   .INC *parameters_file_name*

   where *parameters_file_name* is the same file in which you entered the .PARAM statements for the circuit.

   Example: To MYAMP.CIR, add the following statement anywhere after the first line in the file:

   .INC “myamp.par”

Setting up and running the PSpice Optimizer

Before optimizing, you need to define all of the parameters declared in the circuit file, and the goals and constraints. Setup is exactly the same as for a schematic-based design.

To complete setup and optimize the design

1. Start the PSpice Optimizer by doing one of the following:
   - From the Windows Start menu, choose the OrCAD program folder and then the PSpice Optimizer shortcut.
   - From the Windows Start menu, choose Run, enter the command line for OPTIMIZE.EXE, and click OK.
2. Add a parameter definition for each of the parameters used in the circuit file:
   a. From the Edit menu, choose Parameters.
   b. Click Add.

c Fill in the dialog box.

3 Add a specification definition for each goal and constraint:

a From the Edit menu, choose Specifications.

b Click Add.

c Fill in the dialog box.

In the Analysis frame, be sure to enter the name of the circuit file you parameterized earlier.

4 Run the optimization.

a From the Tune menu, choose Update Performance.

b From the Tune menu, choose Auto and click Start.

To save the optimization results:

1 Save the results to the optimization file:

a From the File menu, choose Save As.

b Enter the name of the optimization file (.OPT) to contain the setup and results information.

2 From the File menu, choose Report to create a PSpice Optimizer report file (.OOT) containing a readable summary of the setup and results information.

Example: Parameterizing the circuit file

The following circuit file (call it DBIAS.CIR) contains the netlist for a voltage source / resistor / diode series combination.

```
 * diode bias circuit
.LIB
V1 1 0 5v
R1 1 2 5k
D1 2 0 D1N914
.DC  V1  LIST  5V
.Probe
.END
```

Suppose you want to optimize the bias current in a voltage source / resistor / diode series combination. Specifically, your goal is to achieve a current of 1 mA through the diode. To realize this goal, you must vary the value of resistor R1 (call the variable value, or parameter, Rbias).

To parameterize the circuit file

1. Create the `dbias.par` file containing the parameter definition for Rbias using this syntax:

   ```
   .PARAM Rbias = 5k
   ```

2. Parameterize the circuit file as shown (added or changed items are in bold):

   ```
   * diode bias circuit
   .INC "dbias.par"      ; this line added
   .LIB
   V1 1 0 5v
   R1 1 2 {Rbias}        ; this line modified
   D1 2 0 D1N914
   .DC  V1  LIST  5V
   .Probe
   .END
   ```

   Now you are ready to start the PSpice Optimizer, complete setup, and run the optimization.

Index

D

E

F

G

I

L

M

N

O

P

R

S

T

U

W

X

Y