Crosstalk Analysis User's Guide

HyperLynx

December, 1999

HyperLynx
14715 NE 95th Street
Redmond, WA 98052
support@hyperlynx.com

Table of Contents

Chapter 1: Overview of LineSim and BoardSim Crosstalk

Summary

This chapter gives a brief overview of LineSim's and BoardSim's crosstalk options.

For LineSim Crosstalk, it describes:

- how the crosstalk option extends the base LineSim product to allow crosstalk, coupled-transmission-line, and differential-signal analysis
- how Chapters 2-5 of this manual (the ones covering LineSim Crosstalk) is organized
- two different ways in which you can learn to use the crosstalk option

For BoardSim Crosstalk, it describes:

- how the crosstalk option extends the base BoardSim product
- for what kinds of simulation BoardSim Crosstalk is important
- recommended ways to use BoardSim Crosstalk's features
- how Chapters 6-9 of this manual (the ones covering BoardSim Crosstalk) is organized
- recommended minimum computer requirements for BoardSim Crosstalk

Overview of LineSim Crosstalk

How the Crosstalk Analysis Option Works with the Base LineSim Product

The “base” LineSim product (i.e., the standard product, without crosstalk capability) allows you to simulate arbitrarily complex schematics that contain virtually any kind of interconnect — generic transmission lines, lines referenced to a specific kind of PCB cross section, lines referenced to a complete PCB stackup, connectors, cables, and so forth. However, you cannot *couple* any of these lines together, i.e., you cannot simulate the effect of one line electromagnetically inducing currents and voltages on others.

Certain phenomena in high-speed digital systems arise from exactly this effect. *Crosstalk*, for example, occurs when a signal is intentionally driven down one conductor (board trace, cable wire, connector pin, etc.), but causes an unwanted signal to appear on another nearby conductor. The induced signal appears even though the two conductors are not conductively connected to each other.

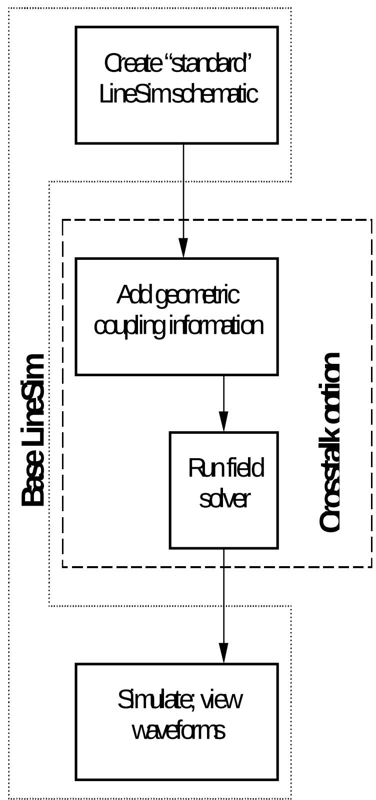
Differential signaling (increasingly used for high-speed data transfer) also makes use of electromagnetic coupling, although in this case, the coupling is intentional rather accidental. By placing two PCB traces, for example, in close proximity, you can ensure that noise externally induced on one of them also appears on the other (and is therefore rejected by a differential receiver). But placing traces in close proximity causes them to couple, and concepts that are key to differential signaling — like differential impedance — arise directly from this coupling.

In essence, what LineSim’s crosstalk option adds to the base LineSim product is the ability to add coupling to a “standard” LineSim schematic, i.e., information about how various transmission lines are coupled together. You supply this information geometrically (trace separations, stackup layer, trace widths and thicknesses, etc.), and LineSim automatically converts it to electromagnetic coupling parameters. That data, in turn, is used to include the effects of the coupling in simulation waveforms.

The conversion from geometric to electromagnetic data occurs by running a “field solver,” an analysis engine that uses the basic equations of electromagnetics ("Maxwell's equations") to calculate the properties of coupled conductors. This tool is built-in to LineSim's crosstalk option; it runs automatically when needed.

Figure 1-1 shows how LineSim's crosstalk option adds coupling analysis to the base LineSim product.

Figure 1-1: How LineSim's crosstalk option adds coupling analysis to base LineSim



How the LineSim Crosstalk Portion of this Manual is Organized

Chapters 2-5 of this manual are devoted to LineSim Crosstalk. Chapter 9 gives general background information regarding crosstalk and coupled transmission lines.

The following table gives more details:

Topic	Comments
How to add coupling to a schematic (normal mode) (Chapter 2)	Describes the mechanics of adding coupling to a schematic; most of what you'll ever need to know, unless your schematic become very complex; omits discussion of the "advanced" coupling mode (see below)
About the field solver (Chapter 3)	Describes what a field solver is and how it works; shows how to view the detailed results from the solver and plot field lines
How to add coupling to a schematic (advanced mode) (Chapter 4)	Describes an additional coupling feature ("coupling dots") that allows greater flexibility in drawing schematics; many users will never need this
Application examples (Chapter 5)	Gives detailed application examples, including "basic crosstalk," differential signals, how to minimize crosstalk on a bus, and "guard" traces
Technical background (Chapter 10)	Background information on crosstalk, coupled transmission lines, and differential pairs

How to Learn LineSim Crosstalk

If you prefer to learn by first reading about all of a tool's capabilities, then the following chapters are recommended:

- Chapter 2 (how to add coupling, in "normal" mode)
- Chapter 3 (how to run the field solver and view its results)
- optionally, Chapter 4 (advanced method of adding coupling)

If you prefer to learn by seeing application examples, see:

- Chapter 5 (detailed application examples)
- Chapter 2, 3, and 4 only as needed (to learn about features not described in detail in the examples)

If you want more background information on crosstalk, coupled transmission lines, and differential signaling, see Chapter 9.

Overview of BoardSim Crosstalk

What BoardSim Crosstalk Adds to the Base BoardSim Product

The “base” BoardSim product (i.e., the standard product, without crosstalk capability) allows you to simulate traces on an actual routed PCB, including all the details of each trace’s layout: individual metal segments, vias, pads, and so forth. However, you cannot analyze the effects of *coupling* between multiple traces, i.e., you cannot simulate the effect of one trace electromagnetically inducing currents and voltages on others.

Certain phenomena in high-speed digital systems arise from exactly this effect. *Crosstalk*, for example, occurs when a signal is intentionally driven down one board trace, but causes an unwanted signal to appear on another nearby trace. The induced signal appears even though the two traces are not conductively connected to each other.

Differential signaling (increasingly used for high-speed data transfer) also makes use of electromagnetic coupling, although in this case, the coupling is intentional rather accidental. By placing two PCB traces in close proximity, you can ensure that noise externally induced on one of them also appears on the other (and is therefore rejected by a differential receiver). But placing traces in close proximity causes them to couple, and concepts that are key to differential signaling — like differential impedance — arise directly from this coupling.

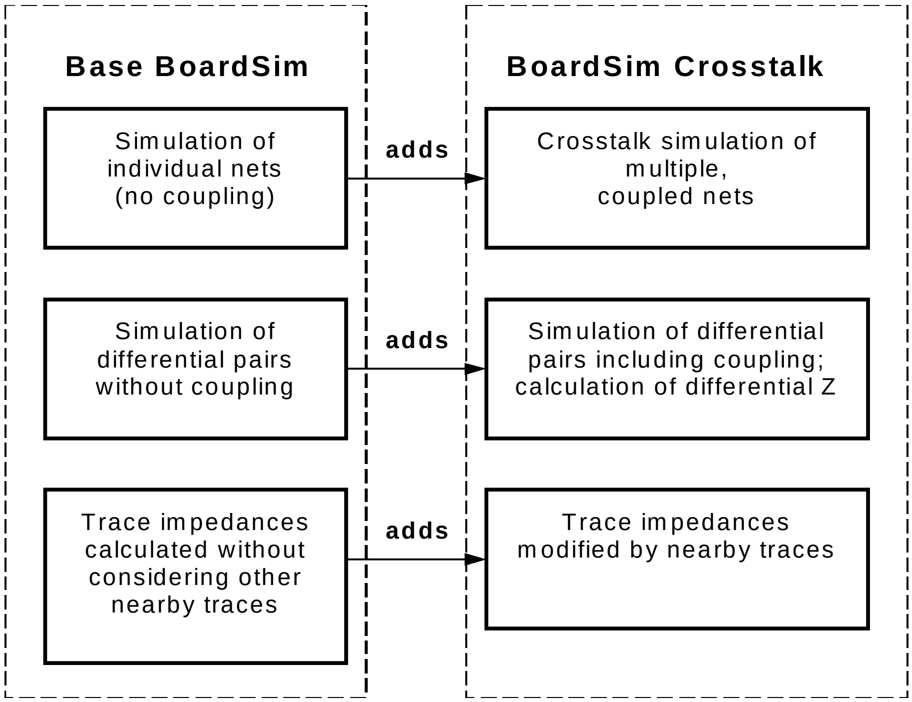
In essence, what BoardSim's crosstalk option adds to the base BoardSim product is the ability to include the effects of trace-to-trace coupling in simulations. As in base BoardSim, the electromagnetic modeling is automatic: you choose a net for analysis, and BoardSim Crosstalk does the difficult work of determining which other nets are significantly coupled to the chosen net. Then all of the involved nets are modeled and simulated together; the effects of any crosstalk or coupling between them appears automatically in the simulation results.

The calculation of exactly how traces are coupled to each other occurs by running a "field solver," an analysis engine that uses the basic equations of electromagnetics ("Maxwell's equations") to calculate the properties of coupled conductors. This tool is built-in to BoardSim Crosstalk; it runs automatically when needed. For maximum performance, the field solver has a caching mechanism which allows it to calculate a given cross section once and then store that section's solution for fast retrieval later, when needed again.

Applications Made Possible by BoardSim Crosstalk

BoardSim Crosstalk adds several important application abilities to the base BoardSim product. Figure 1-2 summarizes these; the sections following describe them in more detail.

Figure 1-2: Capabilities added to the base BoardSim product by BoardSim Crosstalk



Crosstalk Analysis

If you suspect that your PCB's traces may crosstalk with each other (because, for example, your routing density is very high or your driver ICs switch quickly), BoardSim's crosstalk option can show you whether the problem is likely to be serious or not, and help you find ways to reduce crosstalk if it does occur. In particular, you can perform any mix of the following kinds of crosstalk simulation:

- **interactive:** choose a "victim" net; BoardSim automatically finds likely aggressor nets; simulate to see exactly how much crosstalk will occur on the victim in your real system

- **quick batch-mode:** run a fast analysis on your entire PCB to estimate the maximum amount of crosstalk that could occur on each net; the results, summarized in a report file, serve as a guide to which nets you should examine in more detail
- **detailed batch-mode:** run detailed analysis on selected nets in a batch fashion (all nets, if you want, although choosing a critical subset is smarter); BoardSim automatically sets the driver IC on each victim net as "stuck high" and "stuck low" and permutes through min/typ/max settings for the drivers on aggressor nets; crosstalk-amplitude and timing results are written to a report file

For more details on how HyperLynx recommends mixing and using the features, see section "Recommended Way to Use BoardSim Crosstalk's Features" below.

Differential-Signal Analysis

When the base BoardSim product analyzes differential signals, it models each trace in the pair separately, without accounting for the coupling between the traces. This is adequate for loosely coupled pairs, but not if the traces are tightly coupled.

BoardSim Crosstalk adds the ability to automatically account in detail for the coupling between traces in a pair. This not only increases simulation accuracy, but also allows you to view directly the differential- and common-mode impedances of the pair. And it enables the Terminator Wizard to automatically calculate the resistor values needed to optimally terminate the pair.

Effects of Nearby Traces for Signal-Integrity Simulations

As PCBs become increasingly dense, there is a greater possibility that the behavior of individual traces is affected by the presence of other, nearby traces. For example, you may calculate exactly how to achieve 50-ohm trace impedances on your PCB (for example, by controlling dielectric thicknesses and trace widths), but then discover after fabrication that the proximity of nearby traces reduces your nominal impedance to 47 ohms.

When you simulate with the base BoardSim product, the trace you select for analysis (and nets electrically connected to it, i.e., associated nets) are simulated ignoring the effects of other nearby traces. Depending on your PCB's density, this may be perfectly sufficient. However, in some situations on dense PCBs, you may want to include the effects — e.g., impedance changes — due to neighboring traces.

BoardSim Crosstalk optionally allows these "proximity effects" to be included. It does so in the following ways:

- *in interactive simulation,* by allowing you to enable crosstalk simulation so that nearby coupled nets are included in simulations; you can set the driver ICs on these nets into a static, "stuck-high" or "stuck-low" state, while the driver IC on the selected net switches high or low
- *in batch-mode simulation,* by enabling a "high-accuracy" mode that automatically enables coupling and includes nearby, coupled nets in simulations; the driver ICs on these neighboring nets are automatically stuck low

For more details on these methods, see Chapter 6 and Chapter 8, section "Types of Crosstalk Simulation in the Board Wizard."

Performance Trade-Off in Including Other Traces

There is a performance trade-off involved in including the effects of neighboring traces in simulations: the simulations run slower. Therefore if you don't believe that these effects are likely to be significant (e.g., you don't care for a particular board about the last few ohms of impedance accuracy), you should avoid including other, coupled traces in your simulations.

To find out whether neighboring traces **are** affecting simulation results on a given board, you can compare a few representative interactive simulations with coupling turned on and then off, or view the coupled impedances directly in LineSim or BoardSim.

Recommended Way to Use BoardSim Crosstalk's Features

For Crosstalk Analysis

You can use any mix of BoardSim Crosstalk's interactive and batch-mode features, to suit the needs of your particular designs. (See section "Crosstalk Analysis" above for a brief introduction to the available analysis features.)

However, since simulating crosstalk is probably the most-complex of all types of signal-integrity analysis, your goal should always be maximum efficiency: focusing on the likely "problem" nets and getting results as quickly as possible. Therefore, **HyperLynx recommends the following approaches to analyzing a PCB's crosstalk:**

- **If you do not know which nets are likely to exhibit crosstalk, first generate a batch-mode Crosstalk Strength Report.** This is a very fast process that estimates the maximum amount of crosstalk for every net on your PCB, and sorts the nets in order by "most crosstalk" to "least." **Then, depending on the number of "problem" nets in the list, use the report as a guide to either interactive or detailed batch-mode simulation.** For details on the Strength Report, see Chapter 7; on interactive simulation, Chapter 6; on detailed batch-mode simulation, Chapter 8.
- **If you know which nets are likely to exhibit crosstalk, and the number of such nets is limited, use interactive simulation.** See Chapter 6 for details.
- **If you know which nets are likely to exhibit crosstalk, but the number of such nets is large, use detailed batch-mode simulation. You may first want to generate a batch-mode Crosstalk Strength Report, to find out which nets are the *most* susceptible to crosstalk.** For details on the Strength Report, see Chapter 7; on detailed batch-mode simulation, Chapter 8.

For Differential-Pair Analysis

If you are using BoardSim Crosstalk's coupling-analysis features mostly to simulate differential pairs, then you will probably make use mostly of interactive simulation. (BoardSim's batch-mode analysis features are oriented more strongly at crosstalk.) You may also want to use the Coupling-Region Viewer, because it allows you to check the differential impedances (and other electrical characteristics) of your trace pairs.

For details on interactive simulation and how to use the coupling-region viewer, see Chapter 6.

How the BoardSim Crosstalk Portion of this Manual is Organized

Chapters 6-8 of this manual are devoted to BoardSim Crosstalk. Chapter 10 gives general background information regarding crosstalk and coupled transmission lines. The following table gives more details:

Topic	Comments
How to run interactive crosstalk and differential-pair simulations (Chapter 6)	How to run net-by-net interactive simulations; most useful if you have a small set of critical nets to analyze, or for analysis differential pairs; includes details of how to see the field solver's output for a particular cross section (impedances, C and L, etc.)
How to run a quick-analysis batch-mode Crosstalk Strength Report (Chapter 7)	How to quickly generate a report estimating the amount of crosstalk likely to occur each of a PCB's nets; makes an excellent guide to which nets need further detailed analysis; runs very fast and requires no IC models.
How to view coupling regions and see field-solver results (Chapter 8)	Describes what a field solver is and how it works; shows how to view coupling regions and their associated field-solver data.
How to run detailed batch-	How to set up and run detailed crosstalk

Topic	Comments
mode crosstalk simulations (Chapter 9)	simulations for a set of nets; best for dealing with a set of nets too large for interactive analysis; outputs a report detailing SI, crosstalk, and timing data for each selected net.
Technical background (Chapter 10)	Background information on crosstalk, coupled transmission lines, and differential pairs .

Minimum Computer Requirements for BoardSim Crosstalk

Typical simulations in BoardSim Crosstalk are often much more CPU-intensive than simulations in the base BoardSim product. This is true for several reasons:

- simulation of electromagnetic coupling is more mathematically intensive than simulation without coupling
- coupled simulations involve multiple nets and driver ICs simultaneously (sometimes 10 or more); uncoupled simulations model only a single net and driver IC
- for coupled simulations, the field solver must run to calculate trace impedances
- building the simulator’s circuit model for a set of coupled nets involves more steps than building the model of an uncoupled net
- the trace segments in a set of coupled nets have to be more finely subdivided to achieve an accurate simulation than the segments in an uncoupled net

Because of BoardSim Crosstalk's intense CPU requirements, HyperLynx recommends running the product on a machine with at least the following specifications:

- 64 megabytes of RAM (or more)
- 300-MHz Pentium II processor (or faster)

Given the low price of today's high-performance PCs, if you plan to run a serious amount of crosstalk simulation, it makes sense to purchase a fast computer for the task. Doubling the speed of your PC, for example, could reduce the length of a crosstalk batch-mode simulation by multiple hours.

Chapter 2: Adding Coupling to a LineSim Schematic *(LineSim Crosstalk)*

Summary

Important! *This chapter is specific to the LineSim Crosstalk product; it does not apply to BoardSim Crosstalk. For detailed information about BoardSim Crosstalk, see chapters 6-9.*

This chapter describes:

- how to add coupling information to a LineSim schematic, so that LineSim runs crosstalk and differential-signal simulations
- what a "coupling region" is
- how to create a coupling region to model a particular coupled cross section

If you are more interested in application examples of common crosstalk scenarios (like long buses or differential pairs), see Chapter 5. For an overview of how LineSim approaches crosstalk and differential-signal simulation in general, see Chapter 1. This chapter describes the detailed mechanics of creating the "coupling regions" that model specific areas in which crosstalk occurs.

LineSim Crosstalk supports two modes of defining coupling regions, a "normal" mode that is simple to learn and use, but imposes a few restrictions on how your schematic can be drawn; and an "advanced" mode which is more complex but imposes no restrictions on your

schematics. This chapter describes the normal mode; Chapter 4 describes the advanced mode. For more details on the differences between these modes, see "'Advanced' versus 'Normal' Couplings" below in this chapter.

Adding Coupling to a LineSim Schematic

The process of setting up crosstalk or differential-pair simulation in LineSim Crosstalk involves basically two steps:

- creating a "normal" LineSim schematic with transmission lines, IC drivers and receivers, terminating resistors and capacitors, and other passive components…
- …then adding additional information about how certain transmission lines in the schematic are coupled together, e.g., which lines are coupled together, how far apart they are, over what distance they are coupled, and so forth

Adding the additional coupling information is called "creating coupling regions." Each region defines the coupling for a set of transmission lines. The definition is geometric; LineSim Crosstalk's field solver automatically runs to extract the electrical characteristics of the coupling.

This manual assumes that you already understand how to perform the first step, i.e., how to create basic, uncoupled schematics in LineSim. If not, refer to the LineSim User's Guide or the online Help for details.

This chapter describes the mechanics of the second step, i.e., how to create coupling regions. A few simple examples are given, but most of the emphasis is on the details of operating the relevant portion of the user interface, especially the Edit Coupling Regions dialog box.

If you prefer to learn by seeing application examples, then refer to Chapter 5, which has a number of complete examples showing how to model typical crosstalk or differential-signal applications using LineSim Crosstalk.

“Advanced” versus “Normal” Couplings

LineSim Crosstalk’s user interface supports two modes for defining coupling regions:

- a “normal” mode that imposes a few restrictions on how you can place coupled transmission lines in the schematic, but makes defining coupling regions simple and intuitive
- an “advanced” mode that allows more flexibility than the “normal” method, but also is more prone to entry error and requires you to understand more coupling concepts

By default, LineSim Crosstalk’s user interface assumes that you want to use the normal coupling mode. If you prefer to use the advanced mode instead, you can enable it in the Options/Preferences dialog box — Chapter 4 gives full details.

HyperLynx recommends that you use the normal coupling mode whenever possible. The advanced method is more complex, and if you choose to use it, you should first read and understand *all* of Chapter 4, particularly the concept of “coupling dots.” Most of the application examples in this manual are developed using the normal mode (see Chapter 5).

Requirements of the Normal Coupling Mode

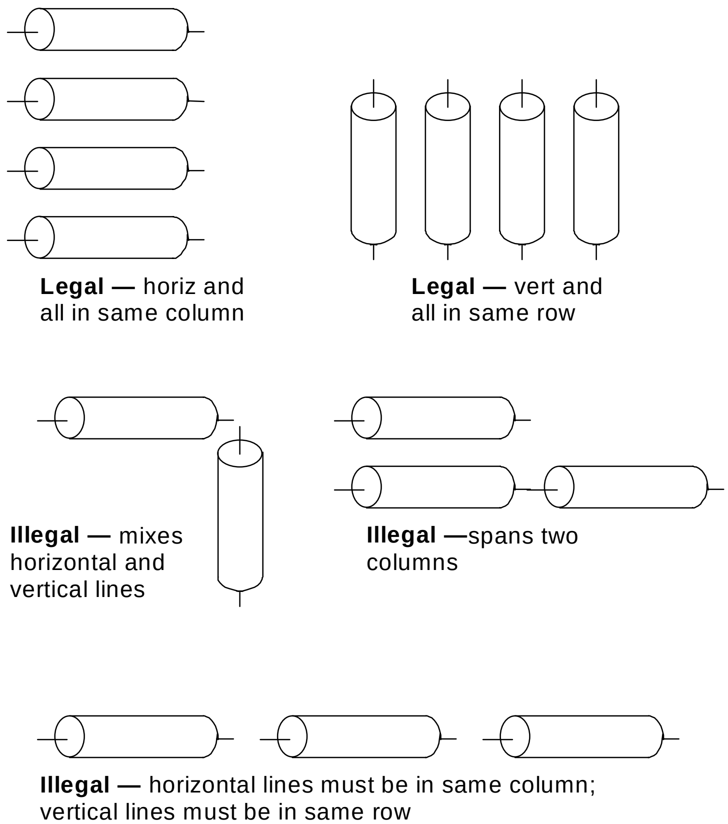
In the normal coupling mode, LineSim Crosstalk imposes one important restriction: that all of the transmission lines in a given coupling region must either be horizontal lines in the same schematic column, or vertical lines in the same schematic row (see Figure 2-1 for examples). The following conditions are not allowed in normal mode:

- mixing horizontal and vertical lines in one coupling region
- mixing lines from different rows or columns in one coupling region
- using vertical lines from the same column (rather than row) or horizontal lines from the same row (rather than column) in one coupling region

Again, these restrictions are absent if you switch LineSim into advanced coupling mode. See Chapter 4 for details.

Figure 2-1 shows some examples of what is and is not allowed in a schematic in normal coupling mode.

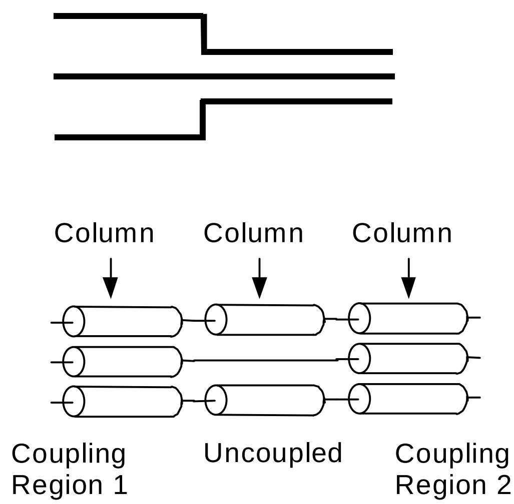
Figure 2-1: Legal and illegal groups of coupled lines, in "normal" coupling mode



If LineSim Crosstalk is running in normal coupling mode, then when you add a transmission line to a coupling region, only the regions to which it is legal to add the line are available for adding — or you can add the line to a new region. For details on adding lines to coupling regions, see “Creating or Choosing a Coupling Region for a Transmission Line” below.

Fortunately, most of the coupling scenarios that you would encounter on a real PCB (or in a cable or connector) map well to the normal mode’s method. For example, consider the trace sections shown in Figure 2-2 and the corresponding “legal” way of representing them in a normal-mode schematic. The way the normal mode encourages you to draw the schematic is the most logical way of doing it anyhow, i.e., most physical couplings lend themselves to a side-by-side-in-a-column or –row representation.

Figure 2-2: How a typical PCB’s trace coupling maps to a normal-mode schematic



What is a "Coupling Region"?

In LineSim Crosstalk, a "coupling region" defines how two or more conductors (usually PCB traces) are coupled together electromagnetically. Such a region occurs when the conductors travel near each other for some length, and the result is usually some amount of crosstalk between the conductors.

Coupling regions are inherently two-dimensional. They are defined in terms of a PCB-stackup cross section that includes the following information:

- the width of each conductor
- the position of each conductor relative to other conductors:
    - what stackup layer
    - separation from other conductors on the same layer, to the right and left
    - horizontal position relative to other stackup layers

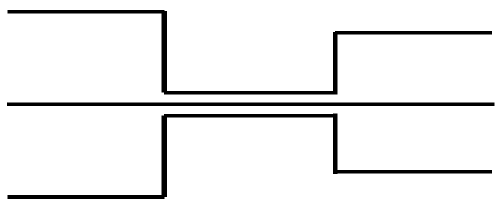
Then, the entire coupling region is assigned a length, which specifies over what distance the conductors are coupled in the way described in the cross section. (See "Example of a Coupling Region" below for details and pictures.)

Notice that while the goal of a coupling region is to define electromagnetic coupling, a region is actually specified geometrically: conductors of certain widths and thicknesses, residing on certain stackup layers, etc. It is LineSim Crosstalk's job to convert these geometric specifications into electromagnetic data that can be used to predict crosstalk voltages and currents. LineSim performs this conversion automatically by invoking a "field solver"; for details see Chapter 3.

Example of a Coupling Region

Suppose, for example, that you had three traces on a PCB that were routed far from each other for several inches; then came close together for several more inches; and then separated again before arriving at their individual destinations. (See Figure 2-3.)

Figure 2-3: Three traces on a PCB



Since the traces have three different separations from each other, you could consider that this routing defines three different coupling regions, as shown in Figure 2-4. Figure 2-5 shows what the cross section associated with Figure 2-4's "Region 2" might look like.

Figure 2-4: One way of defining coupling regions for Figure 2-3's traces

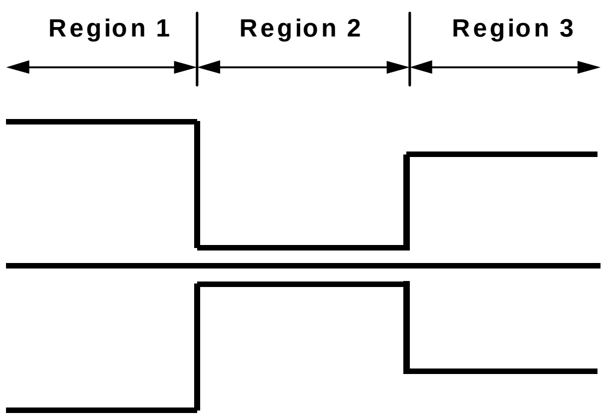
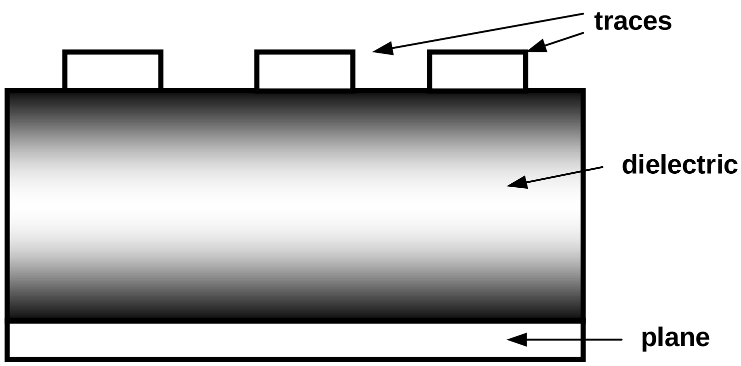
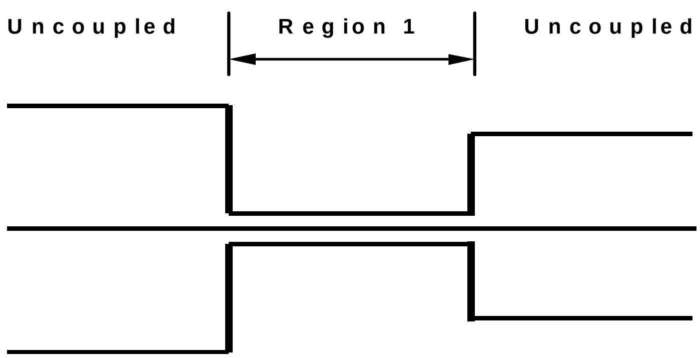


Figure 2-5: Possible cross section for Figure 2-4's "Region 2"



However, because the traces are far apart in Region 1 and Region 3, it's unlikely that any crosstalk will occur in those areas. Therefore, you would probably only bother to define a coupling region for the area in which the traces are close to each other, as shown in Figure 2-6.

Figure 2-6: More efficient way of defining coupling regions (only 1 needed)



Creating a Coupling Region

In LineSim Crosstalk, schematics can freely mix "regular," uncoupled transmission lines with coupled lines. Each coupled line must be a member of a coupling region that defines to which other lines in the schematic it is coupled, and geometrically how the coupling occurs. The lines in a given coupling region must conform to the simple restrictions imposed by LineSim's "normal" coupling mode; see "Requirements of the Normal Coupling Mode" above for details.

Note: *You can bypass these restrictions by switching LineSim's coupling mode to "advanced." However, there's often no need to do this. See Chapter 4 for details on advanced mode.*

For example, the schematic corresponding to Figure 2-6 above would include some uncoupled transmission lines representing the trace sections in the "Uncoupled" areas, and some coupled lines representing the sections in "Region 1."

This section describes how to create a coupling region that causes the transmission lines associated with it to be coupled in a specific way.

Basic Steps for Creating a Coupling Region

To create a coupling region in LineSim Crosstalk, follow these basic steps (see below for more details on each individual step):

1. First, add to the LineSim schematic the transmission lines that you plan to include together in a coupling region. Remember that if LineSim Crosstalk is in its "normal" coupling mode, all of the lines must be horizontal and in the same schematic column, or vertical and in the same row. (If you're unfamiliar with how to add objects to a schematic, see the LineSim User's Guide. For details on the normal and advanced coupling modes, see "'Advanced' versus 'Normal' Couplings" above.)

2. Next, right-click one-by-one on the transmission lines, and for each line:

   — Change the transmission-line type to "coupled stackup."

— Choose a coupling region for the line.

3. Once step 2 has been completed for each transmission line in the coupling region, set the geometric cross-section properties for the lines in the region (how far apart they are, what stackup layer they're on, and so forth).

4. Then proceed with analysis: run a simulation, view the field solver's impedance report, and so forth.

Changing a Transmission Line to be Coupled

The first step in defining a coupling region is to change the type of two or more transmission lines to be "coupled."

Note: *You can actually build a coupling region that contains only one transmission line, although normally you wouldn't bother. See "Modeling Single, Uncoupled Transmission Lines with a Coupling Region" for details.*

To change a transmission line's type to "coupled":

1. In the schematic, point to the transmission line with the mouse; a red highlight box appears around the line's symbol. **Right click** with the mouse. The Edit Transmission Line dialog box opens.

2. In the Transmission-Line Type list, **under the Coupled heading,** click the Stackup radio button. (Be sure to click the Stackup button under the Coupled heading, on the right, rather than under Uncoupled on the left. Clicking the button under Uncoupled will leave the transmission line without coupling.)

3. As soon as you click the radio button, the Add to Coupling Region tab appears and you are automatically jumped to it. LineSim Crosstalk is now asking you to place the transmission in a specific coupling region; continue with the steps in "Creating or Choosing a Coupling Region for a Transmission Line" below.

Creating or Choosing a Coupling Region for a Transmission Line

Creating a New Coupling Region

When you first begin changing transmission lines in a schematic to be of type “coupled stackup” (see “Changing a Transmission Line to be Coupled” above for details), there will be no coupling regions defined into which to place the lines — so you need to create a new region. Likewise, once some coupling regions exist, you may need to define a new one to accommodate additional transmission lines that you make coupled.

To create a new coupling region:

1. First, verify that you are in the Edit Transmission Line dialog box, with the Add to Coupling Region tab selected (see “Changing a Transmission Line to be Coupled” above for a description how to get to this tab when you first make a line coupled).

2. Then, in the Coupling Regions list, either double-click on the entry New-Coupling, ***OR*** highlight New-Coupling and click OK, ***OR*** highlight New-Coupling and click on a different dialog-box tab. LineSim Crosstalk automatically creates a new coupling region with a default name “CouplingXXXX” (where XXXX is a number) and adds the transmission line to the new region.

If you double-click or click OK, the line is added and the dialog box closes. If you click another dialog-box tab, the line is added, but the box stays open and you can perform editing, etc.

Choosing an Existing Coupling Region for a Transmission Line

Whenever you change a transmission line’s type to “coupled stackup” from some other type, you have the option of placing the line into an existing coupling region or creating a new one. For details on creating a new coupling region, see “Creating a New Coupling Region” above. This section describes how to add to an existing region.

To place a transmission line into an existing coupling region:

1. First, verify that you are in the Edit Transmission Line dialog box, with the Add to Coupling Region tab selected (see “Changing a Transmission Line to be Coupled” above for a description of how to get to this tab when you first make a line coupled).

2. Then, in the Coupling Regions list, look for the name of region to which you want to add the transmission line. (The line that gets added is the one on which you right-clicked in the schematic to get into the dialog box.) The list allows you to choose only those regions which are legal for the line to be added to; other “illegal” regions are excluded from the list. (For details on what is meant by a “legal” region, see “Requirements of the Normal Coupling Mode” above.) If the only name in the list is “New-Coupling,” there are no existing legal coupling regions and you must create a new one; see “Creating a New Coupling Region” above for details.

3. In the Coupling Regions list, either double-click on the existing coupling region’s name, ***OR*** highlight the name and click OK, ***OR*** highlight the name and click on a different dialog-box tab. LineSim Crosstalk automatically adds the transmission line to the region.

If you double-click or click OK, the line is added and the dialog box closes. If you click another dialog-box tab, the line is added, but the box stays open and you can perform editing, etc.

Coupled Transmission Lines in the Schematic Editor

How Coupled Lines Highlight

Once a transmission line has been added to a coupling region, it is displayed differently in the schematic editor than when it was uncoupled. In the schematic, when you point with the mouse to a “normal” uncoupled line, a red highlight box appears around the line’s symbol. When you point to a coupled line, both a red and a yellow box appear; the yellow box indicates “coupled line.”

Coupling “Ratsnest”

In addition to highlighting coupled transmission lines in yellow, LineSim Crosstalk’s schematic editor also shows to which other lines a given line is coupled. When you point to a line that shares a coupling region with other lines, the other lines in the region also highlight in yellow, and all of the transmission lines are joined by dashed yellow “ratsnest” lines.

Forcing All of the Ratsnest Lines “On” in the Schematic

Normally, coupling ratsnest lines are visible only when you point to a coupled transmission line with the mouse. However, to quickly see all of the couplings in the schematic, you can also force all of the ratsnest lines to be “on.”

To force all of the coupling ratsnest lines “on”:

1. From the View menu, choose Show Coupling Regions.
   OR
   Click the Show Coupling Regions button on the toolbar.

   The coupling ratsnest lines associated with every coupling region in the schematic become visible. Each coupled transmission line is also highlighted with a yellow box.

The ratsnest lines print and copy to the Windows Clipboard along with the rest of the schematic.

“Coupling Dot”

One other way you can tell that a transmission line in the schematic is coupled is by the appearance of a “coupling dot.” This symbol — a small round dot that appears at the end of the transmission line — indicates that the line is electromagnetically coupled to other lines. It also specifies *which end* of the line is coupled to *which other ends* of the other lines in the coupling region (similar to the way dots are used in transformer-winding symbols).

Note: *With LineSim Crosstalk running in the normal coupling mode, the dots are always at the left ends of horizontal transmission lines and the top sides of vertical lines. Because of the drawing restrictions imposed in normal mode, these are the only dot positions that make sense. If you need to move a coupling dot, so that a different end of some transmission line is coupled to the dotted end of another line, then you must switch LineSim Crosstalk into “advanced coupling mode.” See Chapter 4 for details.*

Removing a Coupled Transmission Line from the Schematic, then Restoring It

Suppose a transmission line is coupled (see the preceding sections for a description of how to make a line coupled), but then you remove the line from the schematic (by left-clicking on it). If you subsequently click the line back into the schematic, LineSim Crosstalk will remember which coupling region the line was in previously, and automatically restore it to that region.

However, if you remove a line from the schematic; edit the coupling region it previously was in and change some cross-section parameters; then click the line back in, LineSim Crosstalk will add the line back into the region, but may not be able to add it in exactly the way it was before (owing to the edits you made). In such a case, be sure to check the line's parameters before proceeding.

Changing a Coupled Transmission Line to be Uncoupled

Suppose a transmission line is coupled (see the preceding sections for a description of how to make a line coupled), but then you edit the line and change it from type “Coupled Stackup” to an uncoupled type. If you subsequently make the line once again coupled, you must manually add it back into a coupling region and set its cross-section properties — LineSim Crosstalk does not remember anything about how a line was coupled once you uncouple it.

Modeling Single, Uncoupled Transmission Lines with a Coupling Region

Although normally you would use a coupling region only to model a cross section containing two or more traces, you can also a construct region that contains only a single trace. The reason to do this would be to get impedance data from the field solver, rather than the closed-form equations that LineSim normally uses for single-transmission-line cross sections.

LineSim's closed-form equations are quite accurate; for the typical cross-section geometries encountered on PCBs, the equation-based impedances are within a few percent of what the field solver would calculate. In fact, LineSim's equation solvers have been calibrated against the field solver, and "tuned" slightly to give the best possible matching.

For "unusual" cross-section geometries — e.g., very wide traces or very thin dielectrics — the deviation between the equation solvers and the field solver may grow larger. Thus, modeling such "atypical" single-trace cross sections with a coupling region may be advisable. Normally, however, there is little advantage in doing so.

Setting a Coupling Region's Cross-Section Properties

Once you've added to a coupling region one or more transmission lines, you can begin adjusting the geometric properties of the region's cross section. Recall that a coupling region essentially consists of a cross section plus a length which specifies over what distance the cross section applies (see "What is a 'Coupling Region'?" above for details).

To edit a coupling region's cross-section properties:

1. In the Edit Transmission Line dialog box, click on the Edit Coupling Regions tab. (To get into the Edit Transmission Line dialog box for a particular transmission line in the schematic, point to the line with the mouse and right-click on it.)

2. Begin editing the cross-section data in the tab's dialog box. See the descriptions below for details on how to change various parameters.

The cross section's properties are based on the following geometric parameters, which can be specified on a per-transmission-line basis. Note that for these parameters, lines are assumed to be PCB traces (or sections of a PCB trace):

- trace layer (i.e., where in the PCB stackup)
- trace width
- "X position" (i.e., horizontal justification on the stackup layer; all traces on the layer move in unison in response to this parameter)
- left- and right-side separations from other traces (or plane edges)

See the following sections for more-detailed descriptions of each parameter.

Before you can effectively edit transmission lines' properties, however, you must understand how the Coupling Region "tree" list works. The next section describes the list; the sections after it describe how to set various properties.

Coupling Region "Tree" List

When the Edit Coupling Regions tab is selected, a Coupling Region list box appears on the left. The purpose of this list is to present, in a hierarchical "tree" form, the contents of the coupling region for the transmission line through which you entered the dialog box. For instance, when you right-click on a line in the schematic and then select the Edit Coupling Regions tab, the tree list shows the contents of that line's coupling region. A coupling region's list includes the PCB stackup layers available for transmission lines in that region; inside each layer is a sub-list of the transmission lines (if any) that currently reside on that layer.

Plus/Minus Signs for Expanding/Collapsing List Items

The Coupling Region list works much like the "folders" tree list in the standard Windows Explorer. The coupling region itself and the stackup-layer items beneath it can be "expanded" so they show the items below themselves, or "collapsed" so they hide their sub-items. When expanded, a region or layer has a minus sign ('-') to its left; when collapsed, it has a plus ('+') sign. You can selectively expand and collapse the region or its stackup layers as needed to view the list's contents.

If a stackup layer in the Coupling Region list has no plus or minus sign to its left, then the layer has no transmission lines on it.

To expand an item in the Coupling Region tree list:

1. Click on the plus sign ('+') to the left of the item's name in the list.
   OR
   Double-click on the item's name in the list.

   The list of sub-items below the item appears, and the item's plus sign changes to a minus sign ('-') to show that the item is now expanded.

To collapse an item in the Coupling Region tree list:

1. Click on the minus sign ('-') to the left of the item's name in the list.
   OR
   Double-click on the item's name in the list.

   The list of sub-items below the item disappears, and the item's minus sign changes to a plus sign ('+') to show that the item is now collapsed.

How Transmission Lines are Labeled in the Tree List

Transmission lines are labeled in the Coupling Region tree list in two ways:

- with the designation "TL(xx:yy)", where "xx" is the schematic cell coordinate of the line's left end (if a horizontal line) or top side (if vertical), and "yy" is the coordinate of the line's right end or bottom side

- with the comment label (if any) that you have added to the transmission line

Adding a Comment Label to a Coupled Transmission Line

To add/change a comment label for a coupled transmission line:

1. In LineSim Crosstalk's schematic editor, right-click on the transmission line whose comment label you want to add/change.

2. With the Transmission-Line Type tab selected, in the Transmission-Line Properties area, type the desired label in the Comment edit box. The full name of the transmission line, including the "TL(xx:yy)" prefix, appears above in the Name field. Then click OK.

The comment label appears adjacent to the "TL(xx:yy)" label in the Coupling Region tree list, and on the transmission line's symbol in the schematic editor.

Transmission-Line Order on a Layer

When a stackup layer is expanded in the Coupling Region tree list, the transmission lines that appear below the layer are listed in order from the left-most line on the layer to the right-most. If you want to change the left-right ordering of lines on a layer, you move lines up and down in the list; see "Changing Left/Right Line Ordering on a Layer" below for details.

Graphical Display for Viewing Physical Coupling Region

Below the Coupling Region tree list is a graphical viewer that shows the physical cross section corresponding to the coupling region shown in the list. This viewer allows you to see the physical implications of changes you make in the tree list and to individual transmission-line properties. It updates real-time as you make coupling-region changes.

Zooming in the Graphical Viewer

By default, the graphical viewer shows the entire cross section for the current stackup. (For details on how to edit the stackup, see the LineSim User's Guide.) To better see how changes you make to the coupling region and its transmission lines affect the cross section, you may want to zoom the viewer in closer to the lines' traces.

To zoom the graphical viewer in closer:

1. Click on the Auto Zoom check box.

Transmission-Line Highlight in the Viewer

If you have a transmission line highlighted in the Coupling Region tree list, the corresponding trace is highlighted in the graphical viewer with a heavy black border. Conversely, if you have a trace highlighted in the viewer, the corresponding transmission line is highlighted in the tree list.

Also, when you point with the mouse in the viewer at a trace, the name of the trace's transmission line is displayed. (For details on how transmission lines are named, see "How Transmission Lines are Labeled in the Tree List" above.) This is a useful way of correlating traces in the viewer to transmission lines in the tree list. If the traces are too small to easily touch with the mouse, try zooming in closer with the Auto Zoom check box (see "Zooming in the Graphical Viewer" above).

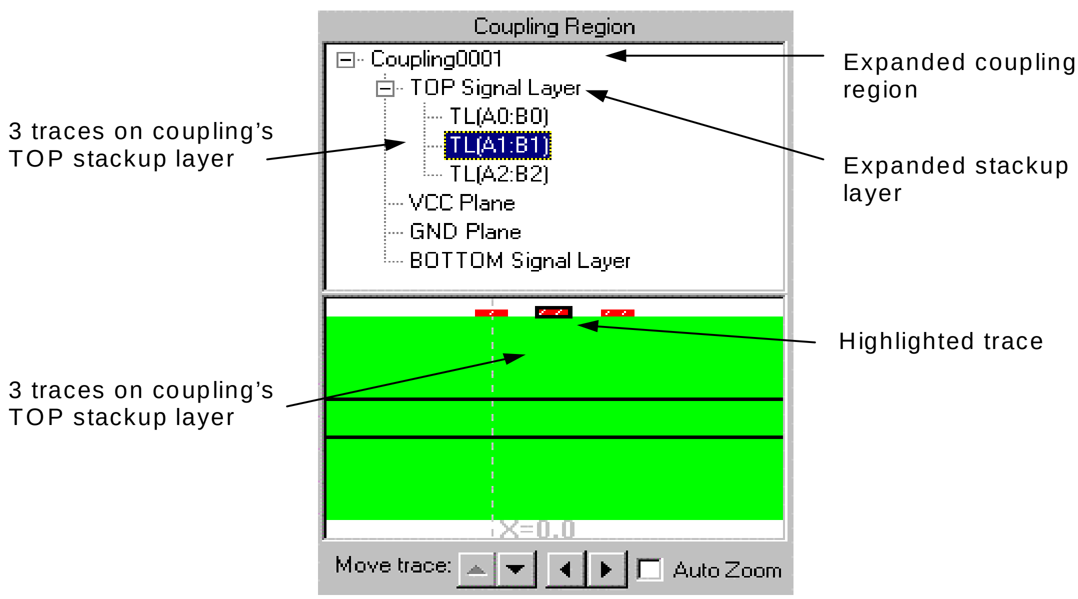
X=0.0 Indicator

One of the parameters in a coupling region which you can adjust is the horizontal or “X” position of the group of traces on each stackup layer. These positions are measured relative to an arbitrary horizontal “0” point. The graphical viewer shows this position with a light-gray dashed, vertical line labeled “X=0.0”. (For details on changing a layer’s X position, see “Changing X Position” below.)

Example of a Coupling Region Tree List and Graphical View

Figure 2-7 shows an example of a Coupling Region tree list and corresponding graphical view.

Figure 2-7: Example of a coupling-region tree list and graphical view



Cross-Section Properties

To set the properties of a particular transmission line in a coupling region:

1. With the Edit Coupling Regions tab selected, in the Coupling Region tree list, find the stackup layer which contains the transmission line. Expand the layer to show the list of lines below it, then click once to highlight the particular transmission line whose properties you want to change.
   OR
   In the Coupling Region graphical viewer, click on the trace corresponding to the transmission line. The trace's stackup layer automatically expands in the tree list, with the transmission line highlighted.

2. Then change some or all of the properties described in the following sections.

Changing Stackup Layer

The Layer parameter specifies on which layer in the PCB stackup the transmission line resides. The list of possible layers is determined by the current stackup in LineSim Crosstalk. (If you're unfamiliar with how to create and edit a stackup in LineSim, see the LineSim User's Guide.)

There are two ways to change a transmission line's layer, one by using the Layer combo box, and the other by using the Coupling Region tree list.

To change a transmission line's layer by using the Layer combo box:

1. In the Coupling Region tree list or graphical viewer, highlight the transmission line whose layer you want to change.

2. In the Transmission Line area, pull down the Layer combo box and choose the layer to which you want to move the transmission line.

To change a transmission line's layer by using the Coupling Region tree list:

1. In the Couplings Region tree list, highlight the transmission line whose layer you want to change.

2. To move the line to a stackup layer that is higher in the tree list, click the up-arrow button (located at the bottom of the dialog box, below the

graphical viewer) until the transmission line appears on the desired layer. To move the line to a layer that is lower in the list, click the down-arrow button.

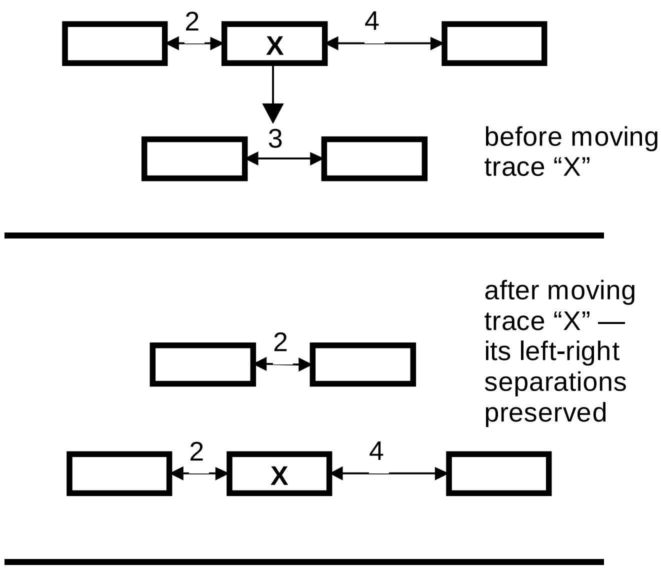
The graphical viewer updates to show the new layer position of the line.

How Transmission Lines are Positioned on a New Layer

When you move a transmission line to a new stackup layer, the line will have the same X position on its new layer as on its former layer. If there are other lines on the new layer that are “in the way” of the new line, the lines already occupying the layer will separate to accommodate the new line. Also, if there were transmission lines on the old layer that were separated by the moved line, they will “collapse” together to fill the void left by the moved line.

Figure 2-8 shows an example of moving a transmission line from one layer to the next, when lines on the destination layer are forced to move apart. In this figure, the line marked with an “X” is moved from the top layer to the layer immediately below. Notice that line X’s left- and right-side trace separations (2 and 4 mils) move with it to the new layer. (See “Changing Left- and Right-Side Trace Separations” below for details on setting trace separations.)

Figure 2-8: Example of moving a transmission line to a new layer



When transmission lines collapse together to fill a void left by another line that was just moved, the separation between the lines is the *smaller* of the two separations that are being merged. For example, in Figure 2-8, when the two lines remaining on the upper layer move together, they are separated by 2 mils, since of the possible separations of 2 mils and 4 mils, 2 is smaller.

Editing the Current Stackup

If you need to edit the current PCB stackup, you can open the stackup editor from within the Edit Coupling Regions dialog box.

To open the stackup editor from inside the Edit Coupling Regions dialog box:

1. In the Coupling Region area, click the Edit Stackup button. The stackup editor opens.

2. Make the desired changes to the current stackup, then close the stackup editor and continue working in the Edit Coupling Regions dialog box. The changes to the stackup are reflected in the Coupling Region tree list and graphical viewer.

Changing Trace Width

The Trace Width parameter specifies the cross-sectional width of the transmission line's trace. This is the same value that you specify in your PCB-layout tool for a trace's width.

To change a transmission line's trace width:

1. In the Coupling Region tree list or graphical viewer, highlight the transmission line whose width you want to change.

2. In the Transmission Line area, type the new value in the Trace Width edit box.

The graphical viewer updates to show the new width of the line.

Changing Left/Right Line Ordering on a Layer

To move a transmission line to the left or right of other traces that share its layer:

1. In the Coupling Region tree list or graphical viewer, highlight the transmission line whose ordering you want to change.

2. To move the transmission line to the left of its current position, click repeatedly on the left-arrow button (located at the bottom of the dialog box, below the graphical viewer), until the line's position on the layer reaches the location you want. To move a line to the right of its current position, use the right-arrow button.

When you click the left- or right-arrow button, the highlighted transmission line "hops" in the desired direction, swapping position with the next line in that direction. The graphical viewer updates each time you click; watch the position of the black-bordered (i.e., highlighted) trace in the viewer.

How Transmission Lines are Positioned After Moving Left/Right

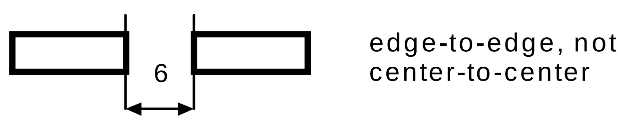
When a transmission line is moved left or right, its left- and right-side trace separations move with it, so that its new neighboring lines are properly separated from it. (See "Changing Left- and Right-Side Trace Separations" below for details on setting trace separations.) The neighboring lines shift position, as needed, to accommodate the moved line's separations.

Changing Left- and Right-Side Trace Separations

The trace-to-trace separation parameters define how far a given transmission line will separate from the traces (if any) on its left and right sides. **Separations are measured from trace edge to trace edge (not from center to center).**

For example, if a line has left and right separations of 6 mils, then if other lines are added to the coupling region and placed on the line's layer on either side of it, the line will automatically be separated from the others by 6 mils, edge-to-edge. Note that the left- and right-side separations may differ from each other, if desired. See Figure 2-9 for an example.

Figure 2-9: Example of how trace separation is measured



Specifying the distance between traces by separation values rather than by forcing you to calculate the equivalent center-to-center distances makes it easy to quickly construct coupling regions.

To change a transmission line's left- or right-side trace separation:

1. In the Coupling Region tree list or graphical viewer, highlight the transmission line whose left- or right-side trace separation you want to change.

2. In the Trace-to-Trace Separation area, in the Left or Right edit boxes, type the new value.

Note: *The trace-to-trace separation parameters control the separation of a transmission line from adjacent traces. If the line is on a stackup **plane** layer rather than a signal layer, and is bordered on either side by plane copper rather than other traces, the **trace-to-plane** separation parameters are used instead. See "Changing Left- and Right-Side Plane Separations" below for details.*

Changing Left- and Right-Side Plane Separations

As a transmission line is moved from layer to layer in a coupling region (see "Changing Stackup Layer" above for details), it can be placed on plane layers as well as signal layers. When a line is placed on a plane layer and is bordered on either side by the plane's copper (rather than another trace), its separation from the plane is governed by the trace-to-plane separation parameters (rather than the trace-to-trace parameters described in the preceding section).

The trace-to-plane separation parameters define how far a given transmission line will separate from a plane layer's copper. For example, if a line is moved to a plane layer and has left and right plane separations of 8 mils, the line will automatically be separated from plane copper by 8 mils, measured from the edge of the trace to the edge of the plane. Note that the left- and right-side plane separations may differ from each other, if desired.

To change a transmission line's left- or right-side plane separation:

1. In the Coupling Region tree list or graphical viewer, highlight the transmission line whose left- or right-side plane separation you want to change. Note that the line must currently reside on a plane layer; otherwise, its plane separations will be grayed out and unavailable.

2. In the Trace-to-Plane Separation area, in the Left or Right edit boxes, type the new value.

Changing X Position (Shifting an Entire Layer Right/Left)

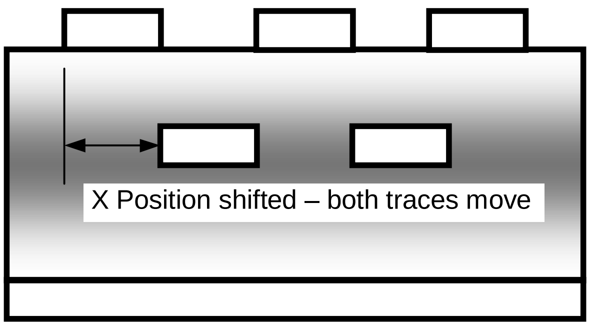
The X Position parameter specifies the distance from the center of a transmission line to an arbitrary horizontal "0" position on the line's stackup layer. (The X=0 point is shown visually in the graphical viewer as a light-gray dashed, vertical line.) However, since the distance between lines is controlled by the separation parameters (see the preceding sections), **X Position is intended only to affect the relative position of the *entire group* of traces on a given layer.**

Therefore, if you change a transmission line's X position, and there are other lines on the same layer, all of the lines will move with the one whose X position just changed, to keep the trace-to-trace separations constant. Effectively, then, X Position is a way of shifting to the right or left *all* of the lines on a layer.

Since LineSim Crosstalk assumes that the edge of the PCB is distant, it makes no sense to vary the X position of a group of transmission lines on one layer if there are no lines on other layers. However, if there *are* lines on multiple layers, then shifting one set of lines relative to another may be important to achieve the required geometry.

Figure 2-10 illustrates a case where X Position has been used to shift the traces on Layer 2 to the right, so that they are aligned in a staggered way versus the lines on Layer 1.

Figure 2-10: A use of the X Position parameter



To change a transmission line's X position:

1. In the Coupling Region tree list or graphical viewer, highlight the transmission line whose X position you want to change.

2. In the Transmission Line area, in the X Position edit box, type the new value.

Changing Coupling-Region Length

All of the parameters described in the preceding sections define the properties of a specific two-dimensional cross section. However, a coupling region's definition is not complete until a third dimension — its length — is added. The length specifies over what distance the region applies and therefore over what distance the region's transmission lines are coupled in the way described in the region's cross section.

When you specify the length of a coupling region, all of the transmission lines in the region take on that same length. E.g., if there are three lines in a coupling region and you set the region's length to 8 inches, all three of the transmission lines become 8 inches long.

To change a coupling region's length:

1. With the Edit Coupling Regions tab selected, in the Coupling Region area, in the Length edit box, type the new value. All of the transmission lines in the coupling region take on the new length.

Naming a Coupling Region

By default, when you first create a coupling region, LineSim gives it a name of "CouplingXXXX," where XXXX is a number. Each time you create a region, the number is incremented; if you save the schematic, re-load it, and create additional couplings, the number is restored and the incrementing begins from the previous value.

However, you can change the name of a coupling region any time you want from its default to something more meaningful for your particular application. The name is used in the Couplings Region list box (when you are choosing a region to which to add or move a transmission line), and in the name of the numerical-results report file generated by LineSim's field solver (see Chapter 3, section "Generating a Report of the Field Solver's Numerical Results") for details.

To name a coupling region:

1. With the Edit Coupling Regions tab selected, in the Name edit box, type the desired name.

Moving a Transmission Line from One Coupling Region to Another

Once a transmission line is in a coupling region (see “Creating or Choosing a Coupling Region for a Transmission Line” for details on first placing a line in a region), you can move it to another, different region. One way of doing this is to change the line back to type “uncoupled stackup,” then make it once again “coupled” and place it in a new region. However, LineSim Crosstalk gives you another, easier way:

To move a transmission line from one coupling region into another:

1. In the schematic, point with the mouse to the transmission line that you want to move; red and yellow highlight boxes appear around the line’s symbol (the yellow box indicates that the line is already coupled). **Right click** with the mouse. The Edit Transmission Line dialog box opens.

2. Click on the Move to Coupling Region tab.

3. In the Coupling Regions list, find the name of the coupling region to which you want to move the transmission line, or the entry “New-Coupling” if you want to move the line to a new region. Then either double-click on the region’s name, ***OR*** highlight the name and click OK, ***OR*** highlight the name and click on a different dialog-box tab. LineSim Crosstalk automatically moves the transmission line to the requested region.

When you click on the Move to Coupling Region tab, the Coupling Regions list box shows only those regions into which it is legal to move the selected transmission line. You may therefore not find all of the coupling regions in your schematic in the list box.

Whether or not there are some coupling regions that are “illegal” for the selected transmission line to be moved into depend on whether LineSim Crosstalk is in normal or advanced coupling mode. If you are running in normal mode, there may well be some regions which are not legal for a given transmission line; see “Requirements of the Normal Coupling Mode” above for details. If you are running in advanced mode, then all coupling regions should be available for moving.

Default Transmission-Line Positions

When you first add a transmission line to a coupling region, the layer on which it appears depends on the state of the transmission line before you added it to the region:

- if the transmission line was previously of type “stackup” (so that the line was already assigned to a layer), it appears in the coupling region on the same layer to which it was previously assigned
- if the transmission line was previously *not* of type “stackup,” it appears in the coupling region on the top layer

On whichever layer the newly added transmission line appears, the line is positioned to the right of the group of lines (if any) already existing on that layer. The separation from the previous right-most transmission line and the new line is determined by a default trace-to-trace separation that you can specify (see “Setting the Default Trace-to-Trace and Trace-to-Plane Separations” below for details).

Once a transmission line has been added to a coupling region, you are free to move the line to any position in the stackup. You can change the line’s layer, move it relative to the other traces on the same layer, and so forth. For details on moving a transmission line once it is added to a coupling region, see the preceding sections.

Setting the Default Trace-to-Trace and Trace-to-Plane Separations

You can set the default trace-to-trace and trace-to-plane separations used when a transmission line is first added to a coupling region. You would typically set these values before constructing your schematic to whatever value will be most common on your PCB.

For example, if you think your traces will likely be “6 and 6” (i.e., 6 mils wide and separated by 6 mils), set the default trace-to-trace separation to 6 mils.

Then all added transmission lines will default to being 6 mils apart and you won't need to change separation values as you simulate and experiment.

To set the default trace-to-trace separation:

1. From the Options menu, choose Preferences. The Options dialog box opens.
2. Click on the LineSim tab.
3. In the Default Trace Separations area, in the Trace to Trace edit box, type the desired default value. Click OK.

To set the default trace-to-plane separation:

1. From the Options menu, choose Preferences. The Options dialog box opens.
2. Click on the LineSim tab.
3. In the Default Trace Separations area, in the Trace to Plane edit box, type the desired default value. Click OK.

After you change one of these values, the new value doesn't take effect until you re-load this existing schematic, or begin editing a new one. For this reason, you should try to set the default separation values at the beginning of your LineSim Crosstalk session.

Chapter 3: Running the Field Solver in LineSim *(LineSim Crosstalk)*

Summary

Important! *This chapter is specific to the LineSim Crosstalk product; it does not apply to BoardSim Crosstalk. For detailed information about BoardSim Crosstalk, see Chapters 6-9.*

This chapter describes:

- what a “field solver” is
- how LineSim Crosstalk’s field solver works
- how LineSim Crosstalk uses its field solver to calculate impedances, delays, and other electrical parameters of coupled transmission lines
- how to view a coupling region’s field lines graphically
- how to view and save a coupling region’s electrical parameters

The emphasis in this chapter is on how to use LineSim Crosstalk’s field solver. For technical background on coupled transmission lines and the electrical parameters that describe them, see Chapter 9.

What is a “Field Solver”?

A field solver is a program that can solve for the electrical characteristics of a system of conductors and dielectrics, using one or more of the basic equations of electromagnetic theory (“Maxwell’s equations”). Specifically, LineSim Crosstalk uses its field solver to solve for the capacitances, inductances, propagation velocities, and characteristic impedances of a coupling region’s cross section. (For details on what is meant by “coupling regions” and how to define them in LineSim Crosstalk, see Chapter 2.)

Because coupling regions consist of two-dimensional cross sections that are assumed to be constant over some specified length, LineSim Crosstalk’s field solver needs to work in only two dimensions. Taking advantage of this fact allows LineSim to calculate coupling parameters accurately, but also very quickly — in fact, interactively, as you work.

Note*: Three-dimensional electromagnetic solutions become important only if the frequencies of the signals traveling on a system of conductors is so high that the wavelengths of the signals’ components are shorter than the various conductor structures in the system (e.g. vias, corner bends, etc.). This condition rarely occurs on PCBs carrying digital signals, so tools that analyze digital PCBs use two- rather than three-dimensional solvers. The big gain for users is speed: solvers run much faster in two dimensions than in three.*

When more than one transmission line is present in a coupling region, the various electrical parameters of the system take on a *matrix* form. For example, for a two-trace coupling region, there is no longer a single value of capacitance that describes the region’s cross section. Rather, there exists a 2x2 matrix which specifies both the capacitances of the individual traces to ground, and the capacitance between the traces.

The matrix nature of the electrical parameters describing a multi-trace coupling region is unfamiliar to many engineers and designers. For some detailed background information on coupled transmission lines and how they are described in matrix form, see Chapter 9. The remainder of this chapter discusses matrix parameters as needed, but concentrates mainly on how to use LineSim Crosstalk’s field solver, rather than the theory underlying it.

It is worth noting that there is no need to understand any of the electromagnetic details in order to successfully use LineSim's crosstalk-analysis features. You can enter all of your problems geometrically, let LineSim Crosstalk's field solver take care of the electrical details automatically, and get results in the form of waveforms and report files. Even a parameter like differential impedance is calculated automatically to prevent you from having to know how to calculate it from a characteristic-impedance matrix.

How LineSim Crosstalk's Field Solver Works

Note: *The information in this section is provided only for readers who are curious about what techniques LineSim Crosstalk's field solver uses to perform calculations. This material is not needed to successfully use LineSim's crosstalk-analysis features, and can readily be skipped.*

In order to completely determine the electromagnetic properties of a coupling region's cross section, LineSim Crosstalk's field solver must calculate the capacitance and inductance matrices for the cross section. These matrices give the conductor-to-ground and conductor-to-conductor capacitances and the self and mutual inductances of the traces in the coupling region.

To calculate capacitance values, LineSim Crosstalk's field solver finds the solution to Laplace's equation, a form of one of Maxwell's basic equations of electromagnetics:

$$\nabla^2 V = 0 \qquad (\textit{subject to all applicable boundary conditions})$$

In the solution, the solver seeks to find charge densities on the conductor surfaces and dielectric boundaries, rather than bothering to calculate the electric potential at all points between the conductors. **This approach makes LineSim's field solver a "boundary-element" solver**. Several proprietary methods are used to speed calculations significantly while maintaining a high level of accuracy.

The solution to Laplace's equation occurs subject to all of the boundary conditions specified in the coupling region's cross section, i.e., it takes into account the exact shapes and locations of the conductors in the region and the locations and material properties of the dielectric boundaries. Special care is taken to calculate charge density accurately in regions in which it changes rapidly (e.g., at the corners of conductors).

Once the coupling region's capacitance values are found, then to calculate the inductance matrix, the field solver takes advantage of the following equation from transmission-line theory:

$$LC = \frac{\varepsilon}{c^2}, \quad \text{or} \quad L = \frac{\varepsilon}{Cc^2} = \frac{1}{C_0 c^2}$$

This allows a second solution to Laplace's equation — one in which all of the dielectrics are replaced by vacuum and the capacitance matrix C_0 is found — to substitute for an explicit calculation of the coupling region's magnetic properties.

Once the capacitance and inductance matrices are both known, then the region's propagation speed(s) and characteristic impedances can be calculated. For the case of inhomogeneous dielectrics (i.e., a mixture of dielectric constants, as occurs with microstrip and buried-microstrip traces), multiple propagation speeds exist. These speeds are found from the eigenvalues of the matrix product LC.

How the Field Solver Runs in LineSim Crosstalk

In LineSim Crosstalk, the field solver's job is to calculate the following information for every coupling region:

- capacitance matrix
- inductance matrix
- characteristic-impedance matrix
- propagation speed(s)

- if multiple propagation speeds exist, the percentage of energy in each trace traveling at each speed
- an optimal resistor termination array for the region's transmission lines

(For background information on why many of these quantities are described in matrices, and what is meant by "multiple propagation speeds" and "optimal resistor termination array," see Chapter 9.)

Note that this information is calculated from the purely geometric and material data you provided in specifying each coupling region's properties. (For details on creating and specifying coupling regions, see Chapter 2.) Therefore, the field solver can be thought of as a calculation engine that transforms geometric/material data into corresponding electromagnetic data.

How the Field Solver's Results are Displayed

In LineSim Crosstalk, uncoupled transmission-line impedance and delay values are displayed explicitly in the schematic editor, inside each transmission line's symbol. They are also shown in the Edit Transmission Line dialog box, in the Values tab.

With coupled transmission lines, LineSim Crosstalk attempts to display electrical information in much the same way as with uncoupled. This is not entirely possible, because the information associated with a collection of coupled lines is more complex than the single-value parameters associated with uncoupled lines. For coupled lines, some information is displayed in the schematic editor; some is shown in the Edit Coupling Regions dialog box; and full details are available from the Field Solver dialog box. The following sections give details on the summary results in the schematic editor and Edit Coupling Regions dialog box. For information on detailed results, see "Viewing Detailed Field-Solver Results" below.

Field-Solver Results in the Schematic Editor

For coupled transmission lines, there is no single value that describes each line's characteristic impedance. Similarly, if the lines are microstrips or buried microstrips (rather than striplines), so that the electromagnetic fields they generate lie in a mixture of dielectrics (e.g., FR-4 and air), then multiple propagation velocities exist per line and there is no single line-delay value. (For details on matrix impedance and multiple propagation velocities, see Chapter 9.)

However, for coupled lines, LineSim Crosstalk does display a single value for impedance and delay in each transmission-line symbol. The values shown are as follows:

Parameter	**What is Displayed in a Transmission-Line Symbol in the Schematic Editor**
characteristic impedance	line's diagonal value from the characteristic-impedance matrix

delay	if line is a stripline (i.e., single dielectric):
	the single delay value
	
	if line is *not* a stripline (i.e., multiple dielectrics):
	weighted average of line's multiple delays; weighting based on percentage of energy traveling at each speed

Schematic Impedance

You can think of each transmission line's diagonal impedance as the impedance of the line to ground, accounting for the presence of the nearby, coupled lines. If the lines in the region are only weakly coupled, the diagonal value is close to what you would calculate for the line in isolation (i.e., ignoring the neighboring traces); as the lines become more strongly coupled, the diagonal impedance deviates more from the isolated value.

Although it is not possible to completely terminate a coupled transmission line with a single resistor (see Chapter 9 for details), if you are forced to use only one resistor and the signal on the line is not either purely differential or common-mode, then the diagonal impedance value is usually the best terminating value to use.

Schematic Delay

For coupled striplines, whose electromagnetic fields exist entirely in dielectric of one type, there is only one signal propagation velocity and therefore a single delay value, which the schematic editor displays.

However, for coupled microstrips or buried microstrips, whose fields penetrate both PCB dielectric and air, there are multiple propagation velocities (specifically, as many velocities as there are traces in the coupling region). In this case, in order to display a single delay value in the schematic editor, LineSim Crosstalk averages each line's multiple delays together. The calculation is a weighted average, with the weighting based on the percentage of signal energy that exists at each velocity. If only a small amount of energy travels at a given speed, then that speed's contribution to the average is small. The final result is displayed in the schematic editor.

Note: *Usually, unless a coupling region's geometry is very asymmetric, the difference between propagation velocities is small. (An example of an "asymmetric" geometry would be a microstrip of one width coupled to a buried microstrip of a different width, with the buried trace below and considerably off to the side of the outer-layer trace.) Therefore, the averaging effect described above is usually not major.*

For more details on multiple propagation velocities and why they occur, see Chapter 9.

Field-Solver Results in the Edit Coupling Regions Dialog Box

Some field-solver results are displayed directly in the Edit Coupling Regions dialog box, so you can monitor the effects of coupling-region geometry changes as you make them. Specifically, in the dialog box's Impedance area, the list box shows impedance data for the current cross section.

1. By default, the field solver recalculates impedances every time you make a change in the dialog box. You *can* optionally run with "auto-calculate" mode turned off, however; see "Auto-Calculate versus As-Needed Modes" below for details.

Exactly how impedances values are displayed varies depending on whether there are two or more than two traces in the coupling region.

Impedance Display

The list box in the Impedance area always shows the following information:

Column	**Description**
Transmission Line	the name of the transmission line; includes schematic-cell coordinates plus any "comment" you enter for the line
Impedance	an impedance value, in ohms; if multiple lines in the coupling region, taken from the diagonal values in the impedance matrix
Notes	a description of the what the impedance is

For details on getting more-complete impedance information (e.g., the full impedance matrix), see "Viewing Detailed Field-Solver Results" below.

Extra Impedance Information (Differential Z) when Only Two Traces

When there are only two traces in a coupling region, LineSim Crosstalk assumes that you may be interested in the differential impedance between the traces. (For details on what is meant by differential impedance, see Chapter 9.) Therefore, with two traces, the Impedance-area list box shows, in addition to the "diagonal impedances" described above, the two-trace differential impedance. This value is labeled "differential" in the Transmission Line column.

Note: *The differential impedance is the correct terminating value to use, line-to-line, only if the two traces in the coupling region carry only differential signals. Again, for details, see Chapter 9.*

Auto-Calculate versus As-Needed Modes

LineSim Crosstalk's field solver is designed to run fast enough that it is interactive, i.e., it can afford to be run whenever its results are needed.

Auto-Calculate Mode

If you are working on small coupling regions (i.e., regions with a small number of transmission lines) and if your computer is fast, you may want the field solver to run any time *any* change is made to a coupling region, even while you're in the middle of working in the Edit Coupling Regions dialog box. In this "auto-calculate" mode, to which LineSim Crosstalk defaults, the field solver runs each time you change a cross-section value anywhere in the dialog box.

To place the field-solver in auto-calculate mode (if previously disabled):

1. In the Edit Coupling Regions dialog box, in the Impedance area, click on the Auto Calc check box. The field solver runs and new results are placed in the Impedance list box.

If you are working on a large coupling region (i.e., one with many transmission lines) or if the field solver is taking several or more seconds to run each time it is invoked, then it is best to leave auto-calculate mode off.

As-Needed Mode

Optionally, the field solver can be set up to run whenever a coupling region's geometry or material data is changed, but *not* while you are working in the Edit Coupling Regions dialog box, in the middle of making changes. (For details on opening the Edit Coupling Regions dialog box and changing coupling-region properties, see Chapter 2, section "Setting a Coupling Region's Cross-Section Properties.") In this mode, the solver runs only when you close the dialog box or click the Transmission-Line Type tab.

To disable auto-calculate mode:

1. In the Edit Coupling Regions dialog box, in the Impedance area, click on the Auto Calc check box to disable it . The field solver runs and new results are placed in the Impedance list box.

If you want see the field solver's output data before you are ready to close the Edit Coupling Regions dialog box, you can force the solver to run. Running it will refresh the data displayed in the Impedance list box and also allow you to switch to the Field Solver tab and immediately click the Numerical Results View button.

To manually force the field solver to run, without closing the Edit Coupling Regions dialog box (only if auto-calculate mode is disabled):

1. In the Edit Coupling Regions dialog box, in the Impedance area, click the Calculate button. The field solver runs and new results are placed in the Impedance list box.

Viewing Detailed Field-Solver Results

What Detailed Results are Available?

The preceding section describe how to get *summary* results from the field solver, i.e., impedances and delays in the schematic editor and impedances in the Edit Coupling Regions dialog box. However, for every coupling region, the field solver has considerably more than this summary information available. The following table describes the additional information:

Information	Description
capacitance matrix	gives the self and mutual capacitances of the traces in the coupling region
inductance matrix	gives the self and mutual inductances of the traces
characteristic-impedance matrix	full matrix impedance for the system of coupled transmission lines; off-diagonal values are small for weak coupling and large for strong
optimal terminator-resistor array	an array of resistors (line-to-ground and line-to-line) that perfectly terminates the system of coupled lines; in theory, this array can completely eliminate crosstalk amongst the lines
list of propagation speeds	gives the velocities at which signals propagate on the traces; there multiple values if the traces' electromagnetic fields "see" more than one type of dielectric (e.g., microstrip or buried microstrip)

Information	**Description**
table of energy percentages in each propagation mode	gives the amount of each trace's energy that travels at each propagation velocity; *for multi-speed coupling regions only*
recommended termination values	list of impedance values, including the differential, common-mode, and line-to-ground values; *for two-trace coupling regions only*
graphical field lines	a picture showing the coupling region's electric field lines and electric equipotentials

The following sections describe how to generate and view this detailed information.

Viewing Graphical Field Lines

When a signal travels along a conductor, it transfers energy in the form of a "wave" that consists of time-changing electric and magnetic fields. The field solver's job, ultimately, is to predict these fields, given a specific cross section containing conductive traces and various dielectrics.

Sometimes it is informative to actually view the field lines calculated by the field solver. (Sometimes, too, they're just fun to look at.) LineSim Crosstalk allows you to graphically view the electric field lines for any coupling region. Though the field plots are of no direct analytic value, they can serve to give an intuitive feeling for how various traces are coupled to each other.

To view the field lines for a coupling region:

1. First, verify that you are in the Edit Transmission Line dialog box (if not, point in the schematic editor to a transmission line that is in the coupling region whose field lines you want to see, and right-click).

2. Then click the Field Solver tab. A large graphical view of the coupling region appears.

3. In the Field Plotting area, pull down the Propagation Mode combo box and choose the propagation mode for which you want to see field lines. (See "Choosing a Propagation Mode to Plot" below for details on this step. Or just leave the mode at its default setting and proceed to the next step.)

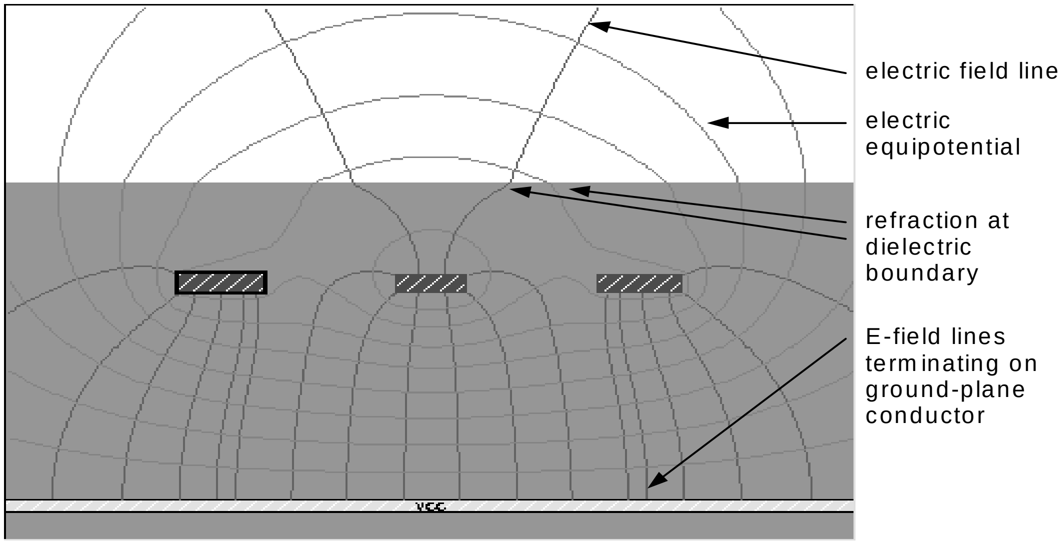
4. If you changed the propagation mode in step 3, plotting starts automatically. If you didn't change the mode, click the Start button. LineSim Crosstalk begins calculating and displaying electric field lines and electric equipotentials. (For details, see "What is Plotted" below.)

What is Plotted

In the graphical viewer, electric field lines are plotted in blue. These can be thought of as the electric field's "lines of force." Note that they begin and end on conductor surfaces (where physical charges reside). They refract (i.e., change direction) at boundaries between different dielectrics. See Figure 3-1.

Electric equipotentials are plotted in red. These are curves along which the electric potential (i.e., voltage) is a constant. They form closed contours around one or more conductors, and refract at dissimilar-dielectric boundaries. Again, see Figure 3-1.

Figure 3-1: An example of a field-line plot



The field lines are not plotted instantly, because LineSim Crosstalk calculates their positions on-the-fly, when you click the Start button (or change propagation mode). The plotting is fairly quick on most computers, but you can interrupt it before completion if you want.

To interrupt or stop field-line plotting:

1. In the Field Plotting area, click the Stop button. The plotting stops immediately.

Note: *The reason LineSim Crosstalk must calculate field-line positions as a post process is that its field solver is a boundary-element solver (see "How LineSim's Field Solver Works" above). To find a coupling region's capacitance and inductance matrices, the solver needs only calculate charge densities on conductor surfaces and dielectric boundaries. There is no information that needs to be calculated or stored about electric potentials in the regions between conductors and dielectric boundaries.*
However, to generate the field-line plots, these potentials are needed. They are

therefore found on-the-fly by solving certain differential equations which they must satisfy.

You can plot another propagation mode's field lines simply by changing the mode selection in the Propagation Mode combo box.

To plot a different mode's field lines:

1. In the Field Plotting area, pull down the Propagation Mode combo box and change from the current value to another. The plotting for the new mode begins automatically.

Choosing a Propagation Mode to Plot

What is a "Propagation Mode"?

Conceptually, the term "propagation mode" refers to a manner in which signals are arranged on a set of traces in order to propagate the signals. A "basis set" of propagation modes is a collection of modes that could be used in some mixture to create any arbitrary set of real signals on the traces.

For example, in the case of two traces coupled together, designers often think in terms of a set of modes consisting of "differential mode" and "common mode." The differential propagation mode is one in which if one trace carries the voltage +V, the other trace carries –V (i.e., the two traces always carry opposite voltages). The common mode is one in which if one trace carries +V, the other also carries +V.

Note that it is conceptually possible to describe *any* pair of real signals traveling on the two traces as some mixture of these two modes. For example, a mostly differential signal that had a small common-mode component to it could be constructed by mixing 80% differential mode with 20% common mode.

Note, too, that it is possible to conceive of other equally valid propagation-mode sets for the two traces. Another possibility, for example, is a set in which mode 1 consists of signal V on trace 1 and no signal on trace 2; and mode 2 consists of no signal on trace 1 and signal V on trace 2. This is a basis set just as valid as the set consisting of differential + common modes — i.e., you can conceptually use either set to construct any real set of signals on the traces.

Propagation Modes for Striplines (One Dielectric Only)

For coupling regions that have only one propagation velocity, i.e., where all of the traces are striplines (whose fields exist only in one type of dielectric), LineSim Crosstalk constructs the conceptually simplest set of propagation modes: one in which mode 1 means trace 1 has a signal V and all other traces have no signal; mode 2 means trace 2 has a signal V and all other traces have no signal; and so forth.

Thus, when you choose mode 1 from the Propagation Mode combo box and click the Start button, you will see lines emanating from and surrounding trace 1. Mode 2 produces lines around trace 2 — and so forth.

It should be emphasized that the construction of a propagation-mode set is arbitrary and has nothing to do with the validity of the electromagnetic solutions generated by the field solver or the waveforms generated by LineSim Crosstalk's simulator. For stripline configurations, any basis set is equally valid, and LineSim Crosstalk really only has to choose one so that it can display field lines in some reasonable manner. Thus, it makes sense to choose a set of modes that is conceptually simple.

Propagation Modes for Microstrips and Buried Microstrips (Multiple Dielectrics)

However, for coupling regions that have multiple propagation velocities, i.e., where the traces are microstrips or buried microstrips (whose fields penetrate multiple dielectrics), the concept of propagation modes takes on added physical significance. In this case, it is possible to construct a set of modes such that each mode represents the amount of signal on each trace that travels at *one* of the propagation speeds. The number of speeds (and therefore modes) equals the number of traces. (For more details, see Chapter 9, section “Multi-Speed Propagation.”)

For example, for the coupling region shown in Figure 3-1, there are three modes, one propagating energy with a speed of 51.6% of the speed of light; another propagating at 50.8% of light speed; and a third propagating at 49.1% of light speed. In general for the multi-speed case, each mode involves some amount of signal on each trace. Therefore, when you plot one of the modes (unlike with the single-velocity case; see “Propagation Modes for Striplines” above), you see lines emanating from and surrounding *all* of the traces. Figure 3-1 shows the plot for this coupling region's propagation mode 1.

Note: *Following the theme of the previous sections' discussion, you could define other mode sets for Figure 3-1's coupling region. Suppose, for example, that you defined mode 1 as [+V,0,0], mode 2 as [0,+V,0] and mode 3 [0,0,+V]. This is conceptually simple at first glance, but now each mode involves a mixture of three different propagating speeds. So the more physically significant mode set for these traces is the one in which each mode propagates signals at one "pure" speed.*

For two-trace microstrip and buried-microstrip configurations in which the traces are symmetrically arranged (i.e., each trace is on the same layer, has the same width and thickness, etc.), it turns out that the mode set that describes the two propagation speed and the differential/common mode set coincide, i.e., they're the same. Thus, for symmetric trace arrangements, driving purely differential signals means that only one mode is excited, and only one propagation speed results. For more details, see Chapter 9, section "Differential and Common Modes."

Zooming in the Graphical Viewer

By default, the graphical viewer shows the entire cross section for the current stackup. (For details on how to edit the stackup, see the LineSim User's Guide.) To see the field-line plots in more detail, you may want to zoom the viewer in closer to the coupling region's traces.

To zoom the graphical viewer in closer:

1. In the Field Plotting area, click on the Auto Zoom check box.

Copying a Picture of the Field Lines to the Windows Clipboard

You can copy a picture of the field-line plot to the Windows Clipboard; the picture can then be pasted into another Windows application (like Word or PowerPoint).

To copy the field-line plot to the Windows Clipboard:

1. In the Field Plotting area, click the Copy to Clipboard button. The plot is now on the Windows Clipboard.

The field-line plot is captured in Enhanced Metafile format, a “smart” Windows format which can be flexibly resized (i.e., “stretched”) in other applications.

Identifying Transmission Lines in the Graphical Viewer

When you point with the mouse in the graphical viewer at a trace, the name of the trace’s transmission line is displayed. (For details on how transmission lines are named, see Chapter 2, section “How Transmission Lines are Labeled in the Tree List.”) This is a useful way of correlating traces in the viewer to transmission lines in the schematic. If the traces are too small to easily touch with the mouse, try zooming in closer with the Auto Zoom check box (see “Zooming in the Graphical Viewer” above).

Generating a Report of the Field Solver’s Numerical Results

Much of the field solver’s output data comes in the form of matrices or lists of numerical parameters (e.g., impedance, propagation speed, etc.). All of this data can be viewed in a report file, which in turn can be printed or saved for future reference.

To create a text report of the field solver’s numerical results:

1. First, verify that you are in the Edit Transmission Line dialog box (if not, point in the schematic editor to a transmission line that is in the coupling region whose numerical results you want to see, and right-click).

2. Then click the Field Solver tab. A large graphical view of the coupling region appears.

3. In the Numerical Results area, click on the View button. If the View button is grayed out (because the coupling region has changed and the field solver has not yet been run), click the Start button. This forces the solver to run. Then click the View button.

A report file is created, and the HyperLynx File Editor opens on it. You can scroll up and down in the editor to see the report's data, and print it if desired. By default, the report is written into a file named <Coupling_Region_Name>.TXT, where <Coupling_Region_Name> is the name of the coupling region for which the data is being reported. The file is located in the same directory as your .TLN schematic file.

To print the numerical results:

1. In the HyperLynx File Editor, from the File menu, choose Print.

To save the numerical results into a file with a name different than the default:

1. In the HyperLynx File Editor, from the File menu, choose Save As.

Note that if you typically leave your coupling regions with default names (e.g., “Coupling0001,” “Coupling0002,” etc.), then there is a strong likelihood the your numerical-results files will get overwritten frequently. Therefore, if you want to preserve a certain set of field-solver data, it's wise to save the results file under a unique name. (For details on naming coupling regions, see Chapter 2, section “Naming a Coupling Region.”)

Contents of the Results Report

The table in the section “What Detailed Results are Available?” above gives an overview of the data contained in the field-solver numerical results file. The following sections provide more details.

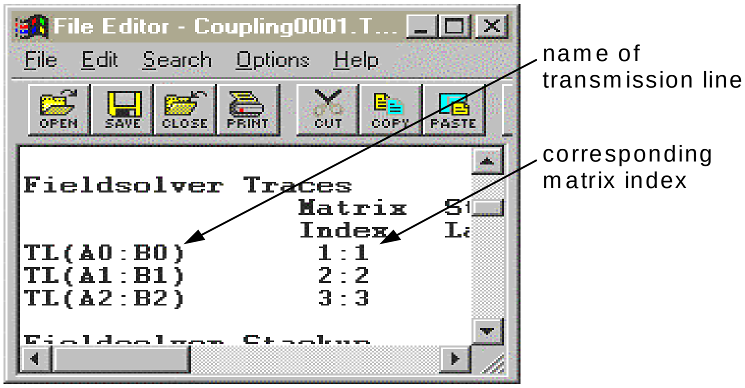
Physical Input Data

This section of the report file shows for what geometric and material data the field-solver results were calculated. The data serves as a record of the input problem, for future reference. Also, for certain coupling regions whose characteristics are important in a key design decision, it may also be worth looking at the input data to verify that the problem on which the field solver ran was exactly as expected. The input data includes information on each trace in the coupling region as well as the region's PCB stackup.

Correlating Transmission Lines and Matrix Indices

In the input data, the Field Solver Traces section lists by name each transmission line in the coupling region, and shows the corresponding trace index by which the line is referred to in the electrical matrix data elsewhere in the file. This data allows you to correlate transmission lines and trace indices. See Figure 3-2.

Figure 3-2: Example of table correlating transmission lines and trace indices



Field-Solver Output Data

This section of the report file shows the electrical data calculated by the field solver for the coupling region. As described earlier, much of this data is in matrix form, because the transmission lines in the region are coupled together. For background information on how electrical parameters are expressed for systems of coupled lines, see Chapter 9.

Optimal Terminator-Resistor Array

Many times, users of LineSim Crosstalk's crosstalk-analysis features are interested in how to terminate traces that are coupled to other traces. This matrix gives the theoretically optimal resistor termination array for the set of coupled lines in the coupling region.

A key fact about coupled lines is that they cannot be perfectly terminated *individually*. Instead, a matrix of resistors that prescribes

both line-to-ground *and line-to-line* resistances is required. (Again, for background information, see Chapter 9.) This termination array has the remarkable property that it not only “kills” single-line reflections at the line ends, but also eliminates arriving crosstalk signals.

On the other hand, there are many situations in digital electronics where line-to-line resistors (in addition to adding undesirably to passive-component count) are simply not permissible for DC-bias reasons. For example, whereas two coupled data lines may require a 160-ohm resistor between them to eliminate line-to-line crosstalk, it is unlikely that the driver ICs on the lines would be “happy” with the resistor when one line was pulled high and the other low.

Still, in some critical situations, especially when the line-to-line coupling is relatively weak and therefore the line-to-line terminating resistances are fairly high, a matrix terminator may be workable.

Note: *There are some IC technologies which are specifically designed to work with line-to-line termination: differential drivers. For these devices, line-to-line termination serves not only to prevent line reflections and eliminate crosstalk, but is often also required to bias the ICs for correct operation.*

To implement the termination described in the Optimal Terminator-Resistor Array matrix:

1. Place the resistors in the diagonal matrix positions between the corresponding trace to ground. (E.g., resistor 2-2 should be placed from trace 2 to ground, at the trace end.)

2. Place the resistors in the off-diagonal matrix positions line-to-line between the corresponding traces. (E.g., resistor 2-1 should be placed between traces 1 and 2, at the trace ends.)

Note that there are twice as many off-diagonal values as there are line-to-line resistors, since, for example, off-diagonal resistance 2-1 refers to the same resistor as resistance 1-2.

Note*: To correlate a specific transmission line in a coupling region to a matrix index, see the field-solver report’s Physical Input Data section, Field Solver Traces table.*

Characteristic-Impedance Matrix

This matrix gives the characteristic impedance (in ohms) of the system of coupled transmission lines in the coupling region. As noted previously (and described in more detail in Chapter 9), coupled lines do not have a single-value impedance, like uncoupled lines. Rather, together, a set of coupled lines share an impedance matrix.

The values in the diagonal matrix positions can be thought of as giving the impedances to ground of the corresponding transmission lines, accounting for the presence of the other nearby, coupled traces. When an IC drives into one of the lines, however, it “sees” not only the diagonal impedance for that line, but also some of the off-diagonal terms in the matrix.

For lines that are only weakly coupled, the diagonal impedance terms are dominant, and the diagonal values are close to what they would be if the lines were completely isolated from each other. As the coupling becomes stronger, the diagonal terms deviate more from their standalone values, and the off-diagonal terms increase. Note that small off-diagonal impedances mean weak coupling; large impedances mean strong coupling.

Barring special cases like two-line pairs in which the two signals are known to be either purely differential or purely common-mode, the diagonal impedances in the matrix are generally the best single-resistor terminators to use. Note, however, that coupled transmission lines cannot be perfectly terminated unless a full matrix termination (including both line-to-ground and line-to-line resistors) is employed. See “Optimal Terminator-Resistor Array” above for details.

***Note**: To correlate a specific transmission line in a coupling region to a matrix index, see the field-solver report’s Physical Input Data section, Field Solver Traces table.*

Capacitance Matrix

This matrix gives the self and mutual capacitances (in pF/m) of the coupled transmission lines in the coupling region. More specifically, the diagonal values in the matrix give the capacitances to ground of the corresponding transmission lines, while the off-diagonal values give the capacitances between the corresponding pair of lines.

Many users are surprised to see that the off-diagonal capacitance-matrix values are negative. The negative sign simply reflects the fact that if a positive charge is placed on a given trace, negative charge will accumulate on all others. **For purposes of judging how much capacitance exists between traces, you can ignore the negative signs. The off-diagonal values do represent real, physical capacitance.**

However, in the mathematical formalism of coupled transmission lines, the negative signs are important. For example, if you transfer the capacitance matrix for a coupling region to another EDA tool (e.g., SPICE), the off-diagonal values **must** be negative.

Note that the values in the capacitance matrix have units of pF/m, rather than simply pF. This means that if you are trying to calculate, for example, the total capacitance-to-ground of a transmission line in the matrix, you must multiply the corresponding diagonal value in the matrix by the length (in meters) of the line.

***Note**: To correlate a specific transmission line in a coupling region to a matrix index, see the field-solver report's Physical Input Data section, Field Solver Traces table.*

Inductance Matrix

This matrix gives the self and mutual inductances (in nH/m) of the coupled transmission lines in the coupling region. More specifically, the diagonal values in the matrix give the self inductances of the corresponding transmission lines, while the off-diagonal values give the mutual inductances of the corresponding pair of lines.

Note that the values in the inductance matrix have units of nH/m, rather than simply nH. This means that if you are trying to calculate, for example, the total self inductance of a transmission line in the matrix, you must multiply the corresponding diagonal value in the matrix by the length (in meters) of the line.

***Note**: To correlate a specific transmission line in a coupling region to a matrix index, see the field-solver report's Physical Input Data section, Field Solver Traces table.*

Propagation-Speeds List

This list gives the speed(s) (in m/s) at which signals propagate along the transmission lines in the coupling region. As noted previously (e.g., see "Choosing a Propagation Mode to Plot" above) and described in more detail in Chapter 9, coupling regions in which there is only dielectric (e.g., for stripline traces) have only one propagation speed; the signals on all traces in the region propagate with this single velocity.

However, coupling regions in which there are boundaries between dissimilar dielectrics (e.g., for microstrip or buried-microstrip traces) have multiple, discrete propagation speeds. Generally, each transmission line in the coupling region propagates some energy at each of the velocities prescribed by the region. There are as many speeds as there are transmission lines in the coupling region.

For most practical cross-section geometries, the multiple speeds are all close to each other. However, it is possible to construct highly asymmetric cross sections in which the speeds are quite different. (An example of a "highly asymmetric" geometry would be a microstrip of one width coupled to a buried microstrip of a different width, with the buried trace below and considerably off to the side of the outer-layer trace.) This is an undesirable condition, however, because multiple, widely varying propagation speeds cause signal distortion, as one portion of the signal races ahead of the other(s).

For convenience, the propagation-speeds list displays velocities not only in m/s, but also as a fraction of the speed of light. For example, a value of "0.4822c" means 48.22% of the speed of light.

Note: *A misconception about propagation velocity on a transmission line is that electrons in the conductor are traveling along the line at the propagation velocity. This is absolutely not true! Electrons in a conductor spend almost all of their time randomly colliding with atoms in the conductor lattice; the mean time between collisions is on the order of 10 femtoseconds (1/100th of a ps).*
As a result, conduction electrons have only a relatively tiny average forward velocity in the presence of a driving voltage. A typical electron "drift velocity" in a conductor is on the order of 1 foot/hour. Instead, what moves at the transmission line's propagation velocity is the electromagnetic wave that constitutes the actual signal on the line. Indeed, this wave is what you measure

in the lab with an oscilloscope: a voltage waveform, which is really a measure of the electric field associated with the traveling electromagnetic wave.

Percentage of Energy Matrix (Multiple-Speed Coupling Regions Only)

If the coupling region supports multiple propagation speeds, this matrix gives, for each transmission line in the region, the percentage of signal energy that travels at each speed. In the matrix, each column represents a line (i.e., a trace); reading down the column shows how much of the signal energy in that line travels in each of the propagation modes listed in the propagation-speeds list. The percentages in each column add to approximately 100%, to fully account for the energy in each transmission line.

The values in this matrix are usually only of limited interest, unless the matrix shows a very uneven breakdown in energy sharing between propagation modes. For example, for certain highly asymmetric (and unusual) geometries, it is possible to have certain transmission lines carrying most of their energy in one mode, while others carry a more even mixture of modes. If the velocities between modes differ significantly, this uneven distribution could lead to noticeable skew between signals on the lines.

Impedance and Termination Summary (Two-Line Coupling Regions Only)

For the special case of a two-line coupling region, the field-solver numerical results report gives additional information about specific termination options. The following table summarizes the additional data.

Termination Type	**Description**
differential	This is the proper line-to-line resistor to use if the two transmission lines are being driven differentially, i.e., with equal-but-opposite signals. Will not terminate common-mode signals at all.
common-mode	This is the proper line-to-ground resistor to use for each line if the two transmission lines are being

Termination Type	**Description**
	driven identically, i.e., with equal signals of the same polarity. Not very useful for signals that sometimes switch together and sometimes oppositely, unless crosstalk is primarily of concern when they switch together.
line-to-ground	This is the best line-to-ground resistor to use for each line if the signals on the transmission lines are completely unrelated. Will not perfectly terminate the line, but is a good single-component “compromise” value.
optimal termination	Describes the theoretically optimal resistor-array termination; consists of a line-to-line resistor plus two line-to-ground resistors. Same as the values given in the Optimal Terminator-Resistor Array matrix. Successfully terminates differential, common-mode, or mixed signals, but may violate DC-bias conditions on the lines.

Chapter 4: Advanced Mode for Coupling Transmission Lines in a LineSim Schematic *(LineSim Crosstalk)*

Summary

Important! *This chapter is specific to the LineSim Crosstalk product; it does not apply to BoardSim Crosstalk. For detailed information about BoardSim Crosstalk, see Chapters 6-9.*

This chapter describes:

- what is meant by the "advanced coupling mode" in LineSim Crosstalk's schematic editor
- what a "coupling dot" is
- how to set LineSim Crosstalk to advanced coupling mode
- how to change the positions of coupling dots in a schematic
- examples of when you might need to use advanced mode

“Advanced” versus “Normal” Coupling Modes

As described in Chapter 2, LineSim Crosstalk’s user interface supports two modes for defining coupling regions:

- a “normal” mode that imposes a few restrictions on how you can place coupled transmission lines in the schematic, but makes defining coupling regions simple and intuitive
- an “advanced” mode that allows more flexibility than the “normal” method, but also is more prone to entry error and requires you to understand more coupling concepts

HyperLynx recommends that you use the normal coupling mode whenever possible; LineSim Crosstalk defaults to normal mode. However, there are situations in which advanced mode is required or useful, and once you understand the concept of “coupling dots,” advanced mode is perfectly safe to use. This chapter describes advanced mode.

For more details on normal mode, or to learn about the basics of adding coupling to a LineSim Crosstalk schematic, see Chapter 2.

What is “Advanced Coupling Mode”?

In the normal coupling mode, LineSim Crosstalk imposes restrictions on which transmission lines can be grouped together into a coupling region. See Chapter 2, section “Requirements of the Normal Coupling Mode” for full details, but generally, the requirement is that all lines in a coupling region must be horizontal and in the same row, or vertical and in the same column.

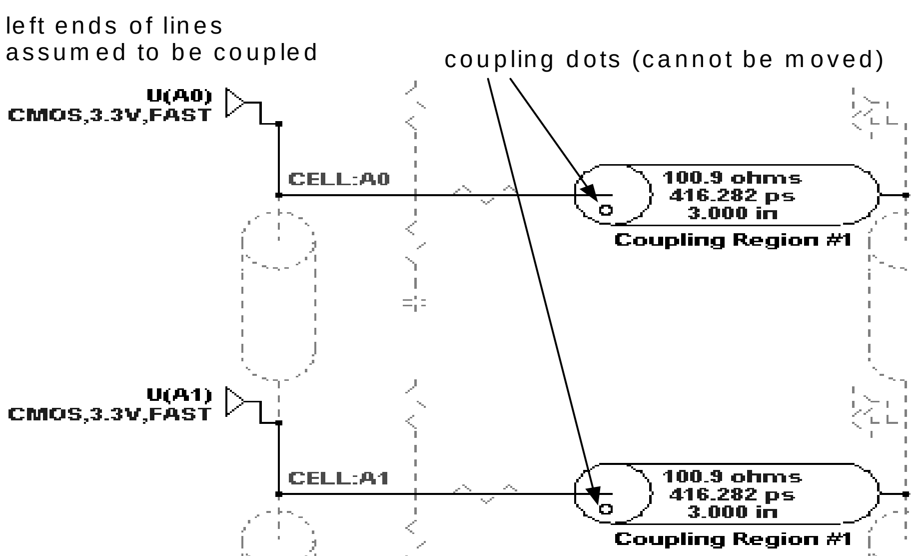
The reason LineSim Crosstalk imposes this restriction is that if you adhere to it, it is obvious which ends of the transmission lines in each coupling region should be coupled to which other ends. For example, if you have three horizontal lines in a column, it is logical that the left ends be coupled to the other left ends, and the right ends to the other right ends.

Coupling Dots

LineSim Crosstalk actually shows this assumed coupling by displaying a "coupling dot" at the left end of each horizontal transmission line. The dotted line ends are coupled to the other dotted ends, and the undotted ends to the other undotted ends. See Figure 4-1.

Note: *The concept of coupling dots is similar to the "dot" convention used in transformer-winding symbols. There, dots are used to indicate which ends of the windings produce positive electromagnetically induced voltages at which other ends, when current is driven into a dotted end. Here (in LineSim Crosstalk), dots are used to show which ends of transmission lines are electromagnetically coupled to which other ends.*

Figure 4-1: Coupling dots in a normal-mode LineSim Crosstalk schematic



The situation is similar with vertical transmission lines. For them, the coupling dots default to the top side, so that the top sides of vertical lines couple to other top sides, and bottom sides to other bottom sides.

But what if you are drawing a complex schematic and you need to couple the right end of a horizontal transmission line to the left end of another horizontal line, or the top side of a vertical line to the bottom side of another? Or what if you need to couple a horizontal line to vertical line, or two horizontal lines in different columns? For these situations, you must switch LineSim Crosstalk into advanced coupling mode, and manage coupling dots yourself.

Advanced Coupling Mode

Advanced coupling mode is really no different than normal mode, *except* that it allows you to couple any transmission line in a schematic to any other line, and allows you to change the position of any line's coupling dot. With this added flexibility, though, comes the responsibility of explicitly managing how lines are coupled.

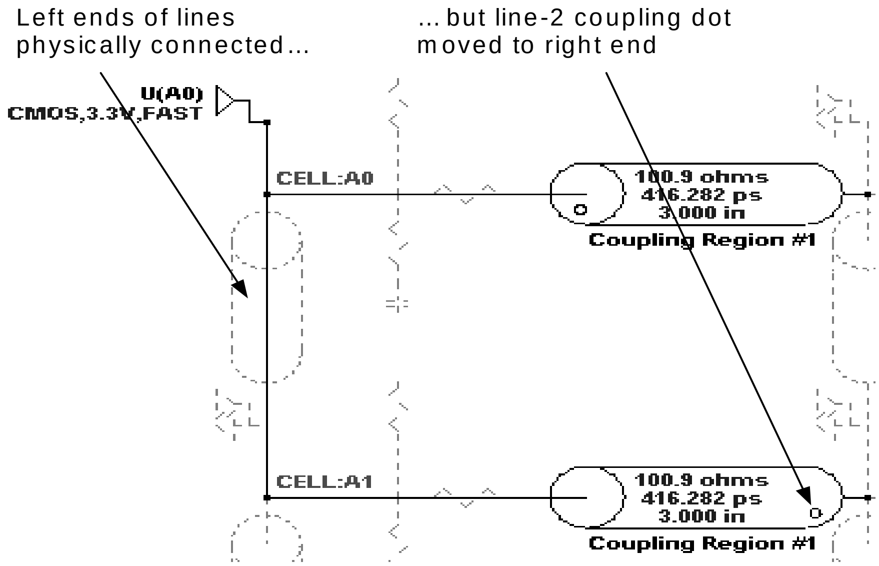
Normal mode imposes drawing restrictions, but by assuming that left line ends are always coupled to other left ends and top ends to top ends, and by enforcing that a coupling region's lines are always "stacked" vertically or horizontally, frees you from having to worry whether your coupling dots are correctly positioned. HyperLynx recommends using normal mode whenever possible, because it reduces the chance of entry error.

Example of a "Bad" Coupling

Figure 4-2 shows an example of a schematic created in advanced coupling mode which probably does not represent what the user actually wanted to simulate. Notice that the left ends of the two transmission lines are physically tied together, and therefore presumably are also electromagnetically coupled. But the coupling dot for the second line has been moved to the opposite end. LineSim Crosstalk will therefore *electromagnetically* couple the left end of trace 1 and the right end of trace 2, but *electrically* short the two left ends. Physically, this makes little sense.

Overall, the message is: advanced coupling mode is powerful; use it when needed, but use it carefully.

Figure 4-2: Example of an advanced-mode schematic that probably has an error



Setting LineSim Crosstalk to Advanced Coupling Mode

By default, LineSim Crosstalk runs in normal coupling mode. However, you can switch to advanced coupling mode any time you want, even in the middle of a schematic-drawing session.

To switch LineSim Crosstalk into advanced coupling mode:

1. From the Options menu, choose Preferences. The Options dialog box opens.
2. Click on the LineSim tab.
3. In the Analysis Options area, click on the Enable Advanced Coupling Mode check box. Click OK.

The schematic editor will now allow you to place or move any transmission line in the schematic into any coupling region. You can mix vertical and horizontal lines in the same coupling region, or add horizontal lines that are in different columns or vertical lines that are in different rows. You can also move transmission-line coupling dots lines from their default positions.

Coupling Mode is Saved Between Sessions

LineSim Crosstalk "remembers" from session-to-session the coupling mode in which it was last running. If you switch to advanced mode, for example; exit LineSim Crosstalk; and later return to it, LineSim will set itself back into advanced mode as it starts up.

Loading an Advanced-Mode Schematic

The mode in which a schematic was created (i.e., normal or advanced) is stored with the schematic's .TLN file. If LineSim Crosstalk is running in normal mode and you load a previously saved schematic which was created in advanced mode, LineSim will switch automatically to advanced mode.

The converse is not true: once you have switched LineSim Crosstalk into advanced mode, it remains there even if you load a schematic which was created in normal mode.

Moving Coupling Dots

Once LineSim Crosstalk is running in advanced coupling mode, you can move the coupling dots on coupled transmission lines from their default positions to the opposite line ends. (For information on what a coupling dot is, see "Coupling Dots" above; for instructions on setting LineSim Crosstalk to advanced coupling mode, see "Setting LineSim Crosstalk to Advanced Coupling Mode.")

To move a transmission line's coupling dot from one end to the other:

1. In the schematic editor, point to the line whose coupling dot you want to move. Right-click with the mouse. The Edit Transmission Line dialog box opens.

2. In the Transmission Line Type area, in the Coupled column, in the Coupling Direction area, click the radio button that is not currently selected. Click OK.

Note: *If the Coupling Direction area is not visible, then LineSim Crosstalk is running in normal coupling mode rather than advanced. For details on switching to advanced mode, see "Setting LineSim Crosstalk to Advanced Coupling Mode" above.*

Back in the schematic editor, the coupling dot on the transmission line which you just edited has changed position, from one end to the other.

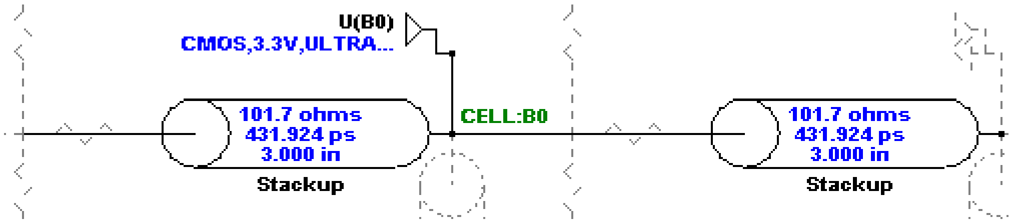
Examples of Using the Advanced Coupling Mode

This section gives some examples of when it might be necessary to use advanced coupling mode. But notice that often, with a different drawing strategy, normal coupling mode could be used just as well.

Example #1: Driving Two Lines in Parallel from a Midpoint

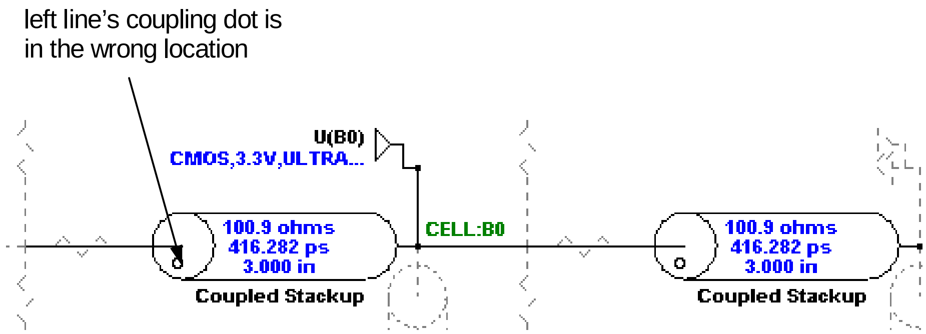
Figure 4-3 shows one way of drawing a layout scenario in which an IC drives two lines in parallel. This is not the only way to draw such a circuit in LineSim Crosstalk— in fact, as Figures 4-6 and 4-7 demonstrate, it is probably not even the best way.

Figure 4-3: One way of drawing an IC driving two lines in parallel



If the two transmission lines are made coupled, then the schematic looks as in Figure 4-4.

Figure 4-4: Figure 4-3's schematic, with the transmission lines coupled

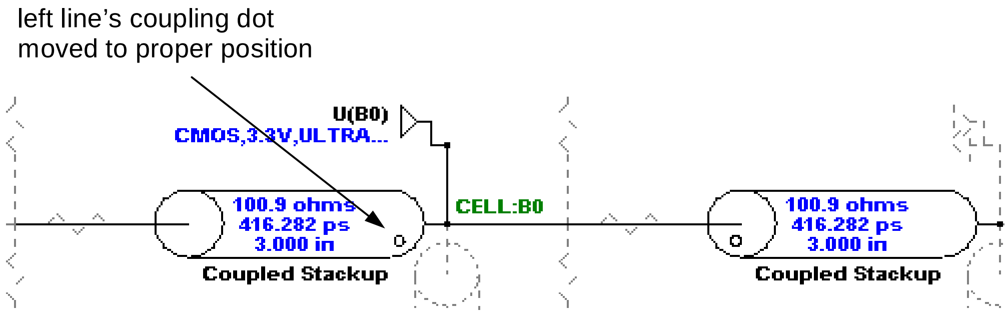


Notice the position of the coupling dots in Figure 4-4: both are on the left ends of the transmission lines. But the electrical connectivity indicates that the coupling should actually be between the right end of the left line and the left end of the right line. If the dots remain as in Figure 4-4 and a simulation is run, the electrical connectivity and electromagnetic coupling will be in conflict.

This is a situation, then, where a coupling dot must be moved, and the advanced coupling mode is required. After switching LineSim Crosstalk to advanced mode and moving the left line's dot, the schematic looks as in Figure

4-5. This corresponds to the actual layout physically, and will give the proper simulation result.

Figure 4-5: Figure 4-4's schematic, with left line's coupling dot moved to the proper position



It is worth noting, however, that the same layout could be drawn in several different ways the do *not* require advanced coupling mode in order to properly represent the problem. Figure 4-6 and 4-7 show two alternate possibilities.

Figure 4-6: Alternate representation of Figure 4-5 that does not require the use of advanced coupling mode

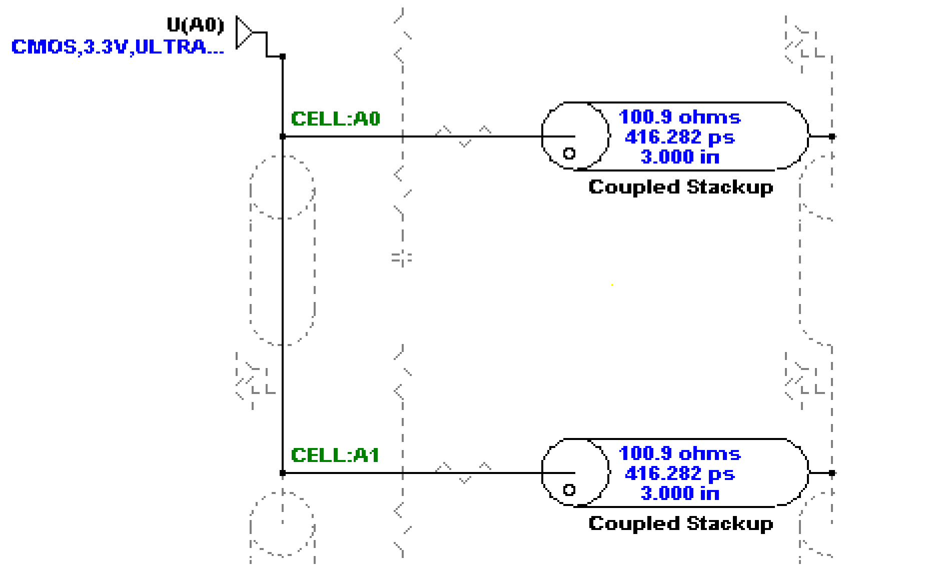
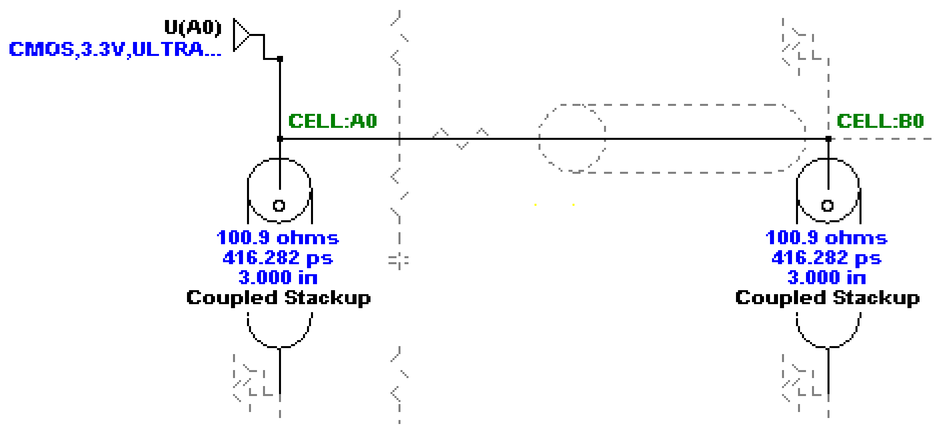


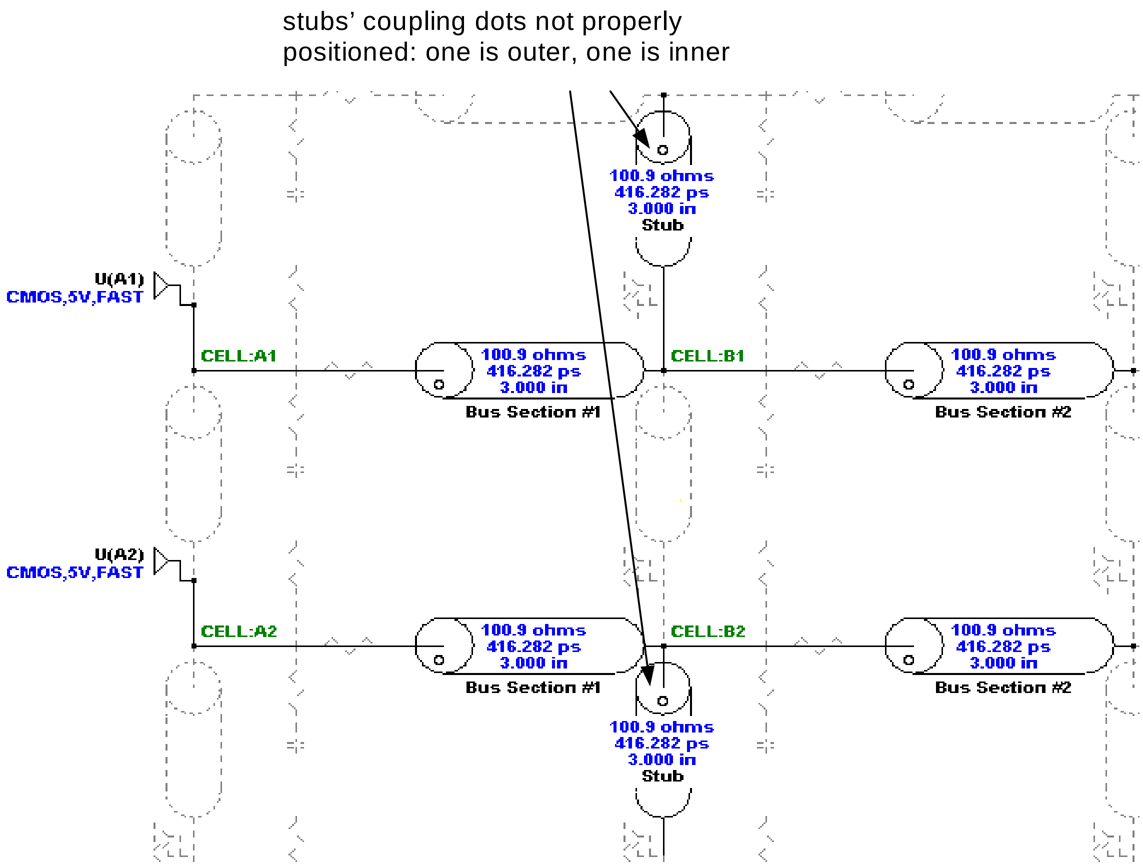
Figure 4-7: Another alternative to Figure 4-5 that does not require advanced coupling mode



Example #2: Two Parallel Bus Lines with Coupled Stubs

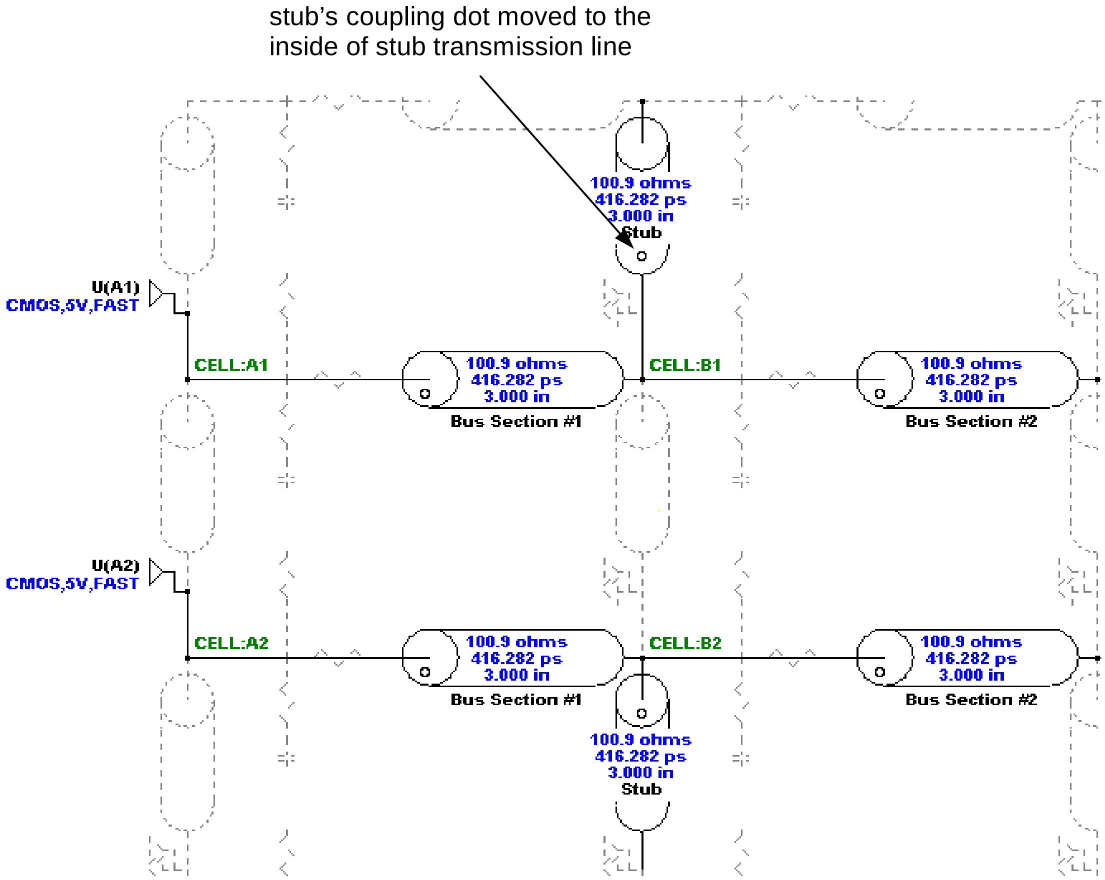
Figure 4-8 shows a portion of a schematic which represents two adjacent bus lines, in which both the traces between bus taps and the stub traces for each tap are coupled together. The schematic was drawn in normal coupling mode, so the coupling-dot positions are the defaults.

Figure 4-8: Parallel bus lines, where all transmission lines — including stubs — are coupled together



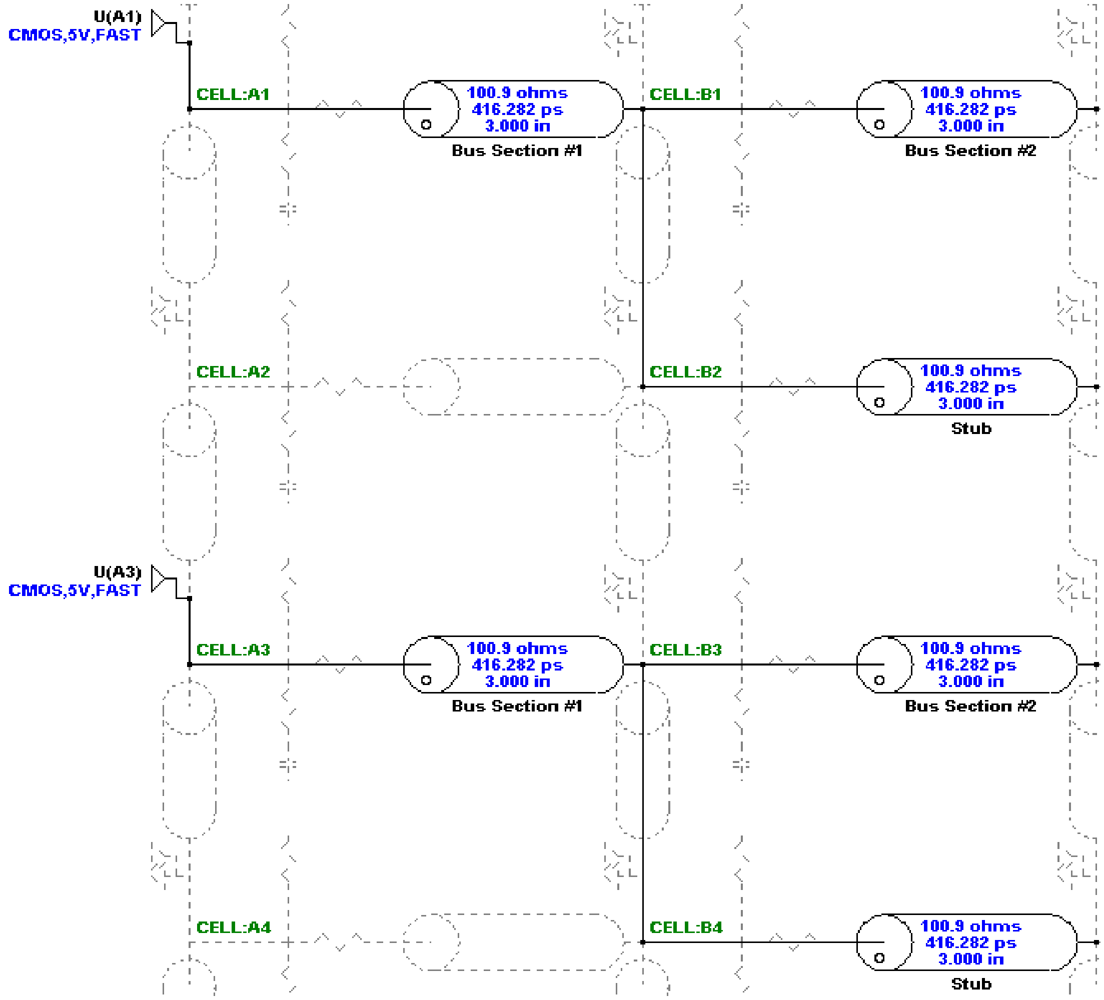
Notice that the positioning of the coupling dots on the two stub transmission lines is incorrect: as drawn, the outer end of the upper stub couples to the inner end of the lower stub. Figure 4-9 corrects this; LineSim Crosstalk was placed in advanced coupling mode and one of the stub's coupling dots was moved.

Figure 4-9: Figure 4-8's schematic, with the stub coupling dots correctly positioned



Again, though (as in the previous example), the same layout could be drawn in a manner that obviates the need for advanced coupling mode. See Figure 4-10.

Figure 4-10: Alternate representation of Figure 4-9 that does not require the use of advanced coupling mode



Chapter 5: Application Examples for LineSim Crosstalk

Summary

Important! *This chapter is specific to the LineSim Crosstalk product; it does not apply to BoardSim Crosstalk. For detailed information about BoardSim Crosstalk, see Chapters 6-9.*

This chapter describes how to create crosstalk simulations for several common applications, specifically:

- a simple two-trace example (basic crosstalk)
- a differential pair
- determining minimum allowable trace separation on a bus
- assessing the effect of guard traces

If you need reference information about the detailed mechanics of creating and editing the “coupling regions” that model specific crosstalk sections, see Chapter 2. If you prefer to learn by seeing examples, this chapter may be more useful.

A Simple Two-Trace Example (Basic Crosstalk)

This example shows the most basic of all possible crosstalk simulations: two side-by-side PCB traces, one driven as an “aggressor” net and the other as a "victim." Once the basic problem is set up, several variations are explored, to see what methods could best reduce the crosstalk in this case.

Note: *The schematic for this example is installed in the HYPFILES directory as "XT Manual Basic Crosstalk Example.tln." However, you'll learn more if you create the schematic yourself as you follow the steps in the example.*

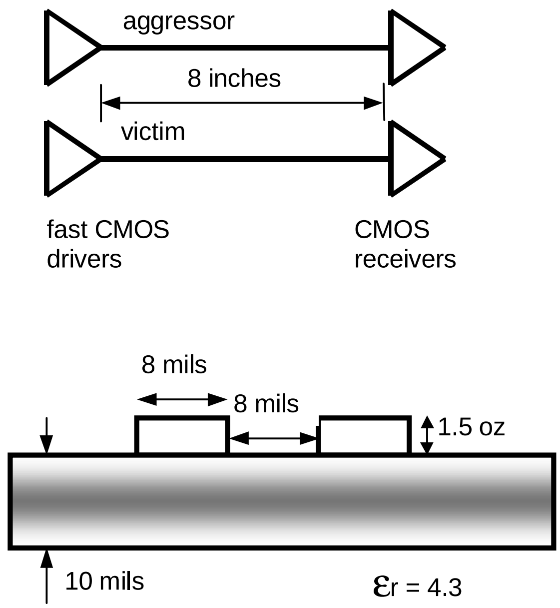
Application Description

Suppose you want to simulate the amount of crosstalk that will occur between two side-by-side PCB traces with the following characteristics:

- microstrip (i.e., outer-layer) traces
- 8 inches long
- 8 mils wide
- 8 mils between traces (edge to edge)
- 0.5-ounce base copper with 1-ounce plating
- 10 mils above a ground plane
- PCB dielectric constant of 4.3
- both traces driven at their left end with a fast 3.3-V CMOS driver
- consider trace 1 to be the "aggressor" and trace 2 the "victim"

Figure 5-1 summarizes the problem graphically.

Figure 5-1: Geometry for basic two-trace crosstalk example



Creating the Schematic In LineSim Crosstalk

The first step in this example is to draw a LineSim Crosstalk schematic that includes the two transmission lines and the ICs attached to them. This involves only features that even the base LineSim product has (basic schematic-drawing and IC-modeling capabilities). Then, once the schematic is created, you use additional features from LineSim Crosstalk to couple the transmission lines together.

Note: *This manual assumes that you already understand the basics of creating and editing schematics in LineSim. The emphasis here is on the additional features that LineSim Crosstalk adds to base LineSim. If you are unfamiliar with the base LineSim product, refer to the LineSim User's Guide or the online Help system.*

Create a schematic representing the two-trace crosstalk example (see Figure 5-2):

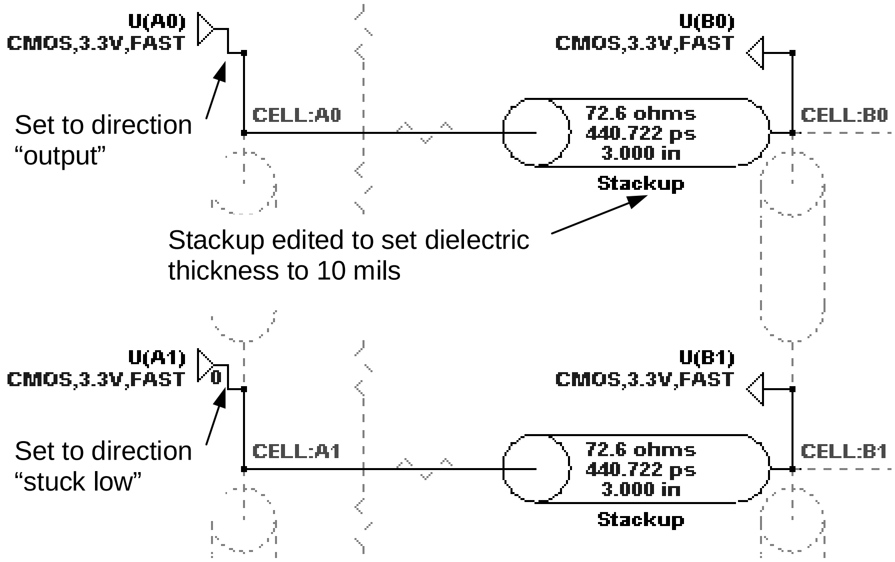
1. Create a simple schematic that contains two horizontal transmission lines, each with ICs at its left and right end. The two lines must be in the same schematic column (e.g., one above the other; see Figure 5-2).

> ***Note:*** *You can couple together transmission lines that are not in the same schematic column or row, but this is an advanced feature that most users don't need. For details on enabling the "advanced coupling mode" that allows this, see Chapter 4.*

2. Model all of the ICs with the "CMOS,3.3V,FAST" model from the library EASY.MOD. (Hint: use the Paste All button in the Assign IC Models dialog box to quickly apply this model to all ICs in the schematic.)
3. Change the driving direction of the upper line's left-end IC (pin U(A0)) to be "Output." Change the direction of the lower line's left-end IC (pin U(A1)) to be "Stuck Low." Leave the right-end ICs set as "Input."
4. Use the stackup editor to change the dielectric between the TOP layer and VCC plane to be 10 mils thick.

The schematic you've created so far should look like Figure 5-2.

Figure 5-2: Basic schematic for two-trace crosstalk example



By making the driver IC at the left end of the upper trace a switching "output," you have effectively made this trace the "aggressor" in the simulation. By making the driver IC at the left end of the lower trace "stuck low," you have made this trace the "victim."

It is important to include non-switching (i.e., "stuck") driver ICs on victim nets, because on a real PCB, victim nets do have drivers. If you omit the "stuck" driver(s), the crosstalk waveforms will look much different than if you include them. This occurs because driver ICs are typically low impedance and will reflect, rather than absorb, crosstalk signals.

In this example, we're bothering to simulate only with the victim-net driver stuck low. However, in general, you should check both the stuck-low and stuck-high cases, to see which generates more crosstalk.

Adding Coupling Information to the Schematic

Continue by placing the transmission lines in a coupling region:

1. Point to the upper transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Make the line coupled by clicking on the Stackup radio button **under the Coupled heading.** The Add to Coupling Region tab is automatically selected.
2. In the Coupling Regions list, double-click on the New-Coupling item. This tells LineSim Crosstalk to automatically create a new coupling and add the transmission line to it. The dialog box closes.
3. Repeat step 1, but for the lower transmission line.
4. In the Coupling Regions list, double-click on the Coupling0001 item. This adds the lower transmission line to the coupling region that was automatically created for the upper transmission line. The dialog box closes.

Back in the schematic editor, there are several “clues” that the transmission lines are coupled together:

- If you point to one of the lines with the mouse, it highlights not only in red (like normally in LineSim) but also in yellow, which indicates “coupled line.”
- If you point to one of the lines, a dashed yellow “ratsnest” line appears, joining the line you pointed at to the other line to which it is coupled.
- A “coupling dot” appears on the left end of each transmission line. The dots indicate that the lines are coupled, and specifically that they are coupled left-end-to-left-end, and right-end-to-right-end. (The dots are meant to be reminiscent of the dots used in transformer symbols to show which ends of various windings are coupled together.)

Note: *It is possible to move a coupling dot from one end of a transmission line to the other in order to accommodate certain kinds of complicated coupling schematics. However, this is considered an advanced feature that you normally do not need. For details on enabling coupling-dot editing, see Chapter 4.*

Define the new coupling region's cross section:

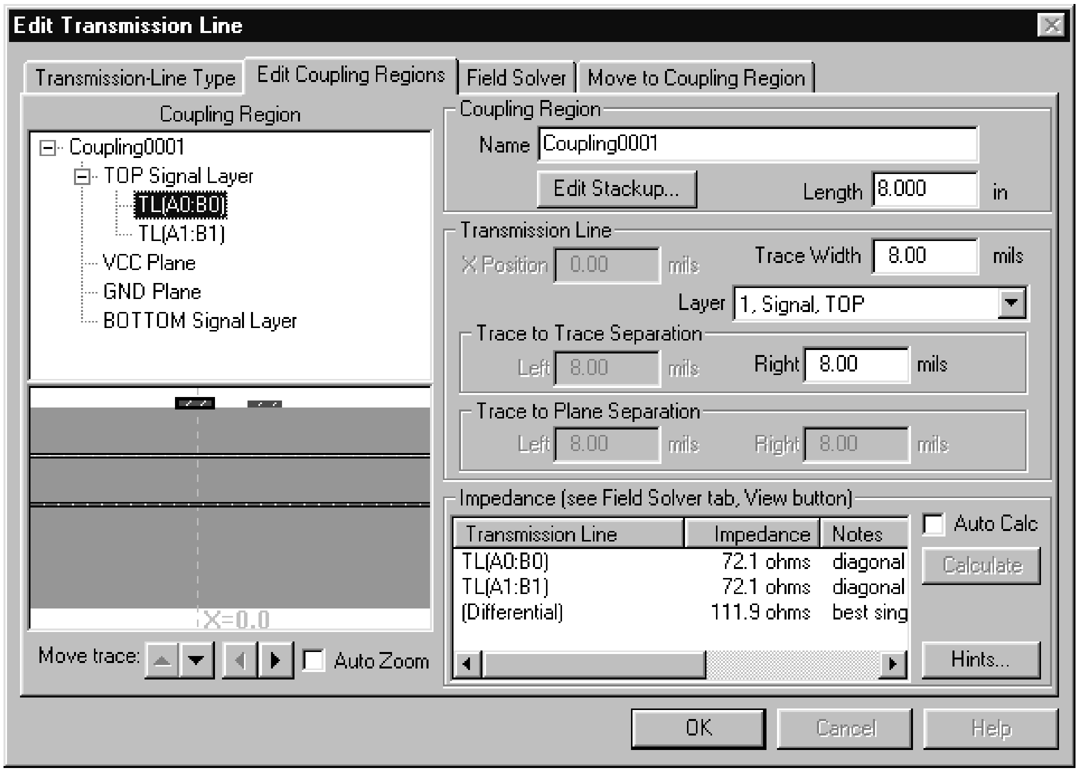
1. Point to either transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Select the Edit Coupling Regions tab.

2. Notice that the Coupling Region tree list shows the new coupling region (auto-named by LineSim "Coupling0001") expanded, so that the current stackup layers and the transmission lines that were added to the region are visible. LineSim Crosstalk defaults to 8-mil-wide traces and 8-mil edge-to-edge trace separations (unless you've changed the Default Trace Separations setting in the Options/Preferences dialog box; see Chapter 2, section "Setting the Default Trace-to-Trace and Trace-to-Plane Separations" for details), so the two lines already have the widths and separation specified in our example.

   However, the length of the coupling region is incorrect, so in the Coupling Region area, in the Length edit box, type "8" to change the length to 8 inches.

Figure 5-3 shows what the Edit Coupling Regions tab should look like at this point.

Figure 5-3: Contents of Edit Coupling Region tab for two-trace crosstalk example



Notice how the transmission lines are identified in the Coupling Region tree list: with a name of the form “TL<xx:yy>”, where xx is the left-end cell coordinate of the line and yy is the right-end cell coordinate; *plus* an optional label comment that you supply. The default label is suppressed from the name; however, it can be edited. Let’s change it for these lines so that it’s easier to remember which is the aggressor trace and which the victim.

Change the transmission lines' label comments to "Aggressor net" and "Victim net":

1. Click OK to close the Edit Transmission Line dialog box.
2. Point to the upper transmission line, and right-click on it. The Edit Transmission Line dialog box opens. In the Transmission-Line Properties area, in the Comment edit box, type "Aggressor net." Click OK.
3. Repeat step 1 for the lower transmission line; type "Victim net" in the Comment edit box. Click OK.

Notice that the transmission lines are now labeled appropriately in the schematic editor.

Simulating to See Crosstalk Waveforms

Now that the schematic has been created and coupling information has been added to it, we're ready to simulate to see how much crosstalk would occur between the two traces.

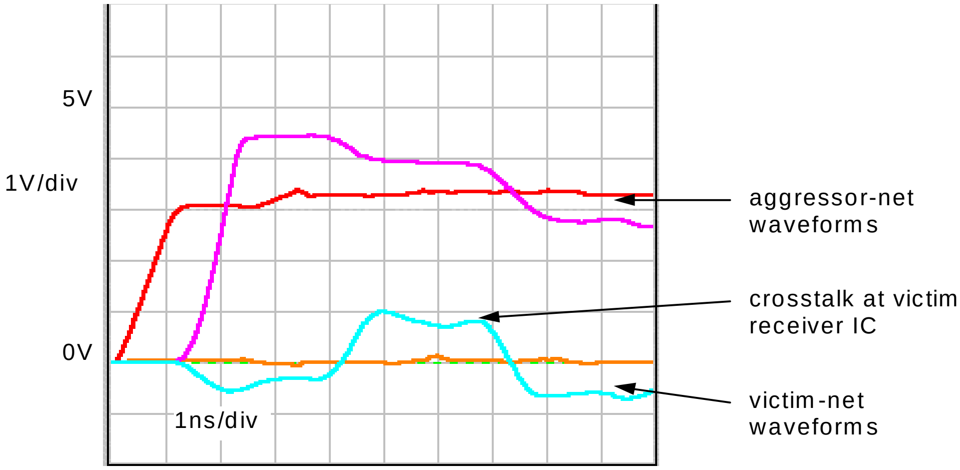
Run the oscilloscope to see the rising-edge crosstalk waveforms:

1. From the Scope/Sim menu, choose Run Scope. The Digital Oscilloscope dialog box opens.
2. In the Driver Waveform area, click on the Rising Edge radio button. In the IC Modeling area, verify that the Typical setting is selected (change it if necessary).
3. Click the Start Simulation button to run the simulator and view the resulting waveform.

A significant amount of crosstalk (about 1.0V worst-case amplitude) appears on the victim net at its receiver IC (blue waveform). The crosstalk is a combination of forward and backward components; the backward portion is reflected initially from the victim-net "stuck low" driver IC, and sent down the victim line along with the forward portion. The backward portion is easily identified because its time duration is twice the delay length of the victim line (approximately 2.3 ns).

Figure 5-4 shows what the rising-edge crosstalk waveform should look like.

Figure 5-4: Rising-edge crosstalk waveform for two-trace crosstalk example



Repeat to see the falling-edge crosstalk:

1. In the Driver Waveform area, click on the Falling Edge radio button.
2. Click the Start Simulation button.

The falling-edge waveforms are similar, but of the opposite polarity.

Determining the Effect of Varying Trace Separation and Stackup Layer

One of LineSim Crosstalk's strengths is the ease with which you can use it to explore design alternatives. For example, in the preceding simulations, we've found a large amount (1 V) of crosstalk on the victim net. What can be done to reduce the crosstalk to an acceptable level? LineSim Crosstalk lets you vary aspects of the problem to quickly find out.

The following sections illustrate some of the alternatives you can explore.

Changing the Driver IC

The rise/fall time of the aggressor-net driver IC has a significant influence on the amount of crosstalk generated. Many texts suggest that crosstalk amplitude is directly proportional to driver edge rate. This is not strictly true, although edge rate is certainly an important factor. LineSim Crosstalk can easily show you exactly how much effect it has in a particular scenario.

To see the effect of reducing aggressor-net driver edge rate by a factor of three:

1. Change the IC model for component U(A0) to “CMOS,3.3V,MEDIUM” from library EASY.MOD. This model has a 3-ns rise time, as compared to the 1-ns rise time of the “CMOS,3.3V,FAST” model used previously.

2. Open the oscilloscope, change the timebase to 2 ns/div and the driver waveform to Rising Edge, and re-simulate. The resulting crosstalk waveform at the victim net’s receiver IC has a peak amplitude of approximately 540 mV. (Increase the oscilloscope’s vertical scale, if desired, to make measurements more easily.)

3. Change the driver model back to “CMOS,3.3V,FAST” for use in the subsequent sections.

Notice that while the driver edge rate decreased by a factor of three, the crosstalk amplitude decreased by only a factor of two.

Reducing the Length of the Coupling

The length over which coupling occurs also has a significant influence on the amount crosstalk generated. Again (like with driver edge rate), many texts suggest that crosstalk amplitude is directly proportional to coupling length. This is not strictly true, particularly for couplings along traces whose delay time is longer than the driver’s rise/fall time.

To see the effect of reducing the length of the coupled traces by a factor of two:

1. In the schematic, point to one of the transmission lines and right-click on it. The Edit Transmission Line dialog box opens.

2. Select the Edit Coupling Regions tab. In the Coupling Region area, in the Length edit box, type “4” to shorten the length of both traces to 4 inches.

3. Open the oscilloscope, set the timebase to 1 ns/div, and re-simulate. The resulting crosstalk waveform at the victim net’s receiver IC has a peak amplitude of approximately 830 mV.

4. Change the coupling-region length back to 8 inches for use in the subsequent sections.

Notice that while the length over which the traces are coupled decreased by a factor of two, the crosstalk amplitude decreased only slightly.

Widening the Trace Separation

The separation between coupled traces has a significant influence on the amount crosstalk generated. Some texts suggest that crosstalk amplitude is inversely proportional to the square of the separation distance — e.g., if the separation distance doubles, the crosstalk amplitude will fall by a factor of four. While this is relationship is not strictly true, trace separation is indeed one of the most important weapons in fighting crosstalk.

To see the effect of increasing trace separation by a factor of two:

1. In the schematic, point to one of the transmission lines and right-click on it. The Edit Transmission Line dialog box opens.

2. Select the Edit Coupling Regions tab. Depending on which transmission line you clicked on, in the Transmission Line area either the Left or Right edit box in the Trace-to-Trace Separation area will be ungrayed. In this box, type “16” to increase the trace separation to 16 mils. Note how the graphical viewer shows the traces moved further apart.

3. Open the oscilloscope and re-simulate. The resulting crosstalk waveform at the victim net’s receiver IC has a peak amplitude of approximately 650 mV.

4. Change the trace-to-trace separation back to 8 mils for use in the subsequent sections.

Notice that while the trace separation increased by a factor of two, the crosstalk amplitude decreased by less than two times.

Changing Stackup Layers

The stackup layer on which coupled traces are located has a significant influence on the details of the coupling, i.e., the electromagnetic parameters that govern how crosstalk occurs. Varying the parameters of a given layer, e.g., leaving the traces on the same layer but changing the distance from the traces to a ground plane, has the same effect.

In this example, let's actually move the coupled traces to an inner layer, so that they are in a stripline (rather than microstrip) configuration.

To see the effect of changing the traces to an inner (stripline) stackup layer:

1. Open the stackup editor (from the Edit menu, choose Stackup). Point to the TOP layer and drag it with the mouse so that it is just below the VCC plane. Then drag the VCC plane so that it is the top layer in the stackup. Note that the TOP layer is now in a stripline configuration. Click OK.

2. Open the oscilloscope and re-simulate. The resulting crosstalk waveform at the victim net's receiver IC has a peak amplitude of approximately 350 mV — a reduction in crosstalk of about a factor of three compared to the original result.

3. Return the stackup to its original configuration (with the TOP layer on top and the VCC plane second), for use in the following section.

Notice that the change to a stripline configuration — which has the effect of significantly increasing the extent to which each trace "sees" the ground planes — generated a larger reduction in crosstalk than any other method we've tried. This will not always be the case (sometimes other parameters will be more dominant), but it does demonstrate that the choice of PCB stackup and its details can significantly affect crosstalk. These examples also show that many of the common "rules of thumb" regarding crosstalk are not strictly true.

About Trace Impedances, Capacitances, Inductances, and Delays

When you couple two or more transmission lines together, the electrical quantities that describe the lines become more-complex than in the single, uncoupled-line case: specifically, the impedances, capacitances, and inductances become matrix quantities, and if the lines are in a microstrip or buried-microstrip configuration, a mixture of propagation delays results.

Details on these phenomena are interesting but complex; if you want to know more about them generally, see Chapter 9. Note that LineSim handles all of these nuances for you automatically: there is no need to understand the sometimes-arcane details in order to set up crosstalk problems or get simulation valid results. However, if you are interested, this section describes briefly some of the effects present for the preceding example.

Impedances

When two or more transmission lines are coupled together, each line ceases to have a single characteristic-impedance value. Rather, the concept of impedance becomes shared between the lines and is expressed in terms of a matrix which shows both line-to-ground and line-to-line impedances.

To view the impedance matrix for the two-trace crosstalk example:

1. In the schematic, point to one of the transmission lines and right-click on it. The Edit Transmission Line dialog box opens.
2. Select the Field Solver tab. In the Numerical Results area, click the View button. The HyperLynx File Editor opens on a report file showing the details of the coupled lines' electrical parameters.
3. Scroll down in the report to see the Characteristic Impedance Matrix.

The diagonal elements in the matrix give the line-to-ground impedances, and the off-diagonal elements the line-to-line impedances. In this case, the line-to-ground values are approximately 72 ohms, and the line-to-line about 16 ohms. The line-to-line values are somewhat abstract; counter to what you might intuitively think, the lower the line-to-line value, the *weaker* the line-to-line coupling.

The beginning of the report file gives differential and common-mode impedance values for the two transmission lines. However, these parameters are primarily of interest when the lines are actually driven differentially, which in this example they were not. For details on differential lines and values, see Chapter

9. here, since the two lines are independent of each other, the line-to-ground impedance is most relevant, as the report indicates in its comments.

Capacitances and Inductances

As with characteristic impedance, in the case of coupled transmission lines, capacitance and inductance become matrix quantities.

To see the capacitance and inductance matrices for the two-trace crosstalk example:

1. Scroll in the report (see the preceding section for details on opening it) to just below the Characteristic Impedance Matrix.

Once again, line-to-ground values are along the matrix diagonal and the line-to-line values in the off-diagonal positions. The negative signs of the off-diagonal capacitance values are surprising, but correct; they arise in the formalism of the mathematics used to calculate coupling parameters. The off-diagonal values *do* represent actual physical trace-to-trace capacitances — you can ignore the sign convention.

Note: *It is important that these values be presented as negative, because if you ever transport the capacitance matrix to another tool, e.g., Hspice, you will get very wrong results if the negative signs are dropped.*

Notice that the values in the capacitance and inductance matrices are expressed per unit length, i.e., in units of pF and nH *per meter*. To get total capacitance and inductance values, multiply by the coupling-region length.

Delays

If a set of coupled traces is in a stripline configuration (i.e., lies between two planes, such that the traces “see” only one type of dielectric), then each trace has a single propagation velocity and delay. However, if the traces are in a microstrip or buried-microstrip configuration, then a *set* of velocities and delays arises, with each trace having certain percentages of its signal traveling with each delay. There are as many velocities as there traces coupled together.

Note: *This is an unfamiliar concept for many engineers and PCB designers. Fortunately, LineSim accounts for it automatically and you don't need to know any of the details unless you want to.*

For example, in this two-trace scenario where the transmission lines are in a microstrip configuration, each line supports two propagation "modes."

To view the details of the two propagation modes involved in the two-trace crosstalk example:

1. Scroll to the bottom of the report. Look for the Propagation Speeds and Percentage of Energy sections.

Here, each transmission line supports a mode which propagates signals at 64% of the speed of light, and another which propagates signals at 57% the speed of light. A given signal on either trace (see the energy matrix) will propagate half its energy in the first mode, and half in the second.

For symmetric trace-pair configurations, the two propagation modes also happen to be the differential and common signal modes. See Chapter 9 for details.

Electrical Values in the Schematic

LineSim's schematic editor has room to display only one characteristic impedance and delay for each transmission line present. In the general coupled case, the impedance is actually a matrix quantity; also, multiple delays will exist. Accordingly, for a given line, LineSim displays the line-to-ground impedance and an average delay value. Here, those values are 72 ohms and 1.13 ns, for each line.

A Differential-Pair Example

This example shows how to use LineSim's crosstalk-analysis option to design a pair of coupled, differential traces. Of particular interest is how to achieve a desired differential impedance.

Note: *The schematic for this example is installed in the HYPFILES directory as "XT Manual Coupled Differential.tln." However, you'll learn more if you create the schematic yourself as you follow the steps in the example.*

Note: *You can simulate differential pairs in the base LineSim product, too, but without the crosstalk option you cannot account for the effect of the coupling between the traces, or automatically calculate differential impedance.*

Application Description

Suppose you want to design and simulate a pair of traces that will be driven by a pair of differential drivers. The specifications for the differential trace pair are:

- trace widths of 6 mils
- trace separation can be varied, but must not decrease below 6 mils (for manufacturing reasons)
- board stackup includes two outer signal layers and two inner layers (inner layers are between ground and Vcc planes); differential pair can be located on any layer
- stackup dielectric thicknesses can be varied, but no dielectric layer can exceed 10 mils
- PCB dielectric constant is 4.3
- target differential impedance is 100 ohms
- differential drivers are from the ECLINPS100K family; they will be run at PECL (positive ECL) levels, i.e., with Vcc=5.0V and Vee=0.0V

Achieving a Specific Differential Impedance

For this example, let's first consider the goal of achieving a 100-ohm differential impedance. We'll vary other parameters (like trace separation and stackup layer) as needed.

"Differential impedance" is different than any of the impedances in a coupled pair's characteristic-impedance matrix. (For details on the characteristic-impedance matrix for coupled traces, see Chapter 9 and/or "A Simple Two-Trace Example" above.) Specifically, differential impedance is the trace-to-trace termination resistance required to perfectly terminate a differential pair *assuming that the signals driven on the pair are perfectly differential.* (A single trace-to-trace resistor will not completely terminate a pair that has some common-mode signal content.)

Note: *This definition of differential impedance assumes that if the traces in the pair are on a microstrip or buried-microstrip layer, the traces are symmetric, i.e., on the same layer, have the same width and thickness, and so forth. See Chapter 6 for more details.*

Fortunately, any time you create a coupling region that contains only two transmission lines, LineSim tailors its field-solver output and shows you not only values from the lines' impedance matrix, but also the differential impedance.

Creating the Schematic

Note: *This manual assumes that you already understand the basics of creating and editing schematics in LineSim. The emphasis here is on the additional features that LineSim's crosstalk option adds to base LineSim. If you are unfamiliar with the base LineSim product, refer to the LineSim User's Guide or the online Help system.*

Create a schematic representing the differential pair (see Figure 5-5):

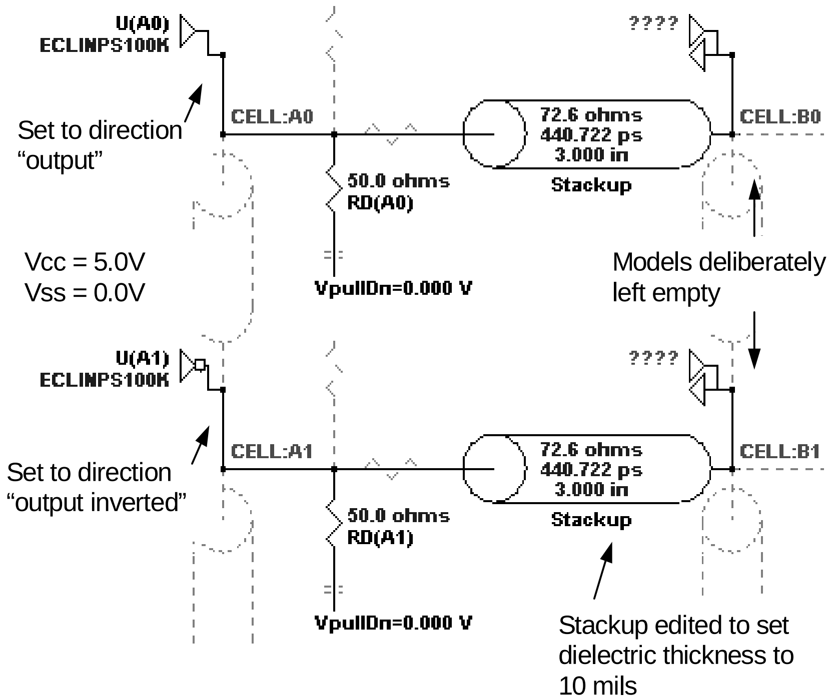
1. Create a simple schematic that contains two horizontal transmission lines, each with ICs at their left and right ends. The two lines must be in the same schematic column (e.g., one above the other; see Figure 5-5).

Note: *You can couple together transmission lines that are not in the same schematic column or row, but this is an advanced feature that most users don't need. For details on enabling the "coupling dot" mechanism that allows this, see Chapter 4.*

2. Model the left-end ICs with the “ECLINPS100K” model from the library GENERIC.MOD. Leave the right-end ICs with no model; we’ll use these positions only to place oscilloscope probes.
3. Change the driving direction of the upper line’s left-end IC to be “Output.” Change the direction of the lower line’s left-end IC to be “Output Inverted.” (These settings causes the transmission lines to be differentially driven.)
4. Since the driver-IC technology is ECL, we need biasing resistors for the output buffers. Add a pull-down resistor to each of the driver ICs. Set the resistor values to 50 ohms.
5. Since our specification calls for PECL operating levels, use the Edit/Power Supplies dialog box to change Vcc to 5.0V and Vss to 0.0V.
6. Use the stackup editor to change the dielectric between the Top layer and Vcc plane to be 10 mils thick.

The schematic you’ve created so far should look like Figure 5-5.

Figure 5-5: Schematic for differential-pair example



Adding Coupling Information to the Schematic

Continue by placing the transmission lines in a coupling region:

1. Point to the upper transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Make the line coupled by clicking on the Stackup radio button **under the Coupled heading.** The Add to Coupling Region tab is automatically selected.

2. In the Coupling Regions list, double-click on the New-Coupling item. This tells LineSim to automatically create a new coupling and add the transmission line to it. The dialog box closes.

3. Repeat step 1, but for the lower transmission line.

4. In the Coupling Regions list, double-click on the Coupling0001 item. This adds the lower transmission line to the coupling region that was automatically created for the upper transmission line. The dialog box closes.

Back in the schematic editor, there are two “clues” that the transmission lines are coupled together: if you point to either line, it highlights in yellow, and the transmission lines are joined by a dashed yellow “ratsnest” line; also, a “coupling dot” appears on the left end of each transmission line. (For more details on coupling dots and what they mean, see Chapter 4.)

Edit the new coupling region’s cross section to change the trace widths to 6 mils:

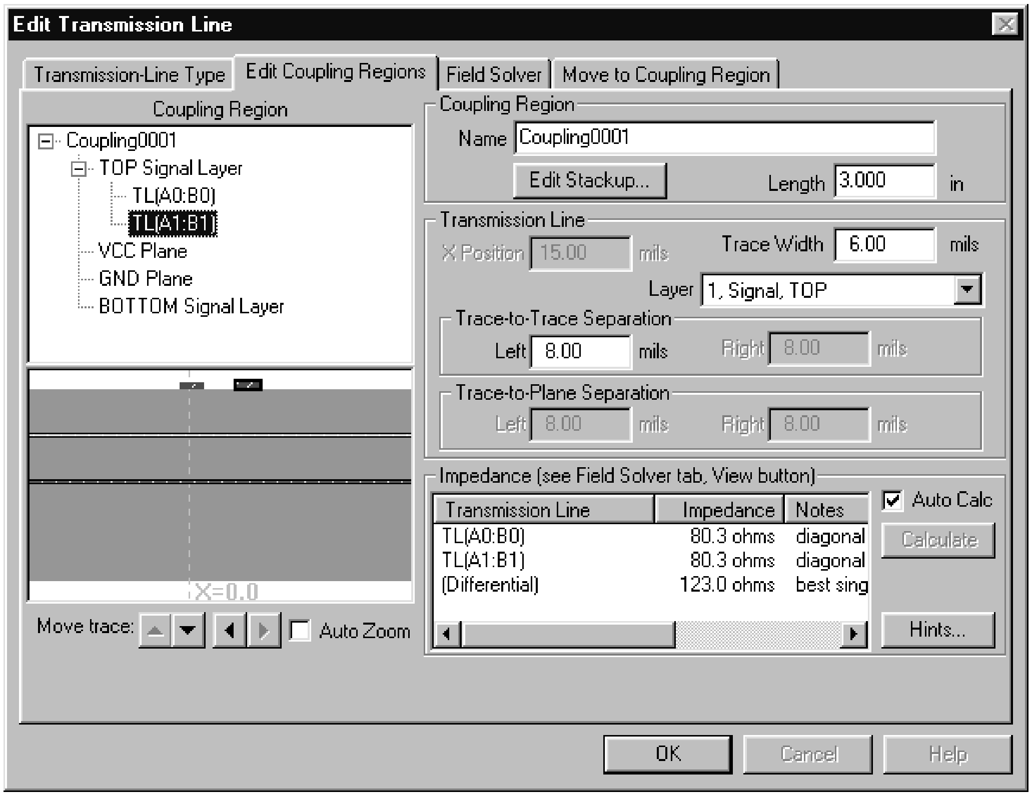
1. Point to either transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Select the Edit Coupling Regions tab. Notice that the Coupling Region tree list shows the new coupling region (auto-named by LineSim “Coupling0001”) expanded, so that the current stackup layers and the transmission lines that were added to the region are visible.

2. Our specification called for 6-mil-wide traces. LineSim defaults to 8, so in the Transmission Line area, in the Trace Width edit box, type “6”. This changes the width of the currently highlighted trace.

3. Highlight the other trace in the tree list (or click on it in the graphical viewer), and change its width to 6 mils also.

Figure 5-6 shows what the Edit Coupling Regions tab should look like at this point.

Figure 5-6: Contents of Edit Coupling Regions tab for differential-pair example



Determine Cross-Section Geometry to Give 100-Ohm Differential Impedance

At this point, we have no idea whether the default trace separation (set to 8 mils) and the trace's position on the TOP stackup give us the desired 100-ohm differential impedance. However, LineSim can easily calculate the value for us. If the impedance needs to be adjusted, we can change aspects of the cross section to "zero in" on the specified impedance.

For coupling-region cross sections that contain only two traces, LineSim adjusts the contents of the list box in the Impedance area in the Edit Coupling Regions dialog box, and automatically calculates differential impedance. Notice that in this case, with the geometry we have, the differential impedance is higher than our target value: 123 ohms versus the desired 100 ohms.

To decrease the differential impedance, we must either increase the coupling between the traces or decrease the trace-to-ground impedances (or both). (For more details on the theory underlying differential impedance, see Chapter 9.)

Adjust the coupling region's cross section to achieve the 100-ohm differential impedance:

1. First, let's try moving the traces closer together. (Closer spacing should increase the trace-to-trace coupling and have little effect on the trace-to-ground impedance.) According to our specification, the minimum acceptable trace separation is 6 mils, so let's try that.

   In the Coupling Region list box, highlight trace TL(A0:B0). Then in the Trace-to-Trace Separation area, in the Right edit box, type "6."

   Now (in the Impedance list box) the differential impedance is 113 ohms — closer than before, but still more than 10% away from the goal of 100 ohms.

2. Next, try changing the thickness of the dielectric between layer TOP and the VCC plane. But notice that it is not completely obvious whether the thickness should increase (which would increase both the trace-to-trace coupling and the trace-to-ground impedance) or decrease (would decrease both trace-to-trace coupling and trace-to-ground Z_0). The easiest way to find out is to try.

   First, in the Edit Coupling Regions dialog box, in the Coupling Region area, click the Edit Stackup button, and change the dielectric thickness between layers TOP and VCC to 15 mils. (If you are unfamiliar with the stackup editor, see the LineSim User's Guide.) Close the stackup editor. Now the differential impedance is 118 ohms — higher than before, not lower.

3. Since increasing the dielectric thickness from 10 mils moved the differential impedance in the wrong direction, let's next try decreasing the thickness. Again, click the Edit Stackup button; this time, change the thickness to 5 mils.

   Now the differential impedance is 97 ohms — a big change, and very close to the desired value. One more small change to the cross-section geometry will probably achieve the goal. Let's try "tweaking" the trace separation slightly.

4. With transmission line TL(A0:B0) highlighted in the Coupling Region list box, in the Trace-to-Trace Separation area, in the Right edit box, type "7" to widen the separation slightly to 7 mils. Now the differential impedance is 100 ohms — exactly what we wanted.

Simulating to See the Differential Waveforms

Now that the differential impedance has been achieved, we're ready to simulate to see how the actual signal waveforms look.

Run the oscilloscope to see the signal waveforms:

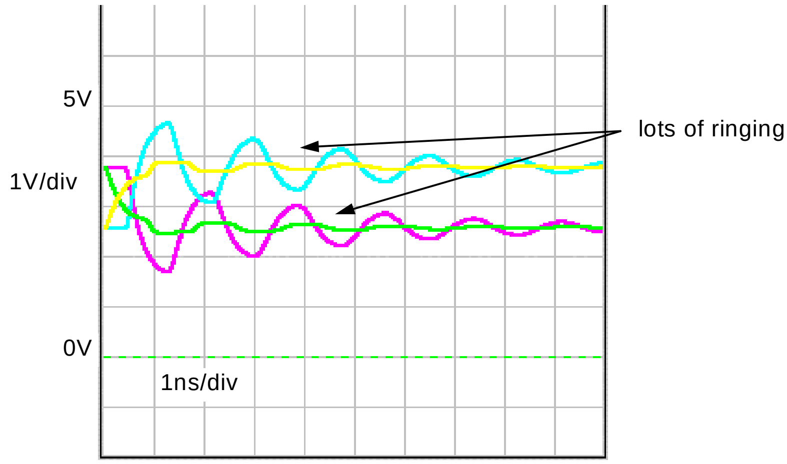
1. Click OK to close the Edit Coupling Regions dialog box.

2. From the Scope/Sim menu, choose Run Scope. The Digital Oscilloscope dialog box opens. Verify that the IC Modeling setting is Typical.

3. Click the Start Simulation button to run the simulator and view the resulting waveform.

The waveforms are switching between the typical PECL levels. But there is a considerable amount of ringing at the receiver end of both traces (purple and blue waveforms). Figure 5-7 shows the waveforms.

Figure 5-7: Differential waveforms, *without* differential termination

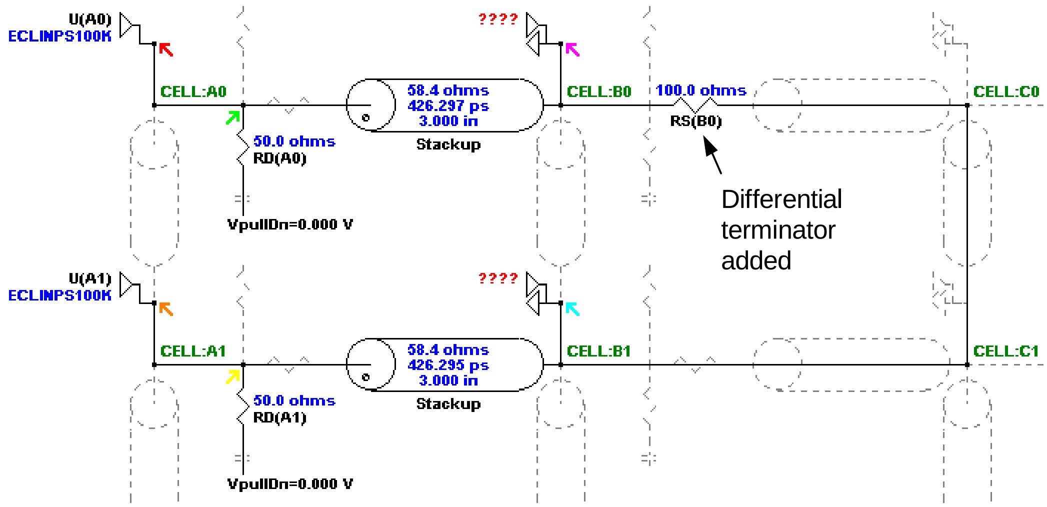


Actually, the extent of the ringing is not very surprising, because the differential pair is not terminated at its far end. Since we are driving with a nearly perfect differential signal, we should be able to terminate with only a trace-to-trace resistor of value equaling the differential impedance.

Add a 100-ohm trace-to-trace terminator between the far ends of the transmission lines:

1. In the schematic editor, click in a series resistor just to the right of the upper transmission line.
2. Then use the nearby transmission lines to make a short-circuited “U turn” that connects the far end of the resistor back to the end of the lower transmission line. See Figure 5-8.
3. Change the resistor’s value to match the line’s differential impedance: 100 ohms.

Figure 5-8: Differential-pair schematic with differential terminating resistor added

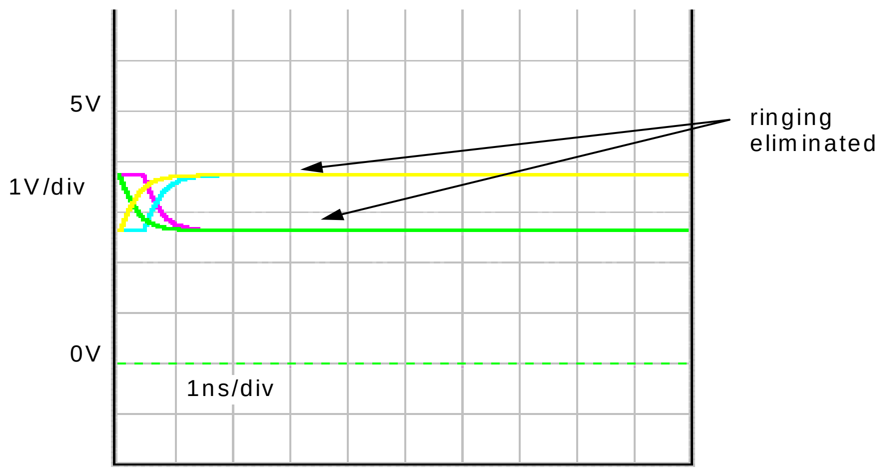


Re-simulate to see the effect of adding the termination:

1. Re-open the oscilloscope, and click the Erase button to erase the previous simulation's waveforms.
2. Click Start to re-simulate.

The new waveforms are nearly perfect: all of the ringing has been eliminated. (Compare to Figure 5-7.) The differential termination works perfectly. See Figure 5-9.

Figure 5-9: Differential waveforms, *with* differential termination



Determining Trace Separation on a Bus

This example shows how to use LineSim's crosstalk-analysis option to design a typical digital bus. Of particular interest is how to plan the geometric properties of the bus (especially trace separation) to keep inter-signal crosstalk less than a certain maximum amount.

Note: *The schematic for this example is installed in the HYPFILES directory as "XT Manual Trace Separation.tln." However, you'll learn more if you create the schematic yourself as you follow the steps in the example.*

Application Description

Suppose you want to design a bus such that no signal in the bus receives more than a certain amount of crosstalk when its neighboring signals switch. The specifications for the bus are:

- stripline traces (i.e., inner-layer traces, “sandwiched” between ground/Vcc planes)
- 12 inches long
- 8-mil wide traces
- no less than 8 mils between traces (edge to edge)
- 1.0-ounce copper
- PCB dielectric constant of 4.3
- all traces driven with fast 3.3-V CMOS drivers
- no more than 500 mV of crosstalk from neighboring traces

Creating the Schematic In LineSim

How Many Traces to Simulate?

A typical bus in a modern digital system contains many physically parallel traces — 16, 32, 64, maybe even more signals. **However, when you simulate to predict crosstalk on such a bus, you definitely do not bother to simulate all of the signals simultaneously.** Rather, you take advantage of the fact that the crosstalk driven into a given “victim” signal comes very predominantly from two other traces: the neighboring ones on either side of the victim trace. **Therefore, most typically, you would simulate only a set of three traces, as demonstrated in this example.**

Note: *There are several disadvantages to simulating an unnecessarily large number of traces in a bus. First, simulating extra traces requires extra set-up time. Second, extra traces require extra computation time, both in the field solver and the transient simulator — and the relationship between number of traces and execution time is very non-linear.*

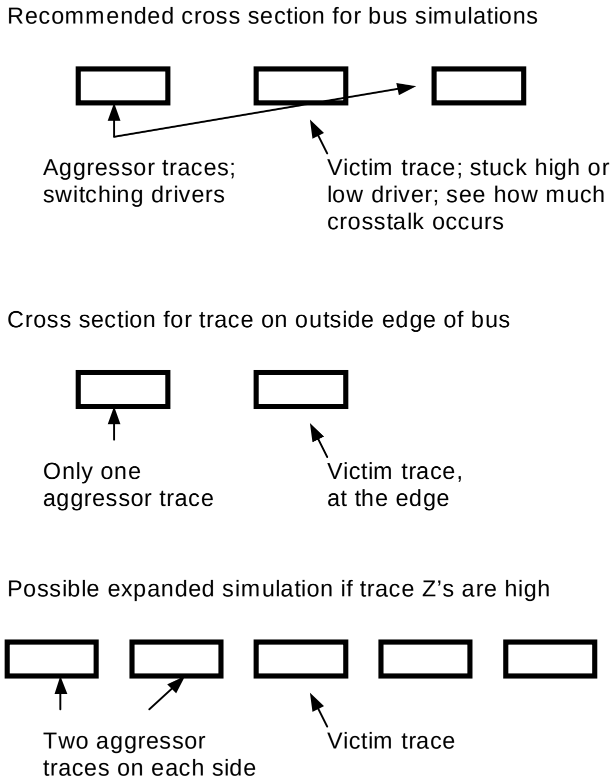
Possible Exceptions

There are possible exceptions. You may want to run a different simulation to see how much crosstalk occurs on the two traces that lie on the outer edges of the bus. Since these traces have only one neighboring trace, the simulation would involve only two transmission lines, instead of three. However, because each of these outside traces has only one “aggressor” trace, the crosstalk is very likely to be less than for traces “inside” the bus, so you might not bother simulating the outside traces at all.

Also, if a bus’ signal layer is located far from a ground/Vcc plane, so that the characteristic impedances of its traces are fairly high, then — because the traces “see” each other strongly — it is possible that you might want to include five traces instead of three, to see if the extra two outer traces reduce the amount of energy going into the victim trace. For most cross sections, though, you’ll find that the amount crosstalk delivered to a central victim trace is hardly affected by the presence (or lack) of traces two positions to the right or left.

Figure 5-10 summarizes these comments.

Figure 5-10: Possible cross sections for simulating crosstalk on a bus trace



Drawing the Schematic

Note: *This manual assumes that you already understand the basics of creating and editing schematics in LineSim. The emphasis here is on the additional features that LineSim's crosstalk option adds to base LineSim. If you are unfamiliar with the base LineSim product, refer to the LineSim User's Guide or the online Help system.*

Create a schematic representing a three-trace-wide section of the bus (see Figure 5-11):

1. Create a schematic that contains three horizontal transmission lines, each with ICs at their left and right end. The three lines must be in the same schematic column (e.g., one above the other; see Figure 5-11).

Note: *You can couple together transmission lines that are not in the same schematic column or row, but this is an advanced feature that most users don't need. For details on enabling the "coupling dot" mechanism that allows this, see Chapter 4.*

---

2. Model all of the ICs with the "CMOS,3.3V,FAST" model from the library EASY.MOD. (Hint: use the Paste All button in the Assign IC Models dialog box to quickly apply this model to all ICs in the schematic.)

3. Change the driving direction of the upper and lower line's left-end ICs to be "Output." Change the direction of the middle line's left-end IC to be "Stuck Low." Leave the right-end ICs set as "Input."

4. Open the stackup editor. Add two signals layers between the VCC and GND planes. Edit the two new layers and name them "Inner1" and "Inner2." (If you're unfamiliar with how to edit stackups in LineSim, see the LineSim User's Guide.)

5. Still in the stackup editor, change the dielectric thickness between the TOP layer and VCC plane to 10 mils. Change the thickness between the GND plane and BOTTOM layer to 10 mils. Exit the stackup editor.

The schematic you've created so far should look like Figure 5-11, and the stackup like Figure 5-12.

Figure 5-11: Schematic for trace-separation example

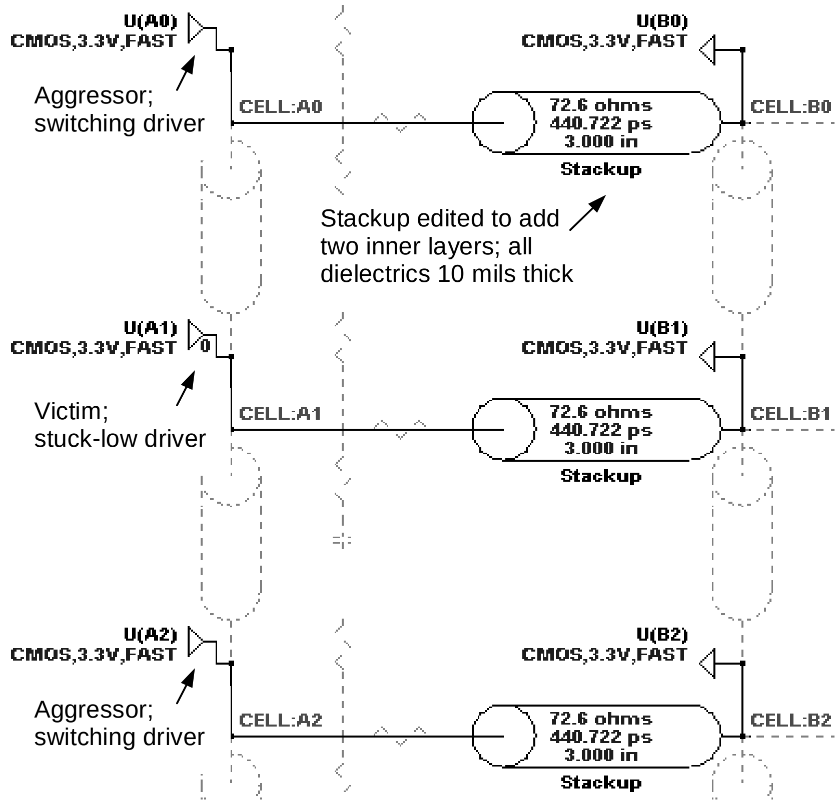
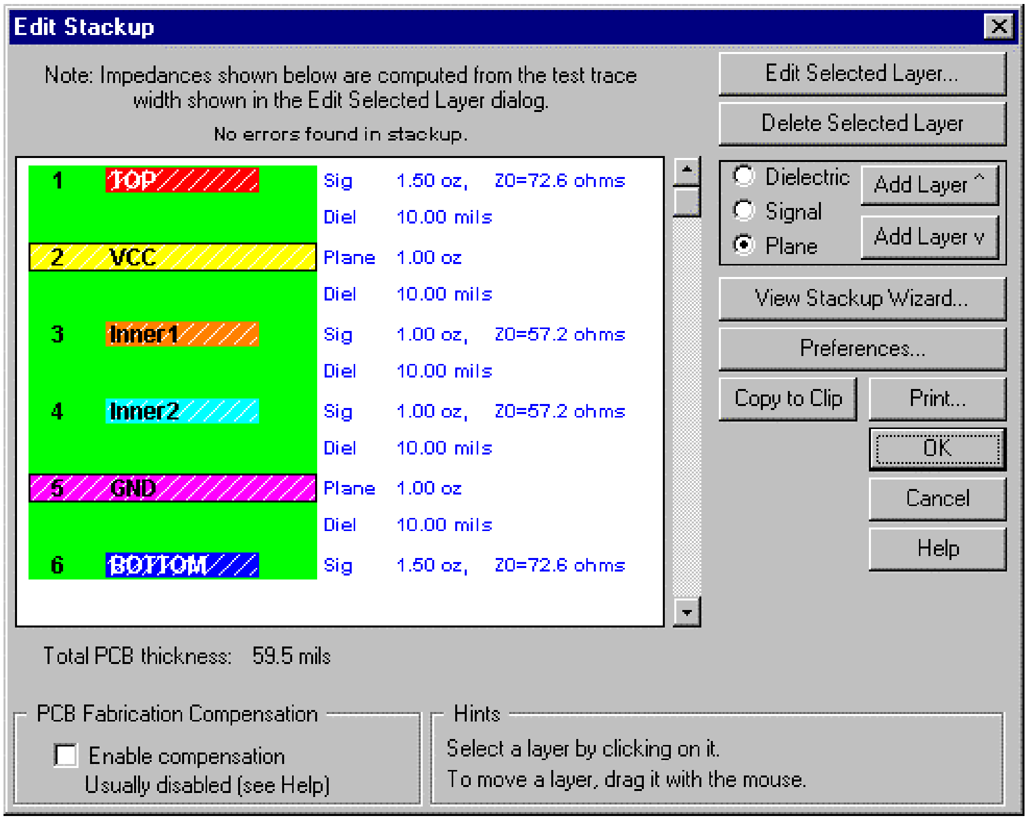


Figure 5-12: Stackup for trace-separation example



By making the driver ICs at the left ends of the upper and lower trace switching “outputs,” you have made those traces the “aggressors” in the simulation — they model the traces on either side of the “victim” trace. By making the driver IC at the left end of the middle trace “stuck low,” you have made this trace the “victim.”

It is important to include non-switching (i.e., “stuck”) driver ICs on victim nets, because on a real PCB, victim nets <u>do</u> have drivers. If you omit the “stuck” driver(s), the crosstalk waveforms will look much

different than if you include them. This occurs because driver ICs are typically low impedance and will reflect, rather than absorb, crosstalk signals.

In this example, we're bothering to simulate only with the victim-net driver stuck low. However, in general, you should check both the stuck-low and stuck-high cases, to see which generates more crosstalk.

Adding Coupling Information to the Schematic

Continue by placing the transmission lines in a coupling region:

1. Point to the upper transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Make the line coupled by clicking on the Stackup radio button **under the Coupled heading.** The Add to Coupling Region tab is automatically selected.

2. In the Coupling Regions list, double-click on the New-Coupling item. This tells LineSim to automatically create a new coupling and add the transmission line to it. The dialog box closes.

3. Repeat step 1, but for the middle transmission line. In the Coupling Regions list, double-click on the Coupling0001 item. This adds the middle transmission line to the coupling region that was automatically created for the upper transmission line. The dialog box closes.

4. Repeat step 3, but for the lower transmission line.

Back in the schematic editor, there are two "clues" that the transmission lines are coupled together: if you point to any line, it highlights in yellow, and the transmission lines are joined by dashed yellow "ratsnest" lines; also, a "coupling dot" appears on the left end of each transmission line. (For more details on coupling dots and what they mean, see Chapter 4.)

Define the new coupling region's cross section:

1. Point to any transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Click on the Edit Coupling Regions tab.

2. By default, the three traces in the coupling region have been placed on the TOP layer in the stackup. Our specification, however, calls for them to be

on an inner layer.

Highlight the first trace in the list (transmission line “TL(A0:B0)”). In the Transmission Line area, pull down the Layer combo box, and select layer “Inner1.” The highlighted trace moves to layer Inner1; the graphical view of the cross section changes accordingly.

3. Repeat step 2 for the other two traces, first “TL(A1:B1)” and then “TL(A2:B2),” moving them also to layer Inner1.

When you move traces between layers, you must be careful that you do not accidentally change the order of the traces. In this example, we added the traces to the coupling region in top-to-bottom order so that the middle trace in the schematic (the victim trace) was in the middle position in the coupling-region cross section. Depending on the details of how you move the traces one-by-one to the inner layer, the trace order left-to-right might change.

To determine if the trace order is still correct, either:

- look at the list of transmission lines below stackup layer Inner1 and verify that the order is TL(A0:B0), then TL(A1:B1), then TL(A2:B2)

OR

- click on the first transmission line in layer Inner1’s list, then the next, then next; and verify that the black highlight in the graphical cross-section viewer moves from the left to the middle to the right trace

If the trace order is not correct, highlight the trace you want to move to the right or left, and click the right-arrow or left-arrow button (below the graphical viewer) to “swap” the highlighted trace with its neighbor. For more details on moving traces in a coupling region, see Chapter 2, section “Cross-Section Properties.”

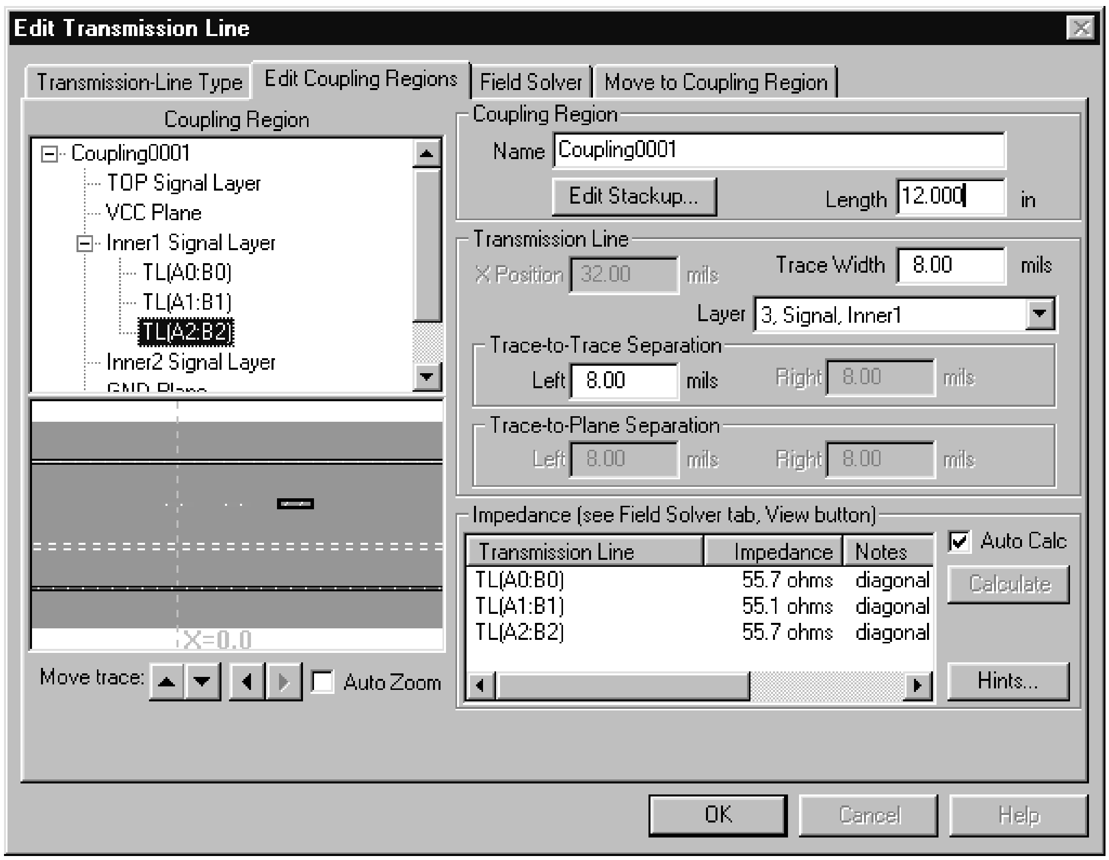
LineSim defaults to 8-mil wide traces and 8-mil trace separation (unless you’ve changed the Default Trace Separations setting in the Preferences dialog box; see Chapter 2, section “Setting the Default Trace-to-Trace and Trace-to-Plane Separations” for details); these match the problem description in “Application Description” above.

Finish defining the coupling region's cross section:

1. The bus was specified as 12 inches long. In the Coupling Region area, in the Length edit box, type "12."

Figure 5-13 shows what the Edit Coupling Regions tab should look like at this point.

Figure 5-13: Contents of Edit Coupling Regions tab for trace-separation example



Simulating to See Crosstalk Waveforms

Now that the schematic has been created and coupling information has been added to it, we're ready to simulate to see how much crosstalk would occur on a typical trace in the bus when the two neighboring traces are both actively driven.

Run the oscilloscope to see the crosstalk waveforms:

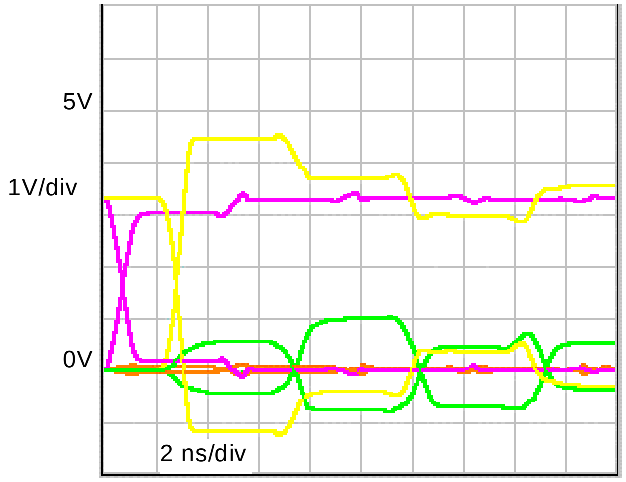
1. Click OK to close the Edit Coupling Regions dialog box.
2. From the Scope/Sim menu, choose Run Scope. The Digital Oscilloscope dialog box opens.
3. In the Horizontal area, click the Scale right-arrow button to increase the timebase to 2 ns/div.
4. In the Driver Waveform area, click on the Falling Edge radio button. Verify that the IC Modeling setting is Typical.
5. Click the Start Simulation button to run the simulator and view the falling-edge waveform.
6. In the Driver Waveform area, click on the Rising Edge radio button. Click Start Simulation to view the rising-edge waveform.

A significant amount of crosstalk appears on the victim net at its receiver IC; the worst-case amplitude is about 1.0V. This significantly exceeds the 500-mV threshold in our specification.

The crosstalk is a combination of forward and backward components; the backward portion is reflected initially from the victim-net "stuck low" driver IC, and sent down the victim line along with the forward portion. The backward portion is easily identified because its time duration is twice the delay length of the victim line.

Figure 5-14 shows what the crosstalk waveforms should look like.

Figure 5-14: Initial crosstalk waveform for trace-separation example



In order to meet our 500-mV crosstalk limit, we must alter the coupling-region cross section.

Adjust the coupling region's cross section to achieve the 500-mV crosstalk limit:

1. First, let's try moving the traces further apart. Point to one of the transmission lines in the schematic, and right-click on it. The Edit Transmission Line dialog box opens. Select the Edit Coupling Regions tab.
2. Highlight the first trace in the list. In the Trace-to-Trace Separation area, in the Right edit box, type "10" to increase the separation between the left and middle trace to 10 mils.
3. Highlight the last trace in the list. In the Left edit box, type "10" to increase the separation between the right and middle trace to 10 mils. Verify in the graphical viewer that the spacing between the three traces is symmetrical.
4. Close the dialog box and re-open the oscilloscope. Re-simulate to see the new waveforms. It is only necessary to run a rising-edge simulation,

because the worst-case crosstalk occurred previously for the rising edge.

The crosstalk is reduced, but not significantly (it is still about 810 mV). Further changes are needed.

5. Try increasing the trace separation to 12 mils. Repeat steps 1 – 4, changing the trace separations to 12 mils, then re-simulating.

   The crosstalk is reduced more, to about 660 mV. But this is still short of our goal, and further increases in trace separation are likely to waste too much board space. Let's try varying a different parameter to see if we can find a more-effective way of decreasing the crosstalk.

6. Open the stackup editor, and change the dielectric thickness between the VCC plane and layer Inner1 to 5 mils. Change the thickness between layer Inner 2 and the GND plane to 5 mils also.

7. Re-open the oscilloscope and re-simulate. Now the crosstalk *is* sharply reduced, to about 220 mV. In this case, a simple stackup-thickness change appears to be a good way to control crosstalk on the bus.

8. Since 220 mV is well below our allowable threshold, let's see if we can move the traces back closer together than 12 mils apart, and still meet the 500-mV limit.

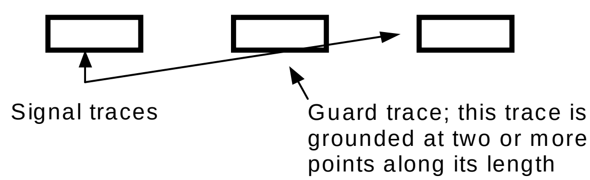
   Right-click on a transmission line; select the Edit Coupling Regions tab. Highlight the first trace and type “8” into the Right Trace-to-Trace Separation edit box. Repeat for the last trace and its Left edit box. This moves the traces back to being 8 mils apart.

9. Re-open the oscilloscope and re-simulate. The crosstalk increases to about 400 mV. This value meets our specification, with about a 25% margin.

Assessing the Effect of Guard Traces

Some high-speed designers recommend using “guard traces” to reduce crosstalk between sensitive signals. A “guard trace” is a grounded trace that lies between two other traces (see Figure 5-15). According to proponents, guard traces tend to electromagnetically isolate the traces they lie between, and therefore minimize the crosstalk between the separated traces.

Figure 5-15: Cross section showing a guard trace between two signal traces



The following example shows how you can simulate guard traces in LineSim. Once the method of analyzing a guard trace is explained, we’ll try to assess how effective a guard trace really is. What would happen, for example, if instead of bothering to add a guard trace, you only moved apart the other two traces by the amount that the guard trace would have required, if you’d actually added it?

Note: *The schematic for this example is installed in the HYPFILES directory as “XT Manual Guard Trace.tln.” However, you’ll learn more if you create the schematic yourself as you follow the steps in the example.*

Application Description

Here’s the description for our guard-trace test case:

- microstrip traces (i.e., outer-layer traces), 12 inches long
- 8-mil wide traces with 8-mil separation

- 0.5-ounce base copper with 1.0 ounce of plating; PCB dielectric constant of 4.3; dielectric thickness between outer signal layer and plane layer is 7 mils
- all traces driven with fast 3.3-V CMOS drivers
- simulate first to see how much crosstalk occurs when two traces are adjacent; then separate them with an end-grounded guard trace and re-simulate to see how significantly the crosstalk is reduced

Creating the Schematic

Note: *This manual assumes that you already understand the basics of creating and editing schematics in LineSim. The emphasis here is on the additional features that LineSim's crosstalk option adds to base LineSim. If you are unfamiliar with the base LineSim product, refer to the LineSim User's Guide or the online Help system.*

Create a schematic representing the two traces which will eventually be separated by a guard trace; initially leave them separated by 8 mils (see Figure 5-16):

1. Create a simple schematic that contains two horizontal transmission lines, each with ICs at their left and right ends. However, to allow room for the eventual addition of a guard trace, place the transmission lines in the first and *third* rows of the schematic (i.e., leave the second row open for the guard trace to be added). The two lines must be in the same schematic column (e.g., one above the other; see Figure 5-16).

Note: *You can couple together transmission lines that are not in the same schematic column or row, but this is an advanced feature that most users don't need. For details on enabling the "coupling dot" mechanism that allows this, see Chapter 4.*

2. Model all of the ICs with the "CMOS,3.3V,FAST" model from the library EASY.MOD. (Hint: use the Paste All button in the Assign IC Models dialog box to quickly apply this model to all ICs in the schematic.)

3. Change the driving direction of U(A0), the upper line's left-end IC, to be "Output." Change the direction of U(A2), the lower line's left-end IC, to be "Stuck High." Leave the right-end ICs set as "Input."

4. Use the stackup editor to change the dielectric between the TOP layer and VCC plane to be 7 mils thick. For symmetry (even though it won't affect the simulation), also change the dielectric between the GND plane and BOTTOM layer to be 7 mils. Exit the stackup editor.

The schematic you've created so far should look like Figure 5-16, and the stackup like Figure 5-17.

Figure 5-16: Schematic for guard-trace example

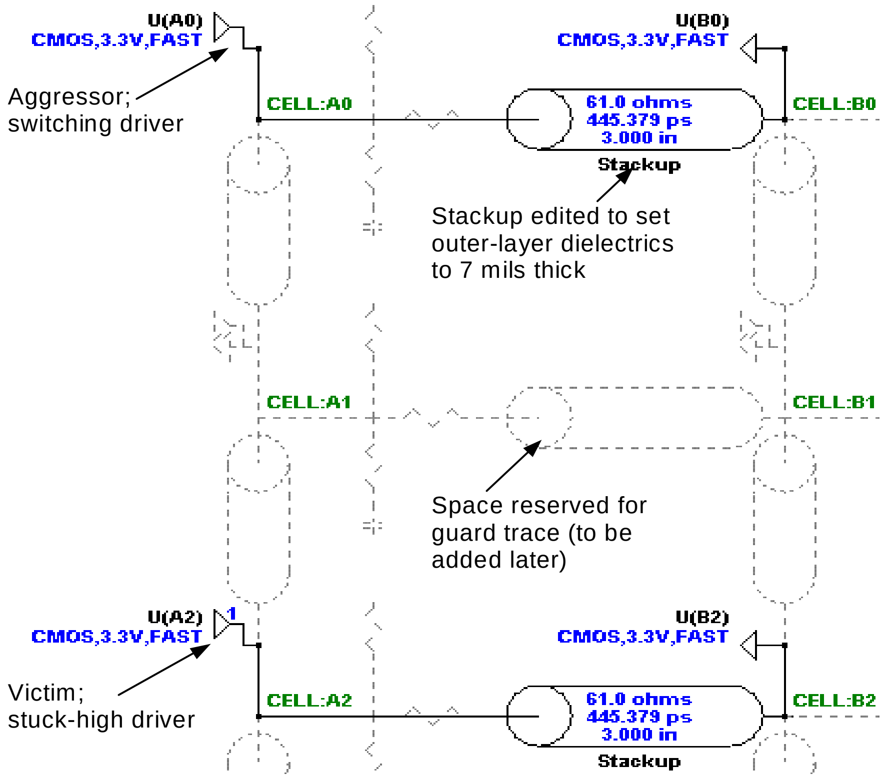
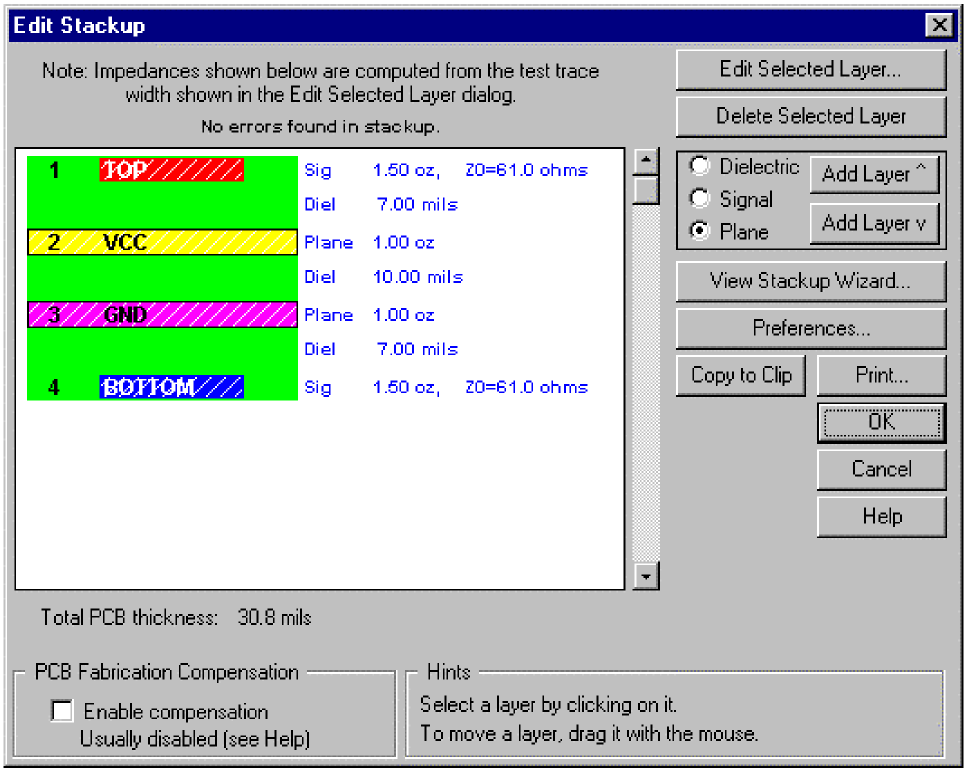


Figure 5-17: Stackup for guard-trace example



By making the driver IC at the left end of the upper trace a switching “output,” you have made that trace the “aggressor” in the simulation. By making the driver IC at the left end of the lower trace “stuck high,” you have made this trace the “victim.”

It is important to include non-switching (i.e., “stuck”) driver ICs on victim nets, because on a real PCB, victim nets do have drivers. If you omit the “stuck” driver(s), the crosstalk waveforms will look much different than if you include them. This occurs because driver ICs are typically low impedance and will reflect, rather than absorb, crosstalk signals.

In this example, we're bothering to simulate only with the victim-net driver stuck high. However, in general, you should check both the stuck-low and stuck-high cases, to see which generates more crosstalk.

Adding Coupling Information to the Schematic

Continue by placing the transmission lines in a coupling region:

1. Point to the upper transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Make the line coupled by clicking on the Stackup radio button **under the Coupled heading.** The Add to Coupling Region tab is automatically selected.

2. In the Coupling Regions list, double-click on the New-Coupling item. This tells LineSim to automatically create a new coupling and add the transmission line to it. The dialog box closes.

3. Repeat step 1, but for the lower transmission line. In the Coupling Regions list, double-click on the Coupling0001 item. This adds the lower transmission line to the coupling region that was automatically created for the upper transmission line. The dialog box closes.

Back in the schematic editor, there are two "clues" that the transmission lines are coupled together: if you point to any line, it highlights in yellow, and the transmission lines are joined by dashed yellow "ratsnest" lines; also, a "coupling dot" appears on the left end of each transmission line. (For more details on coupling dots and what they mean, see Chapter 4.)

LineSim defaults to 8-mil wide traces and 8-mil trace separation (unless you've changed the Default Trace Separations setting in the Preferences dialog box; see Chapter 2, section "Setting the Default Trace-to-Trace and Trace-to-Plane Separations" for details); these match the problem description in "Application Description" above. However, the default coupling-region length does not.

Edit the coupling region to change the length to 12 inches:

1. Point to either transmission line, and right-click on it. The Edit Transmission Line dialog box opens. Select the Edit Coupling Regions tab. Notice that the Coupling Region tree list shows the new coupling region (auto-named by LineSim "Coupling0001") expanded, so that the current

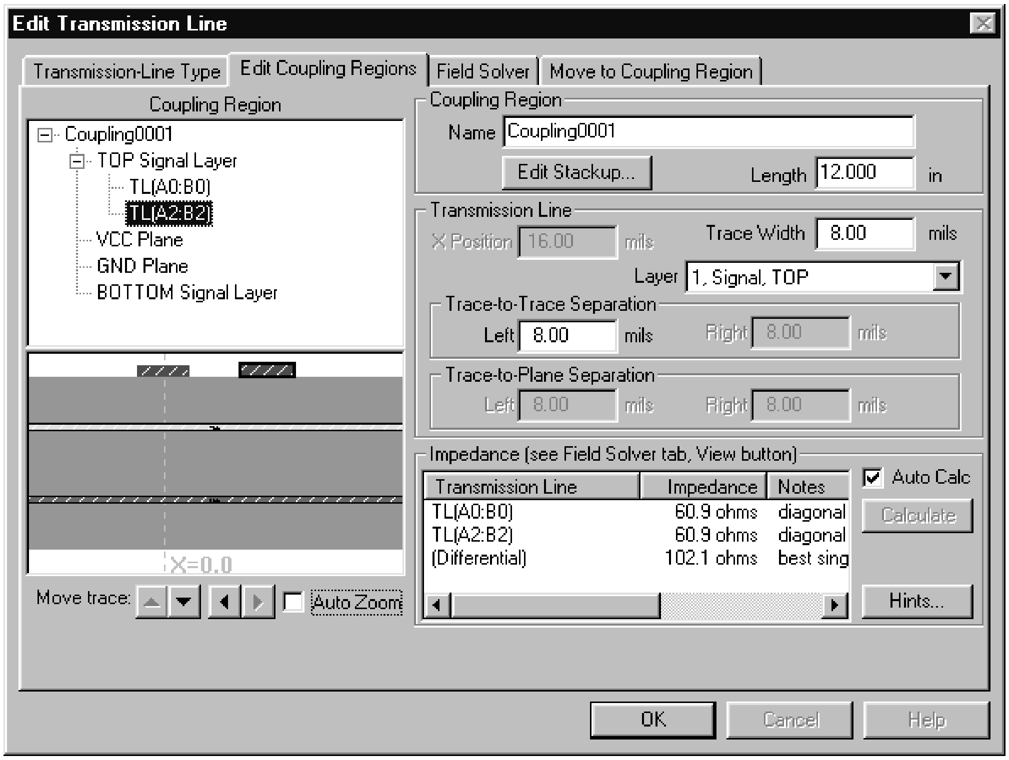
stackup layers and the transmission lines that were added to the region are visible.

2. In the Coupling Region area, in the Length edit box, type “12”.

3. Our specification also called for 8-mil-wide traces separated by 8 mils. Verify that for whichever trace is currently highlighted, the Trace Width edit box shows “8” and the Trace-to-Trace Separation is “8”. In the Coupling Region “tree list” or the graphical viewer, click once to highlight the opposite trace; verify that its Trace Width is also 8 mils.

Figure 5-18 shows what the Edit Coupling Regions tab should look like at this point.

Figure 5-18: Contents of Edit Coupling Regions tab for guard-trace example



Simulate to See the Amount of Crosstalk

Simulate the schematic you've entered to see how much crosstalk will occur between the two traces.

Run the oscilloscope to see the signal waveforms:

1. From the Scope/Sim menu, choose Run Scope. The Digital Oscilloscope dialog box opens. Verify that the IC Modeling setting is Typical and the Driver Waveform is set to Edge, Falling Edge.

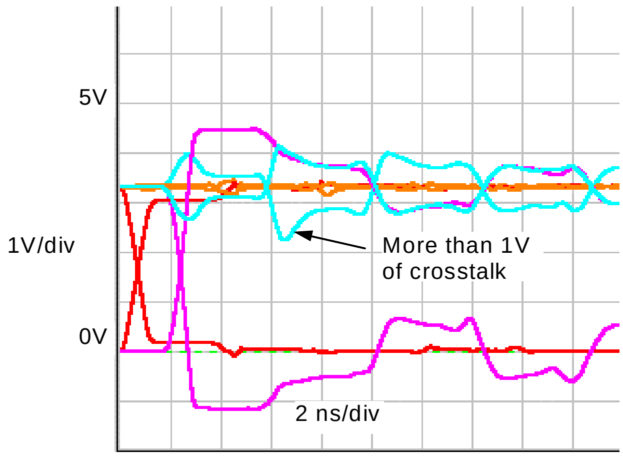
2. In the Horizontal Scale area, click the right-arrow button once to increase the timebase to 2 ns/div.

3. Click the Start Simulation button to run the simulator and view the falling-edge waveform.

4. In the Driver Waveform area, click the Rising Edge button. Then click Start Simulation to display the rising-edge waveform.

The light blue waveform shows the crosstalk at the receiver end of the victim trace. Note that the worst-case crosstalk amplitude is about 1.07 V, which is considerable. Figure 5-19 shows the waveforms.

Figure 5-19: Guard-trace waveforms, *before* the guard trace is added.



Add the Guard Trace

Now let's add a guard trace to the schematic, between the aggressor and victim traces, and see how much it reduces the crosstalk we just observed. A guard trace is simply a trace that runs between two other traces, and is grounded. But what exactly is meant by "grounded"?

Grounding a Guard Trace

Ideally, a guard trace would be perfectly grounded, i.e., it would somehow be constructed such that at every point along it, the voltage was always exactly 0V. However, on a real PCB, it is not possible to build such a trace. The best you can do is to route a trace and then tap it at one or more points into your board's ground plane.

But doing so doesn't cause the entire trace to sit at 0V — only those points that are connected directly to the ground plane are actually grounded. Currents can appear at other points on the trace due to electromagnetic coupling, which means that non-0 voltages will appear. However, the more frequently you connect a guard trace to ground, the smaller these unwanted currents and voltages will be.

Note: *Technically, even if you "stitched" a guard trace to ground with as many vias as you could fit along its length (1000, say), you still wouldn't quite achieve a perfect 0V. Why? Because the vias that connect to the ground plane are not perfect connections (even though they may be close), and can develop an small amount of voltage drop themselves. Also, the ground plane itself isn't ideal; as currents flow in it, the plane develops drops that elevate portions of it above 0V. However, these effects are secondary and minor compared to the first-order problem of where and how often you connect a guard trace to the ground plane.*

One additional point: it isn't necessary to connect a guard trace to a ground plane specifically; you could achieve the same effect by tying the trace to any power-supply plane, if the other plane is well bypassed to the ground plane so that currents flowing in the other plane can eventually return easily to ground.

Thus, when you simulate a ground trace in LineSim, you must decide in how many places and where you will connect the trace to ground. Being able to specify this level of detail ensures that you get a realistic simulation — much better than if LineSim treated ground traces as "ideal."

Vias in LineSim

How do you create a trace-to-ground-plane via in LineSim? A 0-ohm pull-down resistor to 0V works nicely for this. Not only does the resistor tie the trace to ground at the point at which you place it, but it also does so *non-ideally:* resistors in LineSim have parasitic R, L, and C which you can use to imitate a real via's non-ideal characteristics. In fact, the default parasitic values for a LineSim resistor make at least a reasonable via model, so often you can just insert a 0-ohm resistor wherever you want a via and not bother changing its parasitic values.

Add the Guard Trace to the Schematic

Let's ground our guard trace at its endpoints only and see how effective it is.

Add a "middle" trace to the existing schematic; ground it at both ends using 0-ohm resistors:

1. Click Close to close the oscilloscope.
2. In the schematic editor, add a transmission to the second row, between the aggressor and victim traces.
3. At either end of the new transmission line, add pull-down resistors.
4. Edit the resistor values to make them each 0 ohms.

Then add the new trace to the coupling region with the aggressor and victim traces:

1. Point to the new transmission line in the schematic, and right-click on it. The Edit Transmission Line dialog box opens. In the Coupled column, click the Stackup button. The Add to Coupling Region tab opens.
2. In the Coupling Regions list box, click once to highlight "Coupling0001". Then click the Edit Coupling Regions tab.
3. By default, LineSim adds new transmission to the *right side* of the coupling region. But we need the guard trace to be in the *middle* of the other traces. To move the guard trace, first verify that it is highlighted in the Coupling Region tree list (trace "TL(A1:B1)"), then click the left-arrow button below the graphical viewer once to "hop" the trace to the left, into the middle of the other two. Verify that "TL(A1:B1)" now appears in the middle of the other traces in the tree list above.
4. Click OK to close the dialog box.

Re-simulate to see if the crosstalk is reduced:

1. From the Scope/Sim menu, choose Run Scope. The Digital Oscilloscope dialog box opens. The worst-case crosstalk occurred previously on the rising edge, so verify that the Driver Waveform is to Rising Edge.

2. Click the Start Simulation button to run the simulator and view the waveform.

With the guard trace added, the crosstalk at the victim net's receiver end (blue waveform) is sharply reduced (to less than 200 mV). The guard trace appears to have worked well. (To see the crosstalk amplitude more easily, increase the oscilloscope's vertical scale and adjust the vertical position.)

However, when we added the guard trace, we were forced to separate the aggressor and victim traces by an additional 16 mils (8 mils for the guard trace itself, and 8 more for the extra trace-to-trace separation). This leads to an interesting question: how much of the crosstalk reduction was due to the presence of the grounded guard trace, and how much just to the fact the aggressor and victim traces are now further apart?

Remove the Guard Trace, but Leave the Aggressor and Victim Widely Separated

Let's try removing the guard trace, but leaving the aggressor and victim traces where they are, widely separated. This will tell us how much of the crosstalk reduction is due to simply moving the traces further apart, and how much due to the presence of the grounded guard trace.

Remove the guard trace; leave the aggressor and victim traces separated by 24 mils:

1. Click the Close button to close the oscilloscope.

2. In the schematic, point to the middle (guard) trace, and click twice with the left mouse button. The trace disappears. Point to the 0-ohm resistors, and click each one once; the resistors disappear.

3. Point to either the aggressor or victim trace, and right-click. The Edit Transmission Line dialog box opens. Click the Edit Coupling Regions tab.

4. By default, when you remove a trace from a coupling region, LineSim moves the remaining traces together. In this case, we don't want that. The total separation from aggressor trace to victim when the guard trace was present was 24 mils (edge-to-edge); re-establish that by typing into the Left

or Right Trace-to-Trace Separation edit box (whichever is not grayed out) “24”.

5. Click OK to close the dialog box.

Re-simulate to see if the crosstalk is reduced:

1. From the Scope/Sim menu, choose Run Scope. The Digital Oscilloscope dialog box opens.

2. Click the Start Simulation button to run the simulator and view the waveform.

The maximum crosstalk is about 480 mV. This is considerably reduced compared to the original 1.10 V, but not as small as when the guard trace was present. So some of the crosstalk reduction when the guard trace was present was due to the grounded trace itself and some to the fact that adding it caused the aggressor and victim traces to be separated further.

Note: *You can’t safely generalize from this example that, for example, about half the crosstalk reduction is always due to increased separation and half to the presence of the guard trace. These proportions are quite dependent on the exact details of the cross section, so you should be sure to simulate any specific cases that you’re interested in.*

But in situations where most of the crosstalk occurs because of trace separation, you may choose to omit the guard trace so that you don’t have to drop extra vias into your ground plane.

Chapter 6: Running Interactive Post-Layout Crosstalk Simulations *(BoardSim Crosstalk)*

Summary

Important! *This chapter is specific to the BoardSim Crosstalk product; it does not apply to LineSim Crosstalk. For detailed information about LineSim Crosstalk, see Chapters 2-5.*

This chapter describes:

- how to enable interactive crosstalk simulations in BoardSim Crosstalk
- how (and why) to switch back and forth between coupled and uncoupled simulations
- the concept of "victim" versus "aggressor" nets
- how BoardSim Crosstalk finds aggressor nets
- the advantages of electrical (rather than geometric) crosstalk thresholds
- how to set thresholds for finding aggressor nets
- how to view aggressor nets in the board viewer
- setting IC models for crosstalk simulations

- how to run simulations
- how to maximize simulation performance by setting the crosstalk threshold and/or limiting the number of aggressor nets

This manual assumes you that you are already familiar with how to operate the base BoardSim product, and describes only the extra steps required to run coupled simulations. If you need help with the features of base BoardSim, refer to the BoardSim User's Guide or the online Help.

Enabling Interactive Post-Layout Crosstalk Simulations

Enabling interactive crosstalk simulation in BoardSim Crosstalk is easy: with one mouse click, you can tell BoardSim Crosstalk to consider coupling between nets every time you simulate.

To enable interactive crosstalk simulation (if it is currently disabled):

1. From the Crosstalk menu, choose Enable Crosstalk Simulation.
   OR
   Click the Enable Crosstalk Analysis button on the toolbar.

Whenever crosstalk simulation is enabled, BoardSim Crosstalk will consider the coupling effects between neighboring nets *for all interactive simulations.* Coupling analysis is enabled differently for batch-mode simulations; see Chapter 8 for details.

The easiest way to remember whether coupled simulation is enabled or not is to look at the Enable Crosstalk button on the toolbar. If it is toggled "in," coupling will be included during all interactive simulations; if it is toggled "out," coupling is disabled and BoardSim Crosstalk will ignore coupling (just like base BoardSim does for all simulations).

Switching Between Coupled and Uncoupled Simulations

It is often useful when running BoardSim Crosstalk interactively to switch back and forth between coupled and uncoupled simulations. The are several reasons to do this:

- **Performance:** Coupled simulations often take much longer to run than uncoupled. For nets for which you are not concerned about crosstalk or coupling effects, why bother with the overhead of a coupled simulation?
- **Comparison of coupled versus non-coupled simulations, to isolate the effects of coupling**: Sometimes when you are viewing a complex and noisy waveform for a net, it is not easy to determine which effects are due to coupling and which are not. Repeating the simulation with coupling disabled can help you isolate the noise due specifically to coupling.

To switch back and forth between coupled and uncoupled simulations:

1. From the Crosstalk menu, choose Enable Crosstalk Simulation.
   OR
   Toggle the Enable Crosstalk Analysis button on the toolbar.

How BoardSim Crosstalk Finds Aggressor Nets

Suppose crosstalk simulation is enabled (see section “Enabling Interactive Post-Layout Crosstalk Simulations” above for details), and you now select a net for analysis. **BoardSim Crosstalk will automatically identify other nets which are coupled to the selected net, include them in simulation, and show them in the board viewer along with the selected net.**

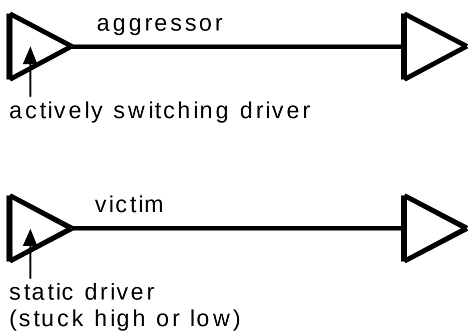
In a crosstalk simulation, the selected net is usually considered to be the “victim” net and the other coupled nets “aggressor nets.” The following section describes the difference between victims and aggressors.

“Aggressor” versus “Victim” Nets

In a crosstalk analysis, any PCB trace that is intentionally driven (usually by a switching IC output buffer) and is therefore a potential source of crosstalk on other traces is called an “aggressor.” Any trace that potentially receives unwanted crosstalk from an aggressor is called a “victim.”

Note that victim traces are *not* undriven. Rather, the victim trace is usually in a static state, “sitting high” or “sitting low” when a nearby aggressor trace is actively switched, and an unwanted signal appears on the victim. See Figure 6-1. Because of reflection effects, the state of the victim trace’s static driver is an important factor in the crosstalk waveforms that actually appear on the victim trace.

Figure 6-1: Aggressor and victim trace



Of course, it is possible to have a collection of traces (e.g., on a microprocessor bus) *all* of which are actively driven and *all* of which receive unwanted signal components from the other traces. In this situation, the distinction between aggressor and victim becomes blurred — each trace is both an aggressor and a victim. Usually for simulation purposes, though, you would make the victim net static, so that the crosstalk appearing on it is not mixed with a driving signal and single-line reflection effects.

Note that in differential signaling, if the differential pair is tightly coupled, then the two traces crosstalk with each other just like any two other coupled traces. However, it is not typical to use the terms “aggressor” or “victim” in a differential case, or even “crosstalk,” because the coupling is actually *wanted*. “Crosstalk” usually refers to *unwanted* coupling. When simulating differential pairs, you would normally drive both traces (rather than considering one trace to be a “victim” and sticking it high or low).

Electrical versus Geometric Identification of Aggressor Nets: Why Electrical is Superior

Most crosstalk-analysis tools allow you to set a geometric “zone” around victim nets; any other net which enters this zone for a sufficient length is considered to be an aggressor net and is included in simulation. For example, you might be able to specify a zone 40 mils wide and with a minimum length threshold of 250 mils, meaning that any net which comes within 40 mils (or less) of the victim net and stays that close for at least 250 mils is automatically assumed to be an aggressor net.

BoardSim Crosstalk supports geometric identification of aggressor nets, but also offers a better, “smarter” method, called an “electrical crosstalk threshold.” In this method (which the tool uses by default), probable aggressor nets are identified *electrically* by how many mV of crosstalk they might generate, regardless of where they are located.

Why are electrical thresholds superior to geometric? There are several reasons:

- Electrical thresholds are expressed electrically, the way an engineer thinks about crosstalk. For example, if you are concerned about any aggressor net that might generate more than 250 mV of crosstalk on a certain victim net, simply set the electrical threshold to 250 mV. BoardSim Crosstalk will attempt to find all nets that might cause 250 mV (or more) of crosstalk, and automatically include them in simulation.
- Geometric thresholds force you to have some knowledge (or to guess) about how much crosstalk can be generated by traces on a certain stackup layer a certain distance away from the victim trace. Suppose your noise budget allows for 300 mV of crosstalk on victim nets. How does that translate into a geometric setting? Should you set the threshold to 25 mils or 50 or 100 to account for 300 mV of crosstalk? BoardSim Crosstalk’s electrical thresholds eliminate having to make complex electrical-to-geometric conversions.
- Geometric crosstalk thresholds are valid for only one setting of a PCB’s other geometric factors. For example, suppose you finally get the geometric

threshold optimally set to pick up 300-mV-or-greater aggressor nets, then decide to experiment with a different dielectric thickness in your board's stackup. How should you now change the crosstalk threshold? Or what if you switch to faster driver ICs? BoardSim Crosstalk's electrical thresholds are independent of such changes: one value holds automatically for all stackups, driver-IC switching rates, etc.

- Geometric thresholds can sometimes be deceivingly optimistic. Suppose you set the threshold to 20 mils and perform your simulations assuming that your setting captures all of the significant crosstalk that can occur — and then find out later that some nets 40 mils away could also be significant aggressors? BoardSim Crosstalk's electrical thresholds are designed to be pessimistic: if you set your threshold to 200 mV, the tool attempts to bring in any net that could *possibly* generate that much crosstalk.

How to Set the Crosstalk Threshold

Whenever you enable interactive crosstalk simulation you should immediately check and set the crosstalk threshold (see section "Enabling Interactive Post-Layout Crosstalk Simulations" above for details). The threshold is used by BoardSim Crosstalk to automatically identify which other nets are coupled to the selected net. If the threshold is not set to match the amount of crosstalk you are concerned about, you could easily run your simulations with too many or too few aggressor nets.

Setting an Electrical Threshold

HyperLynx recommends using electrical crosstalk thresholds rather than geometric, unless you have a compelling reason not to. For a summary of why electrical thresholds are almost always superior to geometric, see section "Electrical versus Geometric Identification of Aggressor Nets: Why Electrical is Superior" above.

To set an electrical crosstalk threshold:

1. From the Crosstalk menu, choose Set Crosstalk Thresholds. The Set Crosstalk Thresholds dialog box opens.

2. Verify that the Use Electrical Thresholds radio button is selected (rather than Use Geometric Thresholds). Change if needed.

3. In the Include Nets with Coupled... edit box, type the desired threshold, in mV. For example, if you would like simulations to include any aggressor nets that could possibly generate more than 250 mV of crosstalk, type “250” into the box.

4. Optionally, also change the values of the Default IC Model. See section “The Default IC Model” below for details on what the default model means and how it is used.

5. Click OK.

The new threshold setting takes effect immediately. If you already had a net selected in the board viewer and just made a significant change to the threshold setting, you will likely see (after a brief pause) a change in the number of aggressor nets displayed. If you set the threshold lower (e.g., dropped it from 150 mV to 75 mV), more potential aggressor nets are likely to be found. If you set the threshold higher, fewer nets will be found.

The Default IC Model

When BoardSim Crosstalk examines a given net to determine if it is an aggressor to the selected net, the program considers a number of factors, one of which is the characteristics of the potential aggressor net’s driver IC. (Driver-IC switching time strongly affects the forward component of crosstalk.) If the net has a specific driver-IC model loaded onto it (e.g., from a .REF file or manually), that IC’s characteristics are used in the determination.

But if there are no models loaded on the potential aggressor net, then BoardSim Crosstalk uses the characteristics specified for the default IC model. Therefore, the details of the default model — especially the Rise/Fall Time — have a strong effect on how many aggressor nets are found for a given victim net. The model’s values default to aggressive but reasonable values. If you set the Rise/Fall Time to a small number (e.g., 100 ps), you may find very large numbers of aggressor nets being found for each victim net; this may be unrealistic and cause very long simulation times.

The following table shows in detail how and when the default IC model is used:

Case	**How the Default IC Model is Used**
Net has at least one output or I/O IC model loaded	IC model's characteristics are used; default IC model is ignored
Net has multiple output or I/O models loaded	Characteristics of the *fastest* IC model are used; default IC model is ignored
Net has one or more IC models loaded, but they are all input-only	Characteristics of the default IC model are used (no output or I/O IC model available)
Net has no IC models loaded	Characteristics of the default IC model are used (no output or I/O IC model available)

To set the characteristics of the default IC model:

1. If the Set Crosstalk Thresholds dialog box is not open, open it by choosing Set Crosstalk Thresholds from the Crosstalk menu.

2. Click the Change Default IC Model button. The Default IC Model Settings dialog box opens.

3. In the Rise/Fall Time edit box, type the value (in ns) of the time in which the default driver IC switches high and low (0%-100%, not 10%-90% or 20%-80%). If the rise and fall times differ, enter the faster of the two. If you're not sure what value to use, 1.0 ns is a reasonable guess for today's fast ICs.

   In the Output Impedance box, type the driving resistance of the default driver. If you're not sure of the value, 5 ohms is a reasonable guess.

   In the Input Capacitance box, type the input capacitance of the default model assuming it stopped driving and acted as a receiver. If you're not sure, 5 pF is a reasonable guess.

4. Click OK.

The Set Crosstalk Thresholds dialog box displays the value of the default IC model's rise/fall time. (This is the most-significant of the models' parameters for predicting crosstalk and finding aggressor nets.) To change the value, click the Change Default IC Model button.

Setting Geometric Thresholds

Again, HyperLynx recommends using electrical crosstalk thresholds rather than geometric (see section "Electrical versus Geometric Identification of Aggressor Nets: Why Electrical is Superior" above for details). However, you can switch to geometric thresholds if you wish.

Searching for aggressor nets geometrically requires two threshold values:

Geometric Threshold	**Description**
Minimum parallelism length	The minimum length over which a neighboring net must be parallel to the victim net before it can be considered an aggressor; the parallelism can come in multiple "sections," i.e., is cumulative
Maximum distance from victim	Defines a "zone" to either side of the victim net; if a neighboring net never comes at least this close to the victim, then the neighbor is not an aggressor

These two values are "ANDed," i.e., both must be satisfied in order for a neighboring net to be considered an aggressor net. The maximum-distance value is measured from trace edge to trace edge (not center-to-center).

Suppose the thresholds are set to 250 mils minimum parallelism and 40 mils maximum distance. Then:

- if a neighboring net ran alongside the victim net for two inches, but never came closer than 50 mils, it would *not* be considered an aggressor net

- if a neighboring net came as close to the victim net as 10 mils, but was closer than (or equal to) 40 mils for a total distance of only 200 mils, it would *not* be considered an aggressor net

- if a neighboring net came in four separate sections within 20 mils of the victim net, and each 20-mil pass was 75 mils long (for a total parallelism of 4x75 = 300 mils), the net would be considered a victim net

Note that an aggressor net's parallelism to the victim net is cumulative, i.e., each time the candidate aggressor comes within the "zone" set by the maximum distance threshold, the amount of parallelism increases. If the total parallelism inside the zone equals or exceeds the minimum parallelism threshold, the candidate becomes an aggressor.

Restoring Default Thresholds

At any time after making changes to the crosstalk thresholds (electrical or geometric), you can reset them back to BoardSim Crosstalk's default values.

To reset the crosstalk thresholds back to BoardSim Crosstalk's default values:

1. Open the Set Crosstalk Thresholds dialog box by choosing Set Crosstalk Thresholds from the Crosstalk menu.

2. In the dialog box, click the Restore Defaults button.

3. Click OK.

How Aggressor Nets are Found

Once you have enabled crosstalk simulation and set crosstalk thresholds (see the preceding sections in this chapter for details), BoardSim Crosstalk will constantly and automatically keep its list of aggressor nets up-to-date. Any change you make that affects crosstalk — to IC models, to the board stackup, to the list of power-supply nets, to trace widths, etc. — will cause the list of aggressors nets to be updated.

Victim Net is Treated as an Aggressor, Too

As BoardSim Crosstalk interactively constructs its list of aggressor nets, its most-basic task is to find which neighboring nets can cause more crosstalk on the selected net than is allowed by the threshold setting. This means that the selected net is inherently considered to be an victim net.

However, because in interactive simulation *you* control the IC models on every net — including the selected/victim net — you have the ability to run the victim net as though it were actually an aggressor. I.e., normally, if you consider the selected net to be a victim, you would set its driver-IC model to “Stuck High” or “Stuck Low” during simulation. However, there is nothing to prevent you from setting the model instead to “Output,” which would make the selected net as much an aggressor to the other aggressor nets as the others are aggressors to the selected net.

Accordingly, when BoardSim Crosstalk searches using an electrical threshold for aggressor nets, it first considers how other nets can aggress onto the selected net, then, in a second pass, how the selected net can aggress onto other nets. The complete set of nets found in these two searches constitutes the list of aggressor nets.

Algorithm for Finding Aggressor Nets

Finding aggressor nets geometrically is a straightforward problem. However, finding aggressor nets using an electrical threshold is a more-difficult problem, particularly considering that the underlying algorithm must run very quickly in order to be interactive.

HyperLynx’s algorithm for finding aggressor nets electrically is proprietary; we believe it to be one of BoardSim Crosstalk’s most-unique and -powerful features. It is based on the “weak coupling” theory of crosstalk, which (at the price of being somewhat approximate compared to a full treatment of crosstalk) yields a set of closed-form prediction equations that can be run real-time. The resulting capability is sufficient to make reasonable first-cut guesses as to how much crosstalk a given net can generate on another.

However, of necessity, any such approximation algorithm has limitations. For nets with “clean” linear routing, which run parallel to each other for a medium distance, the aggressor-finding algorithm is quite accurate.

As the routing topology becomes more complex, then the algorithm's results are more approximate.

HyperLynx has attempted to adjust its aggressor-net-finding algorithm to be conservative, i.e., the goal is to identify as an aggressor any net which *potentially* can generate as much or more crosstalk on the selected/victim net than specified by the threshold. However, there is no guarantee that a given aggressor will generate the amount predicted, and it is quite possible that it will generate less. (Or in some more-unusual cases, it may even generate more.) Only a detailed simulation (interactive or in batch mode) can determine the real amount.

Still , in spite of these limitations, the concept of electrical thresholds is very powerful. It allows you to screen for crosstalk effects in the electrical terms that are the natural language of digital design. It also generally does a good job of rapidly and automatically identifying important aggressor nets, and rarely omits a significant crosstalk contributor from simulation.

Aggressor Nets are Found on *All* Layers

Unlike some tools, which by default look only for aggressor nets on the same layer as the victim net, BoardSim Crosstalk looks at *all* layers when it searches for coupling regions and aggressor nets.

If your PCB has multiple signal layers that are not separated by plane layers (e.g., a microstrip and buried-microstrip layer), you'll likely see some inter-layer coupling if you run the coupling-region viewer (see Chapter 8, section "Walking the Coupling Regions Along a Victim Net" for details).

How Aggressor Nets are Displayed

If you have crosstalk simulation enabled (see section "Enabling Interactive Post-Layout Crosstalk Simulations" above for details), then each time you select a net for simulation, BoardSim Crosstalk automatically searches for all nets that are aggressors to the selected net, and displays them along with the selected net in the board viewer.

Remember that this process of finding aggressor nets is heavily dependent on how you have BoardSim's crosstalk thresholds set; low electrical thresholds will cause many aggressor nets to be found, and high thresholds fewer nets (see section "How to Set the Crosstalk Threshold" above for details). So after enabling crosstalk simulation and before simulating, be sure to set the crosstalk thresholds sensibly.

Distinguishing the Selected Net from Aggressor Nets in the Board Viewer

With crosstalk simulation enabled, the board viewer automatically shows both the selected net *and* its aggressors. The selected net looks just like it does in the base BoardSim product, i.e., it is drawn with solid, colored segments; the colors correspond to different stackup layers. Aggressor nets also display with their layer colors, but they appear "dashed" — or more precisely, have a dashed white line drawn along their centers.

Thus, it's fairly easy in the board viewer to distinguish between the selected and aggressor nets. Only the selected net is *not* dashed.

Another way to distinguish between the selected and aggressor nets is to toggle the toolbar's Enable Crosstalk button on and off several times. The aggressor nets will disappear and re-appear; the selected net will always display.

Identifying a Particular Net in the Board Viewer

Sometimes, especially when there are many aggressor nets displayed, you may want to identify particular nets in the board viewer. The can easily be done by pointing to the net in question with the mouse.

To identify particular nets in the board viewer using the mouse:

1. Point to the net which you wish to identify.

2. Look in the status bar at the bottom left of the BoardSim window. The name of the net you're pointing to is displayed.

This feature actually operates dynamically at all times in BoardSim, whether crosstalk simulation is enabled or not. (As you move the mouse around, it even identifies nets that are not presently visible in the board viewer.) If you point to a position at which there are multiple nets (on different stackup layers), the readout in the status bar lists all net names.

Setting IC Models for Crosstalk Simulations

How you choose IC models for the nets involved in a crosstalk simulation has a major impact on the simulation results. There are several reasons for this:

- First, what models are loaded strongly affects which aggressor nets are found for any selected victim net. Potential aggressors which have no models are assumed to behave as if driven by the default IC model (see section “The Default IC Model” above for details).

- Second, any aggressor net which enters the simulation (perhaps because the default IC model has a fast switching time) but actually has no driver IC loaded will not switch during simulation and therefore won’t contribute any crosstalk to the victim net. So it’s important that all significant aggressor nets have driver-IC models present.

Thus, setting up IC models is an important aspect of running crosstalk simulations. As usual in BoardSim, there are several ways to get models loaded, including from a .REF file or manually in the Assign Models dialog box. And multiple model styles (.MOD, .PML, and IBIS) can be used and mixed in any way desired. For details on loading and using IC models, see the BoardSim Users’ Guide.

Effect of IC Models on Pins in the Assign Models Dialog Box

In the base BoardSim product, when you open the Assign Models dialog box to manually assign IC models (or to check which models were automatically loaded), the Pins list box displays all of the component pins on the selected net and its associated nets. The contents of this list does not change as you add or change IC models. (If you are unfamiliar with the Assign Models dialog box or how to open it, see the BoardSim Users’ Guide)

In BoardSim Crosstalk, on the other hand, the Pins list box displays the component pins on the selected net, on its associated nets, *and* on all nets which are aggressors to the selected net. But the list of aggressor nets changes depending on which IC models are loaded (since the amount of crosstalk generated depends on the characteristics of the driving ICs). **Therefore, when crosstalk simulation is enabled, the Pins list in the Assign Models dialog box changes contents as you add, change, or remove IC models.**

The following table lists some of the effects you can see in the Assign Models Pin list as you make IC-model changes:

Action	**Example Effect**
Change an aggressor net's driver-IC model to a slower model:	Because the net is now driven with a slower switching edge, it will not generate as much crosstalk; if the crosstalk amount drops below the threshold, the net is no longer an aggressor and all of its pins disappear from the Pins list (including the one you just changed).
Remove an aggressor net's IC model:	Several effects are possible. Suppose the aggressor net has no other output or I/O models loaded; then the default IC model takes effect. If the default model is slower, then the aggressor net's pins disappear. If the default model is faster, the net remains an aggressor and no changes occur.
Change an IC model on the victim net to a slower model:	Remember that aggressor nets are judged not only by the effect they have on the victim net, but also by whether the selected/victim net can aggress onto them. Therefore, a slower driver-IC model on the victim net could cause one or more aggressor nets to disappear, and many pins in the Pins list box to vanish.

The key point is that with crosstalk simulation enabled, the contents of the Pins list box in Assign Models is dynamic. BoardSim Crosstalk will constantly update its list of aggressor nets in response to IC-

model changes you make. As you change IC models, the Pins list may grow or shrink.

Identifying Nets in the Assign Models Dialog Box

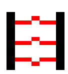
One difference between coupled and uncoupled simulations is that with crosstalk enabled, multiple nets are simulated simultaneously and the Pins list in the Assign Models dialog box tends to be long (compared to when crosstalk is disabled). This can make it difficult to know which pins belong to which nets, and whether a given pin is on an aggressor net or the selected/victim net.

Several features in the Assign Models dialog box assist with this problem (if you are unfamiliar with the dialog box or how to open it, see the BoardSim Users' Guide):

In the Assign Models dialog box, to tell whether a pin is on the selected/victim net or on an aggressor net:

1. In the Pins List, if a pin has a "coupling" icon (see picture below) to its immediate left, the pin belongs to an aggressor net. If the pin has no icon, it belongs to the selected/victim net.

The coupling icon looks like this (indicates an aggressor net):



In the Assign Models dialog box, to tell which net an IC pin is on:

1. In the Pins list box, click once on the IC pin to highlight it.
2. To the right of the list box, look for the Net field. It displays which net the pin is on; the net name appears in blue to make it easy to find.

Running Simulations

Once you have followed the steps described in the preceding sections of this chapter, running interactive crosstalk simulations is no different than running "ordinary" uncoupled simulations in the base BoardSim product.

As a reminder, the following “extra” steps are required to perform crosstalk simulations, compared to uncoupled simulations. For details on any of these topics, see the appropriate sections above in this chapter.

Summary — “Extra” steps before running interactive crosstalk simulations:

1. Enable crosstalk simulation.
2. Set the crosstalk threshold (preferably using the electrical threshold).
3. Optionally, change the characteristics of the default IC model.
4. Add IC models, if needed, to the selected/victim *and* aggressor nets.
5. Note the number of aggressor nets displayed in the board viewer; make adjustments to the crosstalk threshold, if desired (for more details, see section “How to Set the Crosstalk Threshold” above).

The only steps remaining are to apply oscilloscope probes and begin simulating.

Applying Oscilloscope Probes

Oscilloscope scope probes are applied for crosstalk simulations in the same way as for “ordinary” uncoupled simulations. The only difference is that, because there are typically multiple nets involved in a crosstalk simulation, there are usually many more pins available for probing (compared to uncoupled simulations).

Accordingly, you may need to plan your probing strategy more carefully for crosstalk simulations than for uncoupled. In particular, the auto-assignment feature (that places the oscilloscope probes on the first six available IC pins) will tend, for crosstalk simulations, to not place probes where you’d like them.

Normally, you’ll want to probe on the selected/victim net, to see what the crosstalk waveforms look like at various receiver pins. If you are also driving the selected net (so that it aggresses onto the “aggressor” nets), you may want to place some probes on the aggressors.

Running a Simulation

Simulation proceeds exactly in BoardSim Crosstalk as in the base BoardSim product. Once oscilloscope probes are applied, open the digital oscilloscope and click Start Simulation.

How to Maximize Simulation Performance

As noted previously (see Chapter 1, section “Minimum Computer Requirements for BoardSim Crosstalk”), crosstalk simulations are often much more CPU-intensive than uncoupled simulations. This is particularly true if the combination of your board’s geometry and stackup and your crosstalk-threshold settings result in a large number of aggressor nets being found for every selected net.

Although you generally do not want to miss the contribution of any net that could be an aggressor, there are diminishing returns associated with including large numbers of aggressor nets in every simulation. It is important to remember that BoardSim Crosstalk’s aggressor-net-finding algorithm is designed to be conservative: many nets that it selects will actually generate less crosstalk than expected.

Also, if you set your crosstalk thresholds to a reasonable level and are still finding large numbers of aggressor net in each simulation, you may have a serious board-wide crosstalk problem which is better addressed by globally changing your trace separations or PCB stackup than by trying to simulate and “tune” individual nets one-at-a-time.

Do Not Set the Crosstalk Threshold Unrealistically Low

One thing to avoid is setting the crosstalk threshold unrealistically low. For example, suppose you are worried about any crosstalk which exceeds 150 mV. There is no point in setting the threshold even lower — say, 50 mV — to try catching every possible simulation detail. This will only result in much slower simulations that add no information to your analysis results.

Even though BoardSim Crosstalk allows you to set the electrical crosstalk threshold as low as 10 mV, you should rarely (if ever) use such a low setting. Remember that there are many tolerances built-in to signal-integrity simulation: PCB manufacturing tolerances, IC-model approximations, and so forth. For many simulations, results in the 10—20-mV range may be down at the noise floor of these various tolerances.

Limiting the Number of Aggressor Nets

BoardSim Crosstalk has a user-settable limit on the maximum number of aggressor nets to include in a simulation. This limit works as follows:

- **if the limit is set to "N," then the N *strongest* aggressors are included in the simulation; other, weaker aggressors are omitted**

The concept of choosing the strongest aggressor nets is important. If a given crosstalk-threshold setting yields 30 possible aggressor nets, choosing the 12 strongest is quite reasonable; choosing 12 at random would be error-prone. Note that BoardSim Crosstalk's aggressor-net-finding algorithm is "smart" enough to know with good accuracy how aggressor nets rank against each other in terms of crosstalk-generating strength (see section "How Aggressor Nets are Found" above in this chapter for more details).

The limit is defaulted to a reasonable number that will almost always incorporate all of the significant aggressor nets into your simulations. Nevertheless, you can change the limiting value, if you wish.

To change the limit on the maximum number of aggressor nets to include in a simulation:

1. From the Options menu, choose Preferences. The Options dialog box opens.
2. Click on the BoardSim tab.
3. In the Crosstalk Options area, in the Maximum Number of Aggressor Nets data box, type the new value.
4. Click OK.

Note that the limit setting is not completely dynamic. If you change it with a net selected, the number of aggressor nets found will not change until you re-select the net or toggle crosstalk simulation off and back on.

Limit Applies to More than Simulation

The limit on the maximum number of aggressor nets to include in analysis applies to more than just simulation: it actually affects the entire BoardSim Crosstalk product, for consistency. For example, the limit applies in the board viewer, when the Terminator Wizard runs, etc. — any portion of the software which uses aggressor nets.

Each Aggressor Net and Its Associated Nets Count Only Once

Note that the number-of-aggressors maximum applies to *combinations* of aggressor nets and their associated nets. Therefore, the actual number of nets in the simulation may exceed the value of the limit. For example, if the limit is “12” and each aggressor net has one associated net, then there could validly be 25 nets in the simulation (the selected net + 12 aggressors + 12 nets associated with the aggressors).

Using the Aggressor-Net Limit to Improve Simulation Performance

The default value of the aggressor-net limit is fine for most designs and does not need to be changed. However, if you begin running interactive simulations and find them to be running very slowly, you may need to decrease the limit so that fewer aggressors are included and simulations run faster. Depending on the routing of your board and the amount of accuracy you need in your simulations, values as low as “4” or “5” may be reasonable.).

Determining How Many Aggressor Nets Have been Found

You can determine exactly how many aggressor nets have been found for a given selected net. The same feature also lists the names of each aggressor net (and its associated nets

To determine how many aggressor nets there are for the currently selected net:

1. From the Reports menu, choose Net Statistics. The Statistics for Selected Net dialog box opens.
2. After a pause for processing, the Associated Nets list box displays all nets associated with the currently selected net.

Aggressor nets which are in the list because of coupling are denoted with “by coupling” following their name. Nets in the list without this label are electrically associated with the selected net (i.e., connected directly to it through a component, e.g., a resistor).

Chapter 7: Generating a Crosstalk Strength Report *(BoardSim Crosstalk)*

Summary

Important! *This chapter is specific to the BoardSim Crosstalk product; it does not apply to LineSim Crosstalk. For detailed information about LineSim Crosstalk, see Chapters 2-5.*

This chapter describes:

- what the "Crosstalk Strength Report" is
- how IC models are included in the report's analysis
- how the "electrical threshold" is used
- how to generate a report
- how to interpret and use the report's results

The Crosstalk Strength Report

The Crosstalk Strength Report estimates, for each net on your PCB, the maximum crosstalk that could occur on the net. It is generated by BoardSim Crosstalk's batch-mode analysis engine. **The report's list is**

sorted by net from the one with the most possible crosstalk to the one with the least.

The Strength Report is beneficial for several key reasons:

- when you don't know which nets on your board are the ones most likely to experience crosstalk, the report gives you a powerful way to identify and focus on the nets that most likely need analyzing
- the report is generated very quickly, taking only a few minutes for even large PCBs
- for increased accuracy, the report uses whatever IC models you've loaded onto your board's nets, but the report can be generated even if no IC models are loaded (using a default-IC specification)

Whenever you have a PCB for which you don't know the nets that need crosstalk analysis, HyperLynx recommends generating a Crosstalk Strength Report. Performing crosstalk analysis (with HyperLynx or any other signal-integrity tool) can be time-consuming. Therefore, any information which helps you focus on the nets that need analysis (and skip the ones that don't) is valuable. The Strength Report is designed to give you exactly that information.

Furthermore, there is little reason *not* to create a Strength Report. Although the report is more accurate if you have all of your detailed IC models loaded, it can be created even with *no* models loaded. And BoardSim Crosstalk produces the report very quickly: even large PCBs take only minutes.

Numbers in Report are Only Estimates

It is important to remember that, as useful as it is, the data in Crosstalk Strength Report is only *estimated*. There will be situations (for particular nets) in which the estimated number is in error by a factor of 3 or 4 (and others in which it will much closer). The estimate is deliberately conservative; in most cases you will find that the actual crosstalk on a net is equal to or less than the estimated number. This conservatism attempts to ensure that the Strength Report does not miss any nets that might experience significant crosstalk.

Still, the only way to be sure of exactly how much crosstalk will be generated on a net is to perform a detailed simulation, either interactively or in batch mode. For details on interactive simulation, see Chapter 6; on detailed batch-mode simulation, see Chapter 9.

HyperLynx's algorithm for estimating crosstalk is proprietary. It is based on the "weak coupling" theory of crosstalk, which (at the price of being somewhat approximate compared to a full treatment of crosstalk) yields a set of closed-form prediction equations that can be run real-time. The resulting capability is sufficient to make reasonable first-cut guesses as to how much crosstalk each possible aggressor net can generate on a victim net. (The same technology is used to automatically identify aggressor nets; see Chapter 6, section "How Aggressor Nets are Found" for details.)

However, of necessity, any such approximation algorithm has limitations. For nets with "clean" linear routing, which run parallel to each other for a medium distance, the aggressor-finding algorithm is quite accurate. As the routing topology becomes more complex, then the algorithm's results are more approximate.

The Role of IC Models in the Strength Report

When BoardSim Crosstalk estimates crosstalk on a given net, it considers a number of factors, one of which is the characteristics if the driver ICs on aggressor nets. (For an explanation of what is meant by an "aggressor" net, see Chapter 6, section "'Aggressor' versus 'Victim' Nets.") In particular, driver-IC switching time strongly affects the forward component of crosstalk. Thus, if an aggressor net has a specific driver-IC model loaded onto it (e.g., from a .REF file or manually), that IC's characteristics are used in the crosstalk estimate.

The Default IC Model

However, because loading IC models can be time-consuming, you may sometimes want to generate a Strength Report before IC models are specified (or after only *some* are loaded). Accordingly, the report uses a default IC model in its estimates whenever "real" IC models are not available.

Of course, the report is most accurate if you take the time to specify IC models *before* generating it. (For details on the various ways to load IC models, see the BoardSim User's Guide.) But using the default IC model is often sufficient, especially since the report's estimates are only approximate anyhow.

Strategies for Using the Default IC Model

There are several possible strategies for using the default model:

- use only the default IC model when generating the report; this is the fastest approach, and it works especially well if most or all of the ICs on your board switch at approximately the same speed
- specify "real" models on certain critical nets for which you want more-accurate results or nets which have "special" drivers (e.g., ICs that switch faster than others on the board), and rely on the default IC model for all others
- specify "real" models for most or all of the ICs on the board, and use the default IC model for only a few exceptions; this is the most-accurate but also most time-consuming approach; for some boards, it may not be worth the effort

The following table shows exactly how the default IC model is used for a given aggressor net:

Case	**How the Default IC Model is Used**
Net has at least one output or I/O IC model loaded	IC model's characteristics are used; default IC model is ignored
Net has multiple output or I/O models loaded	Characteristics of the *fastest* IC model are used; default IC model is ignored
Net has one or more IC models loaded, but they are all input-only	Characteristics of the default IC model are used (no output or I/O IC model available)

Case	**How the Default IC Model is Used**
Net has no IC models loaded	Characteristics of the default IC model are used (no output or I/O IC model available)

The characteristics of the default IC model are specified during set-up prior to generating the Strength Report.

The Electrical Threshold

Another factor considered by BoardSim Crosstalk when it generates a Strength Report is an “electrical threshold” that you set. This section provides a concise explanation of how the threshold is used in the context of the Strength Report; for more details, see Chapter 6, section “How BoardSim Crosstalk Finds Aggressor Nets.”

Identifying Aggressor Nets

When BoardSim Crosstalk estimates crosstalk amounts, it must first determine, for each victim net on which it seeks an estimate, which *other* nets are aggressors to the victim. (For details on what is meant by “victim” and “aggressor net,” see Chapter 6, section “‘Aggressor’ versus ‘Victim’ Nets.”) Technically, every other net on a PCB couples to any given net, but from a practical viewpoint, only a small number of other nets couple strongly enough to generate any significant crosstalk.

Before it can create a Strength Report, BoardSim Crosstalk must know to what level of potential crosstalk generation you care about nearby, aggressor nets. E.g., are you only concerned about crosstalk that exceeds 300 mV, or is your noise budget tight and are you worried about any aggressor nets that could generate more than 40 mV of crosstalk?

This level of sensitivity is called the “electrical threshold.” In detailed interactive and batch-mode simulation, how you set this value has a strong affect on simulation performance. (For example, see Chapter 6, “How to Maximize Simulation Performance.”) For purposes of the Strength Report, this setting has only a small effect on performance but does determine how much data is listed in the report. In particular, the lower you set the threshold, the more aggressor nets will be listed for each victim net.

Specifically, if you set the threshold to “X” mV, then for every net listed in the Strength Report, only aggressor nets potentially contributing more than X mV of crosstalk to the listed net will be shown.

Threshold Also Used to Generate Warnings

The electrical threshold is also used in a second way in the Strength Report: as a warning threshold, such that any victim net whose summed crosstalk value (i.e., total contribution from strongest aggressor nets) exceeds the threshold is flagged with a warning in the report file. Thus, the threshold serves not only to determine which aggressor nets are considered for each victim, but also which victim nets are considered “in violation” in the Strength Report.

For more details on finding victim nets with threshold warnings in the report, see section “Interpreting and Using the Strength Report” below.

Recommended Threshold Value

Because for the Strength Report there is little performance degradation associated with a low threshold value (unlike for detailed simulations), it is reasonable to keep the value low. This guarantees that you’ll see almost all possible aggressor nets in the Strength Report. Therefore, BoardSim Crosstalk’s default value (which is set fairly low) is usually acceptable.

You might want to increase the threshold if you find that there is too much “clutter” in the Strength Report, i.e., if there are a large number of aggressor nets listed for each victim net, and you would prefer to have only the strongest ones listed. Or if too many nets are being flagged with violation warnings, you may wish to set a higher “standard” for such warnings.

On the other hand, you might want to decrease the threshold if you design has an unusually tight crosstalk-noise budget; or if your driver-IC signal swings are very low-voltage (e.g., less than 2.0V); or if your board is showing little crosstalk and you simply want to see more data in the report.

Generating a Strength Report

You can generate a Crosstalk Strength Report as soon as you have your board loaded into BoardSim. You may want to first specify some detailed IC models, however; see section "The Role of IC Models in the Strength Report" above for details.

Once you have specified IC models (if any), then generate the report as follows.

To generate a Crosstalk Strength Report:

1. From the Wizards menu, choose Board Wizard. The Board Wizard opens to its first page.

2. In the Quick Analysis area, click on the "Show Crosstalk Strength Estimates..." check box, to enable it. Also, click on any other check boxes that are enabled to disable them (unless you want to run other types of analysis simultaneously with generation of the Strength Report.) Disable all check boxes in the Detailed Simulations area.

3. Click the Next button. If you enabled only the Strength Report check box, the Wizard advances to the Batch-Mode Default IC Model Settings page.

4. In the Rise/Fall Time edit box, type the value (in ns) of the time in which the default driver IC switches high and low (0%-100%, not 10%-90% or 20%-80%). If the rise and fall times differ, enter the faster of the two. If you're not sure what value to use, 1.0 ns is a reasonable guess for today's fast ICs.

   In the Output Impedance box, type the driving resistance of the default driver. If you're not sure of the value, 5 ohms is a reasonable guess.

   In the Input Capacitance box, type the input capacitance of the default model assuming it stopped driving and acted as a receiver. If you're not sure, 5 pF is a reasonable guess.

5. Click Next. The Wizard advances to the Batch-Mode Strength/High-Accuracy Threshold page.

6. In the Electrical Threshold area, in the edit box, type the value of potential crosstalk below which you do not care about aggressor nets (see section “The Electrical Threshold” above for details).

7. In the “Show These Nets…” area, click the “All Nets” radio button if you want an entry in the report for every net on your board, or the “Only Nets Whose Crosstalk Exceeds…” button if you only want the report to list nets whose summed crosstalk exceeds the value in the Electrical Threshold box.

8. Click Next, then Finish. BoardSim Crosstalk runs, generating the Crosstalk Strength Report. Usually (unless your PCB is extremely large), the report is generated in a few minutes. When the report is completed, it opens automatically in the HyperLynx File Viewer.

Interpreting and Using the Strength Report

The Strength Report appears as a section in the Board Wizard Batch-Mode Report, which is always generated when the Board Wizard runs. If you enabled only the Strength Report feature when setting up the Wizard, then except for few short preliminary sections, the Strength Report will constitute the remainder of the file.

The report opens automatically in the HyperLynx File Viewer when the Wizard is finished generating it. You can view the report inside the HyperLynx Viewer, or in any text editor. The HyperLynx Viewer has the advantage of having a built-in search feature for easily finding nets that exceed the crosstalk threshold value.

If you want to view the Strength Report outside of the HyperLynx File Viewer, open the file <board_name>.RPT in your favorite text-file viewer (<board_name> is the name of your .HYP file). The report file is located in the same directory as the .HYP file..

Format of the Strength Report

Each net in the Crosstalk Strength Report has a section that looks like the following example (the details may have changed since this manual was published):

```
NET = datald
   ELECTRICALLY ASSOCIATED NETS ------------------------------
     net = datald
   AGGRESSOR NETS (Estimated peak crosstalk)
     en ............................  88 mv
     n00077 ........................  55 mv
     reset .........................  51 mv
     ld ............................  21 mv
     datald ........................  20 mv
     Sum of the two strongest aggressors ............. 143 mv
     ** Warning ** Estimate exceeds maximum allowed
crosstalk!
```

The net name comes first, followed by a list of the nets that are electrically connected to the net (i.e., its associated nets), then the list of aggressor nets to the named net, with an estimated crosstalk value for each aggressor individually. Finally, a summed crosstalk value is given, and then, if the summed value exceeds the value of the electrical threshold (see section "The Electrical Threshold" above for details), a warning is generated for the net.

Note: *Keep in mind that the crosstalk values shown in the Strength Report are estimates only. For details, see section "Numbers in Report are Only Estimates" above.*

Finding Warnings in the Report

If you are viewing the Strength Report in the HyperLynx File Viewer, finding nets in violation of the electrical threshold is easy.

To search for warnings in the Strength Report, using the HyperLynx File Viewer:

1. From the HyperLynx File Viewer's Search menu, choose Find Warning.
   OR
   Click the yellow Find Warning button on the Editor's toolbar.
   The editor jumps to the next warning below the cursor.

To search for the next warning in the report file:

1. From the HyperLynx File Viewer's Search menu, choose Find Next.
   OR
   Click the yellow Find Warning button on the Editor's toolbar.

OR
Press F3.

About the Summed Crosstalk Value

The “summed” crosstalk value in the report file is not the sum of the estimated crosstalk for each aggressor net. Often, because of the details of the PCB layout, how the aggressor signals are skewed in time, etc., the individual aggressor contributions do not add linearly. Instead, the actual achieved crosstalk value is less than the straight sum.

Statistically (by running experiments on a set of real boards), HyperLynx has determined that a useful way to create an estimated sum is to add the contributions of the *two strongest* aggressor nets. This method is employed in the Strength Report.

Note: *In future versions of BoardSim Crosstalk, as more statistical data is gathered, this algorithm may be refined further. The two-strongest-aggressor-nets method was used at the time of this writing.*

Nets are Sorted in Order of Most to Least Crosstalk

One of the most-useful attributes of the Strength Report is that it automatically sorts the nets it lists in order from the net with the most estimated crosstalk to the net with the least. This makes it very easy to see which nets on your board will probably have the most crosstalk.

Using the Strength Report

The Crosstalk Strength Report is intended primarily as a guide to which nets on your board are most likely to have crosstalk, and therefore which ones are the strongest candidates for further, detailed simulation. Even if the absolute crosstalk estimates in the report are not completely accurate (rarely will they be), the ranking of nets in the report is usually quite meaningful — i.e., the nets at the top of the list are indeed usually those with the highest crosstalk levels.

Again (see section “The Crosstalk Strength Report” above in this chapter), performing detailed crosstalk analysis with BoardSim (or

any other signal-integrity tool) is often time-consuming, so the sorted information in the Strength Report can be a powerful efficiency tool. Running interactive simulations on large numbers of nets without any information about which are the ones that really need to be analyzed can take many hours. So, too, can enabling the same large set of nets in batch mode (the run could take many extra hours).

On the other hand, choosing which nets to simulate judiciously, based on the information in the Strength Report, can gain back those same hours back, and make them available for other kinds of high-speed analysis, or other engineering tasks generally.

Chapter 8: Running the Field Solver in BoardSim *(BoardSim Crosstalk)*

Summary

Important! *This chapter is specific to the BoardSim Crosstalk product; it does not apply to LineSim Crosstalk. For detailed information about LineSim Crosstalk, see Chapters 2-5.*

This chapter describes:

- what a “field solver” is
- how BoardSim Crosstalk’s field solver works
- how BoardSim Crosstalk uses its field solver to calculate impedances, delays, and other electrical parameters of coupled transmission lines
- how to “walk” the coupling regions along a set of coupled nets
- how to view and save a coupling region’s electrical parameters

Note: *This chapter repeats some of the material in Chapter 3 regarding field solvers generally and how they work.*

About BoardSim Crosstalk and the Field Solver

In LineSim Crosstalk, the use of the field solver is very explicit: you define "coupling regions" in LineSim's schematic editor, and the field solver is invoked to find the coupled electrical characteristics of each region. (For details, see Chapter 3.)

In contrast, BoardSim Crosstalk's use of the field solver is much more hidden and automatic. When you select a net in BoardSim with crosstalk analysis enabled, the program automatically finds the nets to which the selected net is coupled (for details, see Chapter 6, section "How BoardSim Crosstalk Finds Aggressor Nets"). Furthermore, the program models in detail all of the coupling regions implied by the physical layout of your board. (A "coupling region" is just a cross section of some length which specifies geometrically how a set of traces are coupled to each other.) Then, before detailed simulation can be run, the field solver must be invoked to find the electrical characteristics of each of the regions.

The BoardSim Crosstalk product could have been constructed such that the entire process of finding and characterizing coupling regions is completely hidden. While you can run it that way (without bothering to look at any of the details), it is also possible to request information about which coupling regions have been identified for a particular selected net, and what the electrical characteristics of those regions are. This is strictly optional — there's no need to look at any of the details — but for users who want to see how BoardSim is choosing coupling regions or need information about coupled impedances, etc., the data is available.

This chapter describes how BoardSim Crosstalk's field solver works in general, and describes a feature called the "coupling region viewer" which lets you see the physical and electrical details of a selected net's coupling regions.

What is a "Field Solver"?

A field solver is a program that can solve for the electrical characteristics of a system of conductors and dielectrics, using one or more of the basic equations of electromagnetic theory ("Maxwell's equations"). Specifically, BoardSim Crosstalk uses its field solver to solve for the capacitances, inductances, propagation velocities, and characteristic impedances of the coupling regions it automatically detects when you select a net for analysis (for details on coupling regions, see Chapter 6, section “How BoardSim Crosstalk Finds Aggressor Nets”).

Because coupling regions consist of two-dimensional cross sections that are assumed to be constant over some specified length, BoardSim Crosstalk’s field solver needs to work in only two dimensions. Taking advantage of this fact allows BoardSim to calculate coupling parameters accurately, but also very quickly.

***Note**: Three-dimensional electromagnetic solutions become important only if the frequencies of the signals traveling on a system of conductors is so high that the wavelengths of the signals’ components are shorter than the various conductor structures in the system (e.g. vias, corner bends, etc.). This condition rarely occurs on PCBs carrying digital signals, so tools that analyze digital PCBs use two- rather than three-dimensional solvers. The big gain for users is speed: solvers run much faster in two dimensions than in three.*

When more than one transmission line is present in a coupling region, the various electrical parameters of the system take on a *matrix* form. For example, for a two-trace coupling region, there is no longer a single value of capacitance that describes the region’s cross section. Rather, there exists a 2x2 matrix which specifies both the capacitances of the individual traces to ground, and the capacitance between the traces.

The matrix nature of the electrical parameters describing a multi-trace coupling region is unfamiliar to many engineers and designers. For some detailed background information on coupled transmission lines and how they are described in matrix form, see Chapter 10.

How BoardSim Crosstalk’s Field Solver Works

***Note:** The information in this section is provided only for readers who are curious about what techniques BoardSim Crosstalk’s field solver uses to perform calculations. This material is not needed to successfully use BoardSim’s crosstalk-analysis features, and can readily be skipped.*

Calculation Details

After finding aggressor nets and identifying the coupling regions associated with, then in order to determine the electromagnetic properties of each region's cross section, BoardSim Crosstalk's field solver must calculate the capacitance and inductance matrices of each cross section. These matrices give the conductor-to-ground and conductor-to-conductor capacitances and the self and mutual inductances of the traces in the coupling region.

To calculate capacitance values, BoardSim Crosstalk's field solver finds the solution to Laplace's equation, a form of one of Maxwell's basic equations of electromagnetics:

$$\nabla^2 V = 0 \qquad (\textit{subject to all applicable boundary conditions})$$

In the solution, the solver seeks to find charge densities on the conductor surfaces and dielectric boundaries, rather than bothering to calculate the electric potential at all points between the conductors. **This approach makes BoardSim's field solver a "boundary-element" solver**. Several proprietary methods are used to speed calculations significantly while maintaining a high level of accuracy.

The solution to Laplace's equation occurs subject to all of the boundary conditions specified in the coupling region's cross section, i.e., it takes into account the exact shapes and locations of the conductors in the region and the locations and material properties of the dielectric boundaries. Special care is taken to calculate charge density accurately in regions in which it changes rapidly (e.g., at the corners of conductors).

Once the coupling region's capacitance values are found, then to calculate the inductance matrix, the field solver takes advantage of the following equation from transmission-line theory:

$$LC = \frac{\varepsilon}{c^2}, \quad \text{or} \quad L = \frac{\varepsilon}{Cc^2} = \frac{1}{C_0 c^2}$$

This allows a second solution to Laplace's equation — one in which all of the dielectrics are replaced by vacuum and the capacitance matrix C_0 is found — to substitute for an explicit calculation of the coupling region's magnetic properties.

Once the capacitance and inductance matrices are both known, then the region's propagation speed(s) and characteristic impedances can be calculated. For the case of inhomogeneous dielectrics (i.e., a mixture of dielectric constants, as occurs with microstrip and buried-microstrip traces), multiple propagation speeds exist. These speeds are found from the eigenvalues of the matrix product LC.

Field-Solver Cache

Since for each selected net and its aggressor nets, there are usually many physical coupling regions, BoardSim runs the field solver and repeats the steps described above *for each region.* Sometimes, there may be a hundred or more regions involved in a single selected net's analysis; the solver must run on all of these.

To speed the process, the field solver uses a smart cache which stores previous solutions; when the same cross section (or a geometric "reflection" of it) needs to be solved again, the answer is read from the cache rather than being recomputed. Usually, using this technology, BoardSim Crosstalk can solve all of the required cross sections for a given set of nets in a few seconds; however, very large or complex sets of nets may take longer.

If running the field solver is likely to introduce a noticeable delay, then a progress bar is displayed to show you the solver's status.

Note: *The field solver's cache file is stored in the same directory as the BoardSim Crosstalk executable (BSW.EXE). The name of the cache file is "FS_Cache.cah"; it is a binary file. If the file is deleted, BoardSim Crosstalk will start rebuilding it; you may notice a small slowdown in the program while the cache builds back up. The cache is flushed periodically to prevent the cache file from growing beyond several megabytes in size.*

What Information is Calculated

In BoardSim Crosstalk, the field solver's job is to calculate the following information for every coupling region found for the selected net and its aggressor nets:

- capacitance matrix
- inductance matrix
- characteristic-impedance matrix
- propagation speed(s)
- if multiple propagation speeds exist, the percentage of energy in each trace traveling at each speed
- an optimal resistor termination array for the region's transmission lines

(For background information on why many of these quantities are described in matrices, and what is meant by "multiple propagation speeds" and "optimal resistor termination array," see Chapter 10.)

Note that this information is calculated from the purely geometric and material data provided in your board's .HYP file and its stackup. Therefore, the field solver can be thought of as a calculation engine that transforms geometric/material data into corresponding electromagnetic data.

It is possible, if you want, to view the results of the field solver's calculations for a selected net's coupling regions. For details, see section "Viewing Coupling Regions and Field-Solver Data" below.

Viewing Coupling Regions and Field-Solver Data

BoardSim Crosstalk identifies aggressor nets, finds their coupling regions, and generates field solutions for each region automatically. There is absolutely no requirement for you to look at the regions that are found or the field solver's results for them.

Sometimes, though, you may want to know some details. For example, suppose you have one or more differential trace pairs on your PCB. You may be interested in knowing what trace-to-trace impedance the solver calculated for your pairs. Or you may be wondering where along a selected victim net the strongest coupling to nearby aggressor nets is, and so want to see the coupling regions found.

To allow coupling regions and field-solver results to be available for interested users (but also hidden from those who want to leave the process completely automated), BoardSim Crosstalk has a special dialog box called the "coupling-region viewer." It lets you "walk" along a selected net's physical coupling regions, and for each region, optionally look at the underlying electromagnetic data found by the solver.

Viewing Coupling Regions

Any time you have crosstalk enabled, a net selected, and the crosstalk threshold set such that at least one aggressor net is found, you can "walk" along the coupling regions that BoardSim Crosstalk has found. This section describes how to use the coupling-region viewer.

Opening the Coupling-Region Viewer

To view the coupling regions along the selected net:

1. From the Crosstalk menu, choose Walk Coupling Regions. The coupling-region viewer opens.

The viewer shows the details of one coupling region at a time along the selected net. (A "coupling region" is one coupling cross section encountered along the selected net; usually there are multiple such regions along the whole length of the net.) Each region consists of a cross section with specific electrical characteristics; two or more trace segments are involved (by definition, in order for there to be coupling).

Usually, these segments will be from different nets (one from the selected/victim net and others from various aggressor nets). However, since a net can couple to itself (e.g., in a “serpentine” route), it is possible to have a coupling region that consists entirely of segments from the same net.

The following sections describe the features of the coupling-region viewer.

Note: *The concept of a “coupling region” plays an important role in the LineSim Crosstalk product, where you manually construct such regions. For more information on coupling regions, see the LineSim Crosstalk user’s guide or Help.*

Moving from Region to Region

Since the coupling-region viewer displays the details of only one region at a time, it gives you a way to move from one region to the next. Each region has an arbitrarily assigned ID number; the number appears in the viewer’s title bar along with the total number of regions found along the selected net.

To move from one coupling region to the next:

1. In the lower left corner of the viewer, click the Next button. The viewer displays data for the next region; the ID number in the title bar increments.

If you repeatedly click the Next button, you move from region to region, eventually reaching the last region. If you click again, the viewer “rolls back” to the first region.

Note: *The coupling regions are ordered arbitrarily. There is no geometric significance to their order.*

You can also traverse the regions backward.

To move back to the previous coupling region:

1. In the lower left corner of the viewer, click the Back button. The viewer backs up to the previous region.

Seeing Regions in the Board Viewer

To make it easy to understand where in your actual PCB layout a given coupling region is, the region currently selected in the coupling-region viewer is also highlighted in the board viewer. The highlighting works by displaying the trace segments in the coupling region in white. Thus, as you move from region to region, you'll see sets of white lines moving in the board viewer.

The board viewer does not automatically pan to keep the current coupling region in view, so depending on the board viewer's current zoom level, you may need to pan it yourself. When the coupling-region viewer is open, most of BoardSim Crosstalk's functionality is grayed out, but the scroll bars and View menu (and associated toolbar buttons) can still be accessed. Thus, you can move and zoom in the board viewer as needed to see various coupling regions.

Note: *It is easiest to see coupling regions highlighted if you leave the board zoomed out fairly far (you'll have to do less panning) and the coupling-region viewer fairly small in size (it can be re-sized). See "Viewing Panes" and "Re-sizing the Viewer" below for more details.*

Viewing Panes

The coupling-region viewer displays three types of information:

- a summary of the nets in the currently displayed region, with their geometric relationships *("nets" information)*
- the electrical characteristics of the region, as generated by the field solver (including capacitance, inductance, impedance, etc.) *("impedance" information)*
- a graphical view of the region's cross section *("cross section" information)*

Each of these types of information has its own viewing pane. You can open or hide each pane, depending on whether or not you are interested in the data it displays. Closing one or more panes leaves more room for the others.

To close a pane:

1. At the bottom center of the coupling-region viewer, toggle off the button for the pane you want to close.

To re-open the pane, toggle its button on.

The Nets Pane

The Nets pane lists, for the currently selected coupling region, the following information:

- The names of the nets in the region. Usually each is different, unless a net couples to itself in this region (e.g., a serpentine-routed net)
- For each net listed, the separation distance from the currently selected net. If the nets are on the same layer, the distance is edge-to-edge; if they are on different layers, the distance is the vertical separation in the stackup. (For the selected net itself, no distance is shown.)
- For each net listed, its layer in the stackup. (Note: A given net may reside partially on several different layers. This value is the net's layer for *this* particular coupling region.)
- For each listed net, its width in this coupling region.

This information is helpful in understanding how each net contributes to the coupling in the region.

The Impedance Pane

This pane actually displays the electromagnetic data calculated by the field solver for the currently selected coupling region. For details about this information, see section "Viewing Field-Solver Data" below.

The Cross-Section Pane

This pane shows a graphical view of the coupling region's cross section. It essentially repeats the information in the "Nets" pane, but in graphical form.

You can easily identify each trace in the graphical viewer, by touching it with the mouse.

To identify a trace in the cross-section pane:

1. Point to the trace with the mouse. The trace's net name appears to the side of the mouse.

Sizing the Viewer

The viewer defaults to a small size, so that you can easily see the coupling regions highlighted in the board viewer as you walk amongst them. However, if you prefer, you can make the viewer larger to more easily see the information in it. (Don't forget that you can also close some of the panes in the viewer to make better use of its size; see the instructions under "To close a pane" above.)

To re-size the coupling-region viewer:

♦ Grab an edge of the viewer with the mouse, and drag it to the desired size.

Viewing Field-Solver Data

With the coupling-region viewer, BoardSim Crosstalk allows you to review in detail the coupling regions found for the selected net and its aggressor nets. In addition to showing you the physical contents of each region, the viewer also allows you to see the region's underlying electrical data as calculated by the field solver.

For details on opening the coupling-region viewer and using it to walk from region to region, see section "Viewing Coupling Regions" above. This section describes how to use the viewer's "impedance pane" to see the field solver's data.

The Impedance Pane

With the coupling-region viewer open and a coupling region selected, the viewer's Impedance pane shows you all of the data calculated by the field solver for the selected region. Specifically, the following information is available:

Information	**Description**
capacitance matrix	gives the self and mutual capacitances of the traces in the coupling region
inductance matrix	gives the self and mutual inductances of the traces
characteristic-impedance matrix	full matrix impedance for the system of coupled transmission lines; off-diagonal values are small for weak coupling and large for strong
optimal terminator-resistor array	an array of resistors (line-to-ground and line-to-line) that perfectly terminates the system of coupled lines; in theory, this array can completely eliminate crosstalk amongst the lines
list of propagation speeds	gives the velocities at which signals propagate on the traces; there multiple values if the traces' electromagnetic fields "see" more than one type of dielectric (e.g., microstrip or buried microstrip)
table of energy percentages in each propagation mode	gives the amount of each trace's energy that travels at each propagation velocity; *for multi-speed coupling regions only*
recommended termination values	list of impedance values, including the differential, common-mode, and line-to-ground values; *for two-trace coupling regions only*

The following sections describe how to generate and view this detailed information.

Details of the Field-Solver Information

The table above gives an overview of the data shown in the coupling-region viewer's Impedance pane. The following sections provide more details.

Physical Input Data

This section of the filed-solver data shows for what geometric and material information the field-solver results were calculated. The information serves as a record of the input problem, for future reference. The input data includes information on each trace in the coupling region as well as the board's stackup.

Correlating Transmission Lines and Matrix Indices

In the input data, the Field Solver Traces section lists by name each net in the coupling region, and shows the corresponding trace index by which the net's segment is referred to in the electrical matrix data elsewhere in the data. This data allows you to correlate nets and trace indices. See Figure 8-1.

Figure 8-1: Example of table correlating nets and trace indices

Field Solver Traces

	Matrix Index	Stackup Layer	X-Pos	Width	Comment
n00077	1:1	4	1600.00	10.00	
rsec0	2:2	4	1500.00	10.00	
n00074	3:3	4	1650.00	15.00	

Field-Solver Output Data

This section of the field-solver data shows the electrical data calculated by the field solver for the coupling region. As described earlier, much of this data is in matrix form, because the transmission lines in the region are coupled together. For background information on how electrical parameters are expressed for systems of coupled lines, see Chapter 10.

Optimal Terminator-Resistor Array

Sometimes, users of BoardSim Crosstalk's crosstalk-analysis features are interested in how to terminate traces that are coupled to other traces. This matrix gives the theoretically optimal resistor termination array for the set of coupled lines in the coupling region.

A key fact about coupled lines is that they cannot be perfectly terminated *individually*. Instead, a matrix of resistors that prescribes both line-to-ground *and line-to-line* resistances is required. (Again, for background information, see Chapter 10.) This termination array has the remarkable property that it not only "kills" single-line reflections at the line ends, but also eliminates arriving crosstalk signals.

On the other hand, there are many situations in digital electronics where line-to-line resistors (in addition to adding undesirably to passive-component count) are simply not permissible for DC-bias reasons. For example, whereas two coupled data lines may require a 160-ohm resistor between them to eliminate line-to-line crosstalk, it is unlikely that the driver ICs on the lines would be "happy" with the resistor when one line was pulled high and the other low.

Still, in some critical situations, especially when the line-to-line coupling is relatively weak and therefore the line-to-line terminating resistances are fairly high, a matrix terminator may be workable.

Note: *There are some IC technologies which are specifically designed to work with line-to-line termination: differential drivers. For these devices, line-to-line termination serves not only to prevent line reflections and eliminate crosstalk, but is often also required to bias the ICs for correct operation.*

To implement the termination described in the Optimal Terminator-Resistor Array matrix:

1. Place the resistors in the diagonal matrix positions between the corresponding trace to ground. (E.g., resistor 2-2 should be placed from trace 2 to ground, at the trace end.)

2. Place the resistors in the off-diagonal matrix positions line-to-line between the corresponding traces. (E.g., resistor 2-1 should be placed between traces 1 and 2, at the trace ends.)

Note that there are twice as many off-diagonal values as there are line-to-line resistors, since, for example, off-diagonal resistance 2-1 refers to the same resistor as resistance 1-2.

Note*:* *To correlate a specific net in a coupling region to a matrix index, see the field-solver data's Physical Input Data section, Field Solver Traces table.*

Characteristic-Impedance Matrix

This matrix gives the characteristic impedance (in ohms) of the system of coupled nets in the coupling region. As noted previously (and described in more detail in Chapter 10), coupled lines do not have a single-value impedance, like uncoupled lines. Rather, together, a set of coupled lines share an impedance matrix.

The values in the diagonal matrix positions can be thought of as giving the impedances to ground of the corresponding nets, accounting for the presence of the other nearby, coupled traces. When an IC drives into one of the lines, however, it “sees” not only the diagonal impedance for that line, but also some of the off-diagonal terms in the matrix.

For nets that are only weakly coupled, the diagonal impedance terms are dominant, and the diagonal values are close to what they would be if the lines were completely isolated from each other. As the coupling becomes stronger, the diagonal terms deviate more from their standalone values, and the off-diagonal terms increase. Note that small off-diagonal impedances mean weak coupling; large impedances mean strong coupling.

Barring special cases like two-line pairs in which the two signals are known to be either purely differential or purely common-mode, the diagonal impedances in the matrix are generally the best single-resistor terminators to use. Note, however, that coupled transmission lines cannot be perfectly terminated unless a full matrix termination (including both line-to-ground and line-to-line resistors) is employed. See “Optimal Terminator-Resistor Array” above for details.

Note: *To correlate a specific net in a coupling region to a matrix index, see the field-solver data’s Physical Input Data section, Field Solver Traces table.*

Capacitance Matrix

This matrix gives the self and mutual capacitances (in pF/m) of the coupled nets in the coupling region. More specifically, the diagonal values in the matrix give the capacitances to ground of the corresponding transmission lines, while the off-diagonal values give the capacitances between the corresponding pair of lines.

Many users are surprised to see that the off-diagonal capacitance-matrix values are negative. The negative sign simply reflects the fact that if a positive charge is placed on a given trace, negative charge will accumulate on all others. **For purposes of judging how much capacitance exists between traces, you can ignore the negative signs. The off-diagonal values do represent real, physical capacitance.**

However, in the mathematical formalism of coupled transmission lines, the negative signs are important. For example, if you transfer the capacitance matrix for a coupling region to another EDA tool (e.g., SPICE), the off-diagonal values **must** be negative.

Note that the values in the capacitance matrix have units of pF/m, rather than simply pF. This means that if you are trying to calculate, for example, the total capacitance-to-ground of a net in the matrix, you must multiply the corresponding diagonal value in the matrix by the length (in meters) of the net.

***Note**: To correlate a specific net in a coupling region to a matrix index, see the field-solver data's Physical Input Data section, Field Solver Traces table.*

Inductance Matrix

This matrix gives the self and mutual inductances (in nH/m) of the coupled nets in the coupling region. More specifically, the diagonal values in the matrix give the self inductances of the corresponding transmission lines, while the off-diagonal values give the mutual inductances of the corresponding pair of lines.

Note that the values in the inductance matrix have units of nH/m, rather than simply nH. This means that if you are trying to calculate, for example, the total self inductance of a net in the matrix, you must multiply the corresponding diagonal value in the matrix by the length (in meters) of the net.

***Note**: To correlate a specific net in a coupling region to a matrix index, see the field-solver data's Physical Input Data section, Field Solver Traces table.*

Propagation-Speeds List

This list gives the speed(s) (in m/s) at which signals propagate along the nets in the coupling region. As described in more detail in Chapter 10, coupling regions in which there is only dielectric (e.g., for stripline traces) have only one propagation speed; the signals on all traces in the region propagate with this single velocity.

However, coupling regions in which there are boundaries between dissimilar dielectrics (e.g., for microstrip or buried-microstrip traces) have multiple, discrete propagation speeds. Generally, each transmission line in the coupling region propagates some energy at each of the velocities prescribed by the region. There are as many speeds as there are transmission lines in the coupling region.

For most practical cross-section geometries, the multiple speeds are all close to each other. However, it is possible to encounter highly asymmetric cross sections in which the speeds are quite different. (An example of a “highly asymmetric” geometry would be a microstrip of one width coupled to a buried microstrip of a different width, with the buried trace below and considerably off to the side of the outer-layer trace.) This is an undesirable condition, however, because multiple, widely varying propagation speeds cause signal distortion, as one portion of the signal races ahead of the other(s).

For convenience, the propagation-speeds list displays velocities not only in m/s, but also as a fraction of the speed of light. For example, a value of “0.4822c” means 48.22% of the speed of light.

Note: *A misconception about propagation velocity on a transmission line is that electrons in the conductor are traveling along the line at the propagation velocity. This is absolutely not true! Electrons in a conductor spend almost all of their time randomly colliding with atoms in the conductor lattice; the mean time between collisions is on the order of 10 femtoseconds (1/100th of a ps).*
As a result, conduction electrons have only a relatively tiny average forward velocity in the presence of a driving voltage. A typical electron “drift velocity” in a conductor is on the order of 1 foot/hour. Instead, what moves at the transmission line’s propagation velocity is the electromagnetic wave that constitutes the actual signal on the line. Indeed, this wave is what you measure in the lab with an oscilloscope: a voltage waveform, which is really a measure of the electric field associated with the traveling electromagnetic wave.

Percentage of Energy Matrix (Multiple-Speed Coupling Regions Only)

If the coupling region supports multiple propagation speeds, this matrix gives, for each net in the region, the percentage of signal energy that travels at each speed. In the matrix, each column represents a line (i.e., a net's trace); reading down the column shows how much of the signal energy in that line travels in each of the propagation modes listed in the propagation-speeds list. The percentages in each column add to approximately 100%, to fully account for the energy on each net.

The values in this matrix are usually only of limited interest, unless the matrix shows a very uneven breakdown in energy sharing between propagation modes. For example, for certain highly asymmetric (and unusual) geometries, it is possible to have certain nets carrying most of their energy in one mode, while others carry a more even mixture of modes. If the velocities between modes differ significantly, this uneven distribution could lead to noticeable skew between signals on the nets.

Impedance and Termination Summary (Two-Line Coupling Regions Only)

For the special case of a two-line coupling region, the field-solver numerical results report gives additional information about specific termination options. The following table summarizes the additional data. **It is this section that reports the differential impedance of a pair of coupled traces.**

Termination Type	Description
differential	This is the proper line-to-line resistor to use if the two nets are being driven differentially, i.e., with equal-but-opposite signals. Will not terminate common-mode signals at all.
common-mode	This is the proper line-to-ground resistor to use for each line if the two nets are being driven identically, i.e., with equal signals of the same polarity. Not very useful for signals that sometimes switch together and sometimes oppositely, unless

Termination Type	**Description**
	crosstalk is primarily of concern when they switch together.
line-to-ground	This is the best line-to-ground resistor to use for each line if the signals on the transmission lines are completely unrelated. Will not perfectly terminate the line, but is a good single-component “compromise” value.
optimal termination	Describes the theoretically optimal resistor-array termination; consists of a line-to-line resistor plus two line-to-ground resistors. Same as the values given in the Optimal Terminator-Resistor Array matrix. Successfully terminates differential, common-mode, or mixed signals, but may violate DC-bias conditions on the lines.

Printing Field-Solver Information

The coupling-region viewer’s Impedance pane does not directly support printing. However, you can easily print the data by copying it from the pane into a Windows program that does print (e.g., Notepad, WordPad, Word, etc.)

To copy field-solver data into a program that can print it:

1. In the coupling-region viewer, in the Impedance pane, highlight the text that you want to copy.
2. On the keyboard, press Crtl-C. The text is now on the Windows Clipboard.
3. In the Windows program you plan to use for printing, press Ctrl-V. The text is pasted in.

Chapter 9: Running Batch-Mode Crosstalk Simulations *(BoardSim Crosstalk)*

Summary

Important! *This chapter is specific to the BoardSim Crosstalk product; it does not apply to LineSim Crosstalk. For detailed information about LineSim Crosstalk, see Chapters 2-5.*

This chapter describes:

- the difference between “victim” and “aggressor” nets, and how driver ICs behave during crosstalk simulation
- the different types of batch-mode crosstalk simulations that can be run in BoardSim Crosstalk
- running a Crosstalk Strength Report to decide which nets to simulate in detail
- how to set up IC models for batch-mode crosstalk simulations
- how to set up simulations in the Nets Spreadsheet
- how to export the Nets Spreadsheet to an external spreadsheet program (like Excel), and how to re-import it into BoardSim Crosstalk
- how to launch a batch-mode simulation
- how to interpret the results of simulations in the batch-mode report file

This manual assumes you that you are already familiar with how to operate the base BoardSim product, and describes only the extra steps required to run crosstalk simulations in the Board Wizard. If you need help with the features of base BoardSim, refer to the BoardSim User's Guide or the online Help.

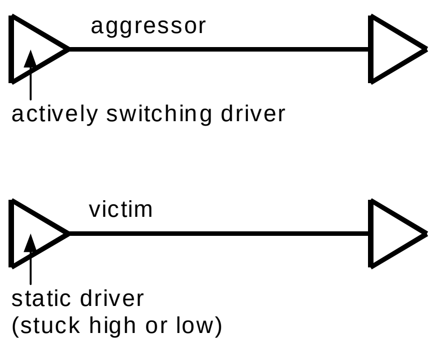
Behavior of Driver ICs During Crosstalk Simulations

"Aggressor" versus "Victim" Nets

In a crosstalk analysis, any PCB trace that is intentionally driven (usually by a switching IC output buffer) and is therefore a potential source of crosstalk on other traces is called an "aggressor." Any trace that potentially receives unwanted crosstalk from an aggressor is called a "victim."

Note that victim traces are *not* undriven. Rather, the victim trace is usually in a static state, "sitting high" or "sitting low" when a nearby aggressor trace is actively switched, and an unwanted signal appears on the victim. See Figure 9-1. Because of reflection effects, the state of the victim trace's static driver is an important factor in the crosstalk waveforms that actually appear on the victim trace.

Figure 9-1: Aggressor and victim trace



How Driver ICs Behave During Batch-Mode Crosstalk Simulations

When you run crosstalk simulations *interactively*, you can set your driver-IC models any way you wish using the Assign Models dialog box. Normally, the drivers on aggressor nets are switching and the one on the victim net is stuck either high or low.

But when you simulate in *batch mode*, since a potentially large number of nets are involved and since a stuck driver in one simulation may need to be switching in another, BoardSim Crosstalk must automatically set the state of driver ICs. It does this in the following way:

- for signal-integrity simulations, the driver IC on the victim net is automatically toggled; driver ICs on aggressor nets (if you choose the "high-accuracy" simulation mode in which coupled traces are included) are automatically stuck low
- for crosstalk simulations, the driver on the victim net is automatically stuck low or high (or both — your choice); driver ICs on aggressor nets are automatically toggled

For more details, see section "Types of Crosstalk Simulation in the Board Wizard" below.

You Must Identify Which ICs are Drivers

In order for BoardSim Crosstalk to know which ICs on a given net are the drivers, you must manually set each driver's buffer direction to "Output." The only exception is for IC pins which are modeled with IBIS or .PML models whose buffer direction is output-only; these will load automatically as outputs. However, all other pins must be explicitly set to direction "Output." This includes any pins with IBIS or .PML models of type "I/O" — unless you tell it explicitly, BoardSim Crosstalk does not know whether I/Os should drive or receive during simulation.

Actually, any "output-type" setting is acceptable for specifying driver direction: "Output," "Output Inverted," "Stuck High," or "Stuck Low." The batch-mode simulation engine will change these states from stuck to driving or vice versa as needed during simulation.

Note: *This restriction — that you must explicitly set driver ICs to buffer direction "Output" — will likely be removed in a future version of the product. If you have ideas about how you'd like BoardSim Crosstalk to treat the setup of driver ICs, please contact us at support@hyperlynx.com.*

Types of Crosstalk Simulation in the Board Wizard

BoardSim Crosstalk's batch-mode simulations are run using a feature called the "Board Wizard" (like all batch-mode features in BoardSim). The Board Wizard allows you to select from a list of several types of crosstalk simulation; you can run any or all of them during a given batch run. This section describes the simulation options. For how to actually enable any of these options, see section "Choosing the Simulation Type(s)" below.

Crosstalk Simulations

For crosstalk simulation, the intention of which is to see how much crosstalk will appear on a given victim net, BoardSim Crosstalk can run either or both of the following options:

- treat the selected net as a victim, stuck low; treat neighboring nets as aggressors, switching together first high, then low
- treat the selected net as a victim, stuck high; treat neighboring nets as aggressors, switching together first high, then low

The following section explains some of these options' details.

Selected Net is the Victim

For batch-mode simulation, the "selected nets" are the ones that you identify in the Nets Spreadsheet as needing simulation. (See section "Setting Up Simulations in the Nets Spreadsheet" for details on the Nets Spreadsheet.)

Each time the Board Wizard chooses one of these nets to simulate, the chosen net automatically becomes the victim, and neighboring nets that are coupled to it become aggressors. This means that the driver IC on the victim net will be stuck low or high, (i.e., will not switch), while driver ICs on aggressor nets will toggle high and low to generate crosstalk.

Importance of Victim Net's "Stuck" State

It is important that BoardSim Crosstalk includes non-switching (i.e., "stuck") driver ICs on victim nets, because on a real PCB, victim nets <u>do</u> have drivers. If the victim-net "stuck" drivers were omitted, the crosstalk results would be much different than if included. This occurs because driver ICs are typically low impedance and will reflect, rather than absorb, crosstalk signals.

As long as there is a driver IC identified on the victim net, the Board Wizard will automatically stick the IC in the proper low or high state. Note that you have a choice of simulating with the victim stuck low, stuck high, or both. If you choose both, twice as much simulation will be run (i.e., your batch run will take twice as long to finish).

It is your choice whether to simulate with victim-net driver ICs stuck in both directions, or only in one. If you choose only one of the two "stuck states," HyperLynx recommends choosing "stuck low." The reason is that for most driver ICs, the impedance of the low stage is lower than or equal to the impedance of the high stage, so that worst-case reflections of crosstalk signals come from the low stage. The reason to not simulate both edges is performance; simulating both will double the length of your batch runs.

Buffer Inversion Affects Stuck State

If a victim-net driver IC's buffer state is set to "Output Inverted" rather than "Output," then its polarity will be inverted relative to a "normal" driver: during a "stuck low" simulation, it will actually be stuck high, and vice versa. (This behavior is required when simulating differential IC outputs.)

Aggressor Nets Automatically Switch Both Low and High

For the victim net, you choose whether the driver IC sits low, high, or both (see preceding sections). However, the driver states on aggressor nets are automatically controlled by the Board Wizard: for every simulation, the aggressor-IC drivers are switched both low and high.

All Aggressor-Net Driver ICs Switch Together

If for a given simulation, multiple aggressor nets are present, BoardSim Crosstalk automatically switches all of the aggressor ICs together, simultaneously. First, all aggressor-net driver ICs switch low, then all aggressor drivers switch high. There is no skew between the aggressor-net switching times. **Using this method, the crosstalk that appears on the victim net is the worst-case, summed effect of all the aggressor nets acting together.**

Batch-Mode Crosstalk Simulation Uses Fast-Strong Drivers

Since some aspects of crosstalk worsen with faster driver-IC switching times, maximum crosstalk will almost always be produced by the fastest possible aggressor-net switching. Therefore, when it performs crosstalk simulations, the Board Wizard automatically sets aggressor-net ICs to their Fast-Strong state, to get the fastest possible edges.

(The forward component of crosstalk is roughly proportional in amplitude to driver slew rate, i.e., a faster driver will generate more forward crosstalk. For details on the Fast-Strong versus Typical versus Slow-Weak IC operating parameters, see the BoardSim User’s Guide.)

Batch Report Shows Worst-Case Crosstalk from All Simulations

Regardless of which crosstalk simulations you run (stuck low and stuck high, stuck low only, or stuck high only), the results reported for a given victim net in the batch-mode report are the worst-case crosstalk from any single simulation. For example, if you run stuck low and stuck high, there are actually four simulations that run:

- victim driver stuck low, aggressor drivers switching low
- victim driver stuck low, aggressor drivers switching high

- victim driver stuck high, aggressor drivers switching low
- victim driver stuck high, aggressor drivers switching high

For *each simulation*, the Board Wizard measures the maximum crosstalk that occurs on the victim net. Then in the batch-mode report file, the maximum of these measurements is reported, regardless of from which simulation it came. For more details, see section “Crosstalk Rule” below.

“High-Accuracy” Signal-Integrity Simulations

BoardSim Crosstalk can also perform a type of batch-mode simulation which takes advantage of its coupling-analysis capability, but is not, strictly speaking, crosstalk simulation. In this type of analysis, the Board Wizard runs signal-integrity simulation, but includes in the simulation all nets coupled to the selected net. This is in contrast to “regular” signal-integrity simulation, in which only the selected net (and “associated nets” connected to it by components) are included in simulation.

HyperLynx terms this kind of signal-integrity simulation “high-accuracy.” This name is used because, on boards where routing density is high (or net-to-net coupling is strong for other reasons), including neighboring, coupled nets in simulation may affect the results. These coupled simulations will generally be more accurate than simulations that exclude coupling.

High-accuracy simulation is an option to “regular” signal-integrity simulation. To run it, you must first select one or more of the basic signal-integrity options, then enable high-accuracy mode. For details, see section “Choosing the Simulation Type(s).”

Note: *The high-accuracy simulation option is available only with BoardSim Crosstalk, not the base BoardSim product. This is so because high-accuracy simulation requires all of the technology used in crosstalk simulation (field solver, aggressor-finding algorithm, coupled-mode simulator, etc.)*

When to Run High-Accuracy Simulation

Mandatory for Differential Signals (Unless Weakly Coupled)

Differential trace pairs, because they usually coupled significantly to each other, should almost always be simulated in high-accuracy mode. If you simulate a differential pair without enabling the high-accuracy option, each trace in the pair will be modeled only with its uncoupled impedance to ground; the presence of the other trace in the pair will not be accounted for.

The only exception is if you know (perhaps by using LineSim Crosstalk, or BoardSim Crosstalk's coupling-region viewer — see Chapter 8 for details) that your differential pairs are weakly coupled. Then you may feel comfortable skipping high-accuracy simulation. (The benefit of skipping high-accuracy is that simulations will run faster.)

Note: *Whether or not you choose high-accuracy simulation, both traces in your pair will be included together in simulation, as long as a differential IC model is used to drive them. This occurs because BoardSim considers the traces in a pair to be electrically associated with each other, i.e., recognition of coupling is not required to "draw" the second trace into simulation. However, if high-accuracy simulation is not enabled, the electromagnetic coupling between the pairs will be ignored.*

For Non-Differential Nets, Optional; Runs Slower but may Increase Accuracy

For non-differential traces, high-accuracy signal-integrity simulation is optional. For boards that are not dense (or that have been carefully designed to minimize crosstalk and coupling), the added accuracy benefit of high-accuracy mode may be quite small. On the other hand, for dense boards, or boards that are likely to suffer from a lot of coupling, high-accuracy mode may be worth running. In general, the trace impedances used in high-accuracy mode are more accurate (because they're modified by coupling) than those used in regular simulation.

The reason to not *always* run high-accuracy simulation is performance: coupled-mode simulation take much longer to run than uncoupled. Therefore, if you can “get away” with skipping high-accuracy simulation, you should do it.

Consider Running High-Accuracy Only on Differential Nets

A possible strategy is to break your batch-mode simulations into two groups: simulation of differential pairs (and/or other nets strongly affected by coupling), and simulation of non-differential nets. Then for batch run #1, enable high-accuracy simulation and use the Nets Spreadsheet to enable only the differential nets; for run #2, disable high-accuracy simulation and enable all non-differential nets.

How Driver ICs Behave During High-Accuracy Simulation

In high-accuracy simulations, driver ICs have the following behavior:

- on the victim net, the driver IC is toggled low and high
- on the “aggressor” net, driver ICs are stuck low (unless inverted; then stuck high)

Note that in this type of simulation, the term “aggressor” is really used to refer to neighboring nets that are coupled to the net being simulated. Strictly speaking, these nets aren’t actually aggressors, because their drivers are stuck during simulation, rather than switching (i.e., they can’t generate crosstalk). This has the end result of including the impedance effects of the neighboring traces in the simulation, but not the crosstalk-generating effects.

Running a Strength Report to Decide Which Nets to Simulate

Batch-mode simulation can be overwhelming (many IC models to set up and slow run times) if you always simulate *every* net on your PCBs. A better strategy is to judiciously choose a subset of nets that are likely to exhibit crosstalk, and concentrate on them. Or you may know that only a certain set of nets is sensitive to crosstalk, and choose to focus on them.

If you do not know which nets on your board are likely to exhibit crosstalk, an easy way to find out is to run a Crosstalk Strength Report. This analysis has several key advantages compared to detailed simulations:

- it runs even if no driver-IC models are loaded, by using a default driver-IC model (although where models *are* present, it uses their more-detailed information)
- it runs very quickly, in a matter of minutes on even large PCBs

HyperLynx strongly recommends running a Strength Report before setting up IC models and running detailed batch-mode simulations. Using the information in the report, you can intelligently choose which subset of nets to analyze in the Board Wizard. It may be that for some PCBs, you *will* need to analyze nearly the entire board (and BoardSim Crosstalk certainly allows you to do that), but if not, there's no reason to waste time on nets that are not critical or are likely to exhibit little or no crosstalk.

For complete details on the Crosstalk Strength Report, see Chapter 7.

Setting Up IC Models before Running the Board Wizard

The Board Wizard allows some types of "quick analysis" to be run without IC models loaded onto the nets it's analyzing (see the BoardSim User's Guide for details). However, for detailed simulations — crosstalk or signal-integrity — the presence of IC models is required. Therefore the first step in running detailed batch-mode crosstalk or signal-integrity simulation is setting up models.

Step #1: Loading IC Models onto Pins

The first step in setting up IC models is to load them onto the IC pins that will be involved in the simulations you plan to run. Note that if you intend to simulate only a subset of your board's nets using the Board Wizard, there's no need to load models onto all of the IC pins on your board: concentrate only on pins attached to the nets you'll be analyzing.

The easiest way to attach large numbers of models to pins is to use the .REF file (i.e., "IC AutoMapping"). This feature allows you to attach models to every pin on an IC with a single-line entry in the .REF file; BoardSim includes a smart editor that makes this mapping especially easy. For details on using .REF files, see the BoardSim User's Guide or online help.

Another way to attach models to pins is interactively, using the Assign Models dialog box (see the BoardSim User's Guide or online help for details). This method is especially appropriate if you are simulating only a small number of nets. You can mix the .REF-file and interactive methods; models specified manually in the Assign Models dialog box take precedence over assignments made in the .REF file.

In a Hurry?: Workaround for Quickly Specifying IC Models

If you're in a hurry to get simulation results and don't have time to find exact IC models, consider supplementing with models from library EASY.MOD. Or you can even run all of your analysis using EASY.MOD models. The results will not be as accurate as if you had taken the time to specify correct models, but the time-savings can be large and the results are usually at least fairly good.

If most or all of the ICs on your board switch at the same rate (i.e., with approximately the same switching time), you may be able to use one model for every IC. Applying the same model repeatedly to every IC reference designator on your board is easy using the .REF-file editor: choose the model in the Library and Component/Model combo boxes, highlight the first reference designator in the Reference Designator/Part Name list box, then click Paste Model(s) repeatedly.

Even if you model all ICs using one model from EASY.MOD, you must still specify buffer directions as described in the next section.

Step #2: Setting Driver-IC Buffer Directions

After IC models are loaded (see the preceding section), the second step in setting up IC models for batch-mode simulation is to specify which IC pins are drivers. Specifically, in order for BoardSim Crosstalk to know which ICs on a given net are the drivers, you must manually set each driver's buffer direction to "Output." The only exception is for IC pins which are modeled with IBIS or .PML models whose buffer direction is output-only; these will load automatically as outputs. However, all other pins must be explicitly set to direction "Output." This includes any pins with IBIS or .PML models of type "I/O" — unless you tell it explicitly, BoardSim Crosstalk does not know whether I/Os should drive or receive during simulation.

Actually, any "output-type" setting is acceptable for specifying driver direction: "Output," "Output Inverted," "Stuck High," or "Stuck Low." The Board Wizard will change these states from stuck to driving or vice versa as needed during simulation.

Buffer directions are set manually in the Assign Models dialog box. If you are unfamiliar with how to set directions or the operation of the dialog box, see the BoardSim User's Guide or online help.

Note: *This restriction — that you must explicitly set driver ICs to buffer direction "Output" — will likely be removed in a future version of the product. In particular, two helpful features are planned: "Auto Driver Direction," which will remove the requirement for driver direction to be specified by the user (BoardSim Crosstalk itself will "turn on" drivers); and a "round-robin" capability, which will cycle through multiple drivers one-at-a-time, if multiple drivers are found on a given net.*
If you have ideas about how you'd like BoardSim Crosstalk to treat the setup of driver ICs, please contact us at support@hyperlynx.com.

When Board Wizard simulates a net (and the nets coupled to it), and no driver ICs are present, simulations will usually stop and record an error in the batch-mode report file. In some cases (for example, aggressor nets which are missing driver-IC models), simulation will run, but the missing models will compromise the accuracy of the results. **Remember, the Board Wizard does not know which ICs on your nets you want to drive with (except for output-only**

models); you must tell it before starting batch-mode simulation. It is not enough to have a model loaded onto the net; the model's buffer direction must be set to "Output" (or "Output Inverted," "Stuck High," or "Stuck Low").

Rules for Setting Driver-IC Direction

The following table shows what happens for each type of batch-mode simulation if nets are missing driver ICs. The behavior is different depending on whether the model is missing from a victim or an aggressor net; victim nets can never be simulated unless a driver IC is present. The table also shows what happens if a net is found with multiple drivers.

Note: in the table, "SI" means "signal integrity." For details on "SI high-accuracy simulation," see section "'High-Accuracy' Signal-Integrity Simulations" above.

Condition	**Simulation Type**	**Result**
No models loaded onto victim net; *OR* models on victim net, but none with buffer direction "output"	SI *or* SI high-accuracy *or* crosstalk	Error in report file; no simulation results
No models loaded onto aggressor net(s); OR models on aggressor net(s), but none with buffer direction "output"	SI	OK — aggressors are not included in simulation anyhow
	SI high-accuracy	Simulation runs, but warnings in report file
	crosstalk	Warnings in report file; if no aggressor net has a driver, results are all 0/NA; if only some have drivers, crosstalk amounts may be too low

Condition	Simulation Type	Result
Multiple ICs with buffer direction "output" on victim net	SI *or* SI high-accuracy *or* crosstalk	Error in report file; no simulation results
Multiple ICs with buffer direction "output" per aggressor net(s)	SI	OK — aggressors are not included in simulation anyhow
	SI high-accuracy	Simulation runs with all aggressor drivers stuck low simultaneously; may result in driver conflicts
	crosstalk	Simulation runs with all aggressor drivers switching simultaneously; may result in driver conflicts

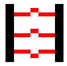
Identifying Nets in the Assign Models Dialog Box

When you are setting driver-IC buffer direction using the Assign Models dialog box, it's often useful to know which pins in the Pins list belong to the selected, victim net (or its associated nets) and which belong to aggressor nets. It's also useful to know exactly which net a specific pin is on. Several features in the Assign Models dialog box assist with this:

In the Assign Models dialog box, to tell whether a pin is on the selected/victim net or on an aggressor net:

1. In the Pins List, if a pin has a "coupling" icon (see picture below) to its immediate left, the pin belongs to an aggressor net. If the pin has no icon, it belongs to the selected/victim net.

The coupling icon looks like this (indicates an aggressor net):



In the Assign Models dialog box, to tell which net a pin is on:

1. In the Pins list box, click once on the pin to highlight it.

2. To the right of the list box, look for the Net field. It displays which net the pin is on; the net name appears in blue to make it easy to find.

Good Chance to Run Some Simulations Interactively

While you are specifying driver-IC buffer directions (“step #2” above), you may want to run some interactive crosstalk simulations, for example on certain critical nets. You can leave driver-IC directions in any of the “output-type” states (“Output,” “Output Inverted,” “Stuck High,” or “Stuck Low”) after interactively simulating; the Board Wizard will change them as needed during its simulations.

Set the Interactive Crosstalk Threshold before Specifying Models

As detailed in Chapter 6, which aggressor nets appear in the board viewer and Assign Models dialog box for any given selected net is determined heavily by the value of the crosstalk threshold. If you set the value high, you may see few or no aggressor nets; if you set it low, you may see many aggressors.

Because setting driver-IC buffer directions is a manual process that uses the Assign Models dialog box, you should set the *interactive* crosstalk threshold to the same value you plan to use for batch-mode simulation in the Board Wizard. (For details on setting the threshold for batch simulations, see section “Setting Electrical Threshold for Crosstalk Simulations.”) **If you plan to use several different thresholds during batch-mode simulation, then set the interactive threshold to the lowest of these levels.**

If you set the interactive threshold too high, then you won’t see some of the aggressor nets that will be used during batch-mode simulation, and therefore you won't be able to control the IC buffer directions on them. Then, during simulation, their contribution to victim-net crosstalk will not be included.

Only Electrical Thresholds Available in Batch Mode

In the Board Wizard, only electrical thresholds can be used to find aggressor nets. (In interactive mode, you can choose between electrical or geometric thresholds, although HyperLynx recommends electrical thresholds as a superior method.) **Therefore, when you set the interactive crosstalk threshold to match what you plan to use in batch mode, be sure to choose an electrical threshold rather than geometric.**

How to Set the Interactive Crosstalk Threshold

For a complete description of how to set the interactive crosstalk threshold, see Chapter 6, section "How to Set the Crosstalk Threshold."

Setting Up Simulations in the Nets Spreadsheet

Once you have run a Crosstalk Strength Report to help decide which nets to simulate in detail, and set up IC models (as described in the previous section), you are ready to prepare specific nets for simulation in the Board Wizard. The Board Wizard leads you through setup page-by-page.

Choosing the Simulation Type(s)

The Board Wizard offers several types of simulation (e.g., crosstalk, signal-integrity with coupling, etc.) For a complete description of these types, see section "Types of Crosstalk Simulation in the Board Wizard" above. The following steps assume you already know which type(s) you want to run.

To choose the simulation type(s) to run in the Board Wizard:

1. From the Wizards menu, choose Board Wizard. The Wizard opens on its initial page.

2. In the Detailed Simulations area, click on the Run Signal-Integrity and Crosstalk Simulations box to enable it. You can also enable other check boxes if you want to simultaneously run other types of analysis.

3. Click Next. The Batch-Mode SI and Crosstalk Options page appears.
4. If you want to run basic signal-integrity simulation (i.e., uncoupled simulation on the selected net), check any or all of the first three boxes in the Signal-Integrity Options area: Simulate Nets Using Fast-Strong; Simulate Nets Using Typical; and/or Simulate Nets Using Slow-Weak.

   If you want these simulation to run in high-accuracy mode (in which coupled nets are also included in the signal-integrity simulations), also check the Run at High Accuracy check box (below and to the right of the first three boxes).

   You can also set the Report Delays Relative to Driver check box as desired. Enabling the check box means that the “starting time” for each delay measurement is the time when the driver crosses Vmeasure; disabling the check box means measure all delays from t=0. (For more details, see the BoardSim User’s Guide or online help.)
5. If you want to run crosstalk simulations, check either or both of the boxes in the Crosstalk Options area: Simulate Selected Net as a Victim, Stuck Low; and/or Simulate Selected Net as a Victim, Stuck High.

Performance Status Bar

As you enable more and more simulation types, the resulting batch-mode simulation will get slower and slower. (This occurs because each simulation type requires one or more complete simulation runs per net.) Additionally, not every simulation type takes as long to run as other types: some are more mathematically complex than others, involve additional nets, require more simulation time to pass, etc.

For convenience, a “Fast-Slow” progress bar displays below the simulation-type check boxes. As you enable each simulation type, the bar moves progressively to the right to show relatively how much simulation time will be required per net. The closer to the “Slow” side the bar moves, the longer you can expect your batch run to take.

Limiting Per-Net Simulation Times

The simulation of individual nets can be time-consuming, especially if a given victim net pulls in large numbers of aggressor nets. (Setting the crosstalk threshold to a low value on a net will sometimes cause this to occur, especially on dense PCBs.)

In order to control the total amount of time spent in a batch run, you may want to place a time limit on the total simulation time spent per net. That way, if some net unexpectedly draws in a huge number of aggressor nets and is about to require 30 minutes to simulate (or, for example, if a power-supply net is incorrectly not identified and a given net is accidentally associated to large numbers of other nets), the Board Wizard will abandon that net's simulation after the specified time, and move on to the next net.

Your total Board Wizard run time is constrained basically by the number of nets you enable for simulation and the maximum simulation time you allow per net.

To set the per-net maximum simulation time:

1. In the SI Settings for Each Net area, in the Max Run Time edit box, type the value, in minutes, of the run-time limit you want to impose.

Selecting Individual Nets for Simulation

Selecting individual nets for simulation is performed in the Nets Spreadsheet.

To open the Nets Spreadsheet:

1. In the SI Settings for Each Net area, click Nets Spreadsheet. The spreadsheet opens.

The spreadsheet is sizeable. To make maximum use of space and see the most nets at a time, maximize the spreadsheet to be full-screen.

How the Spreadsheet Works

Net Names

In the left-hand column labeled “Net Name,” the Nets Spreadsheet lists every net on your PCB, *except* nets which you or BoardSim have identified as power supplies (for details on editing the list of power supplies, see the BoardSim User’s Guide or online help).

Sorting Net Names and Other Items

You can sort the contents of any column in the Nets Spreadsheet by clicking on the column’s header button, at the top of the column. Clicking once sorts in ascending order; clicking again sorts in descending order; clicking a third time returns to ascending; and so forth.

For example, to sort the list of net names from A-Z, click once on the Net Name button at the top of the left-hand column. To re-sort from Z-A, click the button again. Clicking a third time returns to A-Z sorting.

Net Statistics

Immediately adjacent to the Net Name column are two columns showing each net’s width and length. These are display-only columns, i.e., you cannot enter data in them. These columns are useful for sorting nets in an electrically meaningful manner, e.g., so that the longest nets appear at the top of the list.

Note: *For nets with multiple segment widths, the Width column shows the widest segment.*

Enabling Nets for Simulation

To enable a net for crosstalk or signal-integrity analysis (or both):

1. Click the check box in the net’s SI Enable column. Both crosstalk and signal-integrity analysis (”regular” or high-accuracy) are enabled with this single check box; which simulations run is determined by the simulation types you enabled earlier in the Wizard (see section “Choosing the Simulation Type(s)” above for details). A red check mark indicates that the net is selected and will be simulated in detail once the Board Wizard begins running.

To disable a net that has already been selected:

1. Click it's the net's SI Enable check box again. The red check mark disappears, and the net will not be simulated by the Board Wizard.

Note: *The abbreviation "SI" for "signal integrity" is used in several places in the Board Wizard.*

Enabling a Net Also Enables Associated Nets

If you enable a net for analysis that has one or more associated nets (see the BoardSim User's Guide for details on what is meant by the term), the associated nets will also be enabled at the same time. Similarly, if you disable a net, its associated nets are also disabled.

This behavior is required because associated nets are simulated together, as a group. Enabling or disabling one necessarily does the same to the others.

Viewing Which Nets are Enabled

If you enable a number of nets throughout the list for analysis, and want to see a summary of which ones you selected, click the button at the top of the SI Enable column. This will sort the nets to bring the selected ones to the top of the list.

Setting Net-by-Net Compliance Rules

What are "Compliance Rules"?

For every net it simulates, the Board Wizard gives two kinds of information:

- a tabular summary of the waveform that resulted, including information about pin-to-pin delays, overshoot, and crosstalk

- optionally, a warning if user-defined "compliance rules" (e.g., maximum allowed delay, overshoot, or crosstalk) are exceeded

If you were to run a batch-mode simulation without any rules set, you would get tabular data for each net simulated, but no warnings about compliance violations. The value of setting compliance rules is that it enables the Board

Wizard to highlight (with a warning) every net that exceeds the electrical limits you impose. When batch simulation is completed, you can scan the report file and quickly identify the "offending" nets, which you may then want to simulate further, try applying terminators to, etc.

When you first enable a net for analysis, its compliance rules are set to reasonable default values. If you want different values, you can change them net-by-net.

Note: *Actually, there is no way to completely "turn off" a net's rules, but you can make them unlikely to ever be violated by setting them to extreme values. See below for more details.*

Types of Compliance Rules

The Compliance Wizard supports the following types of compliance rules. Four of the five are signal-integrity-related; one is for crosstalk analysis.

Maximum rising-edge overshoot *(SI)*	For all receivers on the *selected* net, specifies the maximum voltage by which the signal can go *above* the final DC value
Maximum falling-edge overshoot *(SI)*	For all receivers on the *selected* net, specifies the maximum voltage by which the signal can go *below* the final DC value
Maximum pin delay *(SI)*	Specifies the maximum delay to any receiver on the *selected* net; measured at each receiver from the time the driver switches until the receiver's *farther* threshold is crossed for the *final* time
Minimum pin delay *(SI)*	Specifies the minimum delay to any receiver on the *selected* net; measured at each receiver from the time the driver switches until the receiver's *nearer* threshold is crossed for the *first* time
Maximum crosstalk *(crosstalk)*	For all receivers on the *selected/victim* net, specifies the maximum amount of peak crosstalk (positive or negative); measured relative to the DC level on the net

Crosstalk Rule

Only one of the compliance rules applies to crosstalk simulation: the maximum crosstalk value. During each net's crosstalk simulation (if you enabled crosstalk for a given net) the maximum crosstalk value is calculated as follows:

For the selected/victim net and its associated nets only (not for aggressor nets), find the maximum peak voltage excursion away from the net's DC voltage, positive or negative, at any receiver IC. The magnitude (i.e., absolute value) of this excursion is the maximum crosstalk. If both "stuck high" and "stuck low" simulations are run, take the maximum crosstalk that occurred in either simulation.

When the batch-mode report file is generated, this value is compared against the compliance rule you specified. If the actual simulation value exceeds the rule, the net is flagged with a crosstalk violation.

For example, if during a stuck-high simulation (in which the victim net sits at 3.3V at DC), the "noisiest" receiver on the victim net peaks up to 3.6V and down to 3.1V; and then during stuck-low simulation (in which the victim sits at 0.0V at DC), the noisiest receiver peaks up to 0.7V and down to -0.8V; then the maximum crosstalk occurred during the stuck-low simulation and has value 0.8V.

To enter a new crosstalk compliance rule for a net:

1. In the Crosstalk Max cell for the net, type the desired maximum value, in mV.

Specifying No Crosstalk Checking

On some nets, you may not care to set a crosstalk threshold. Instead, you may plan simply to view in the report file what crosstalk value occurs. (See section "Viewing the Board Wizard's Results" below for details on reading the batch-mode report file.)

To disable the crosstalk rule on a net, set its value to a large number that couldn't possibly occur, e.g., 20,000mV (= 20V).

To "turn off" crosstalk rule checking on a net:

1. Set the net's Crosstalk Max value to a very large number, e.g., 20,000 mV.

Crosstalk Rule Also Serves as Electrical Threshold

The crosstalk compliance rule for each net also serves a second purpose: it defines the electrical crosstalk threshold used for finding each net's aggressor nets. For more details, see section "Setting Electrical Threshold for Crosstalk Simulations" below.

Signal-Integrity Rules

The other (non-crosstalk) compliance rules in the Nets Spreadsheet are for signal-integrity simulations ("ordinary" or high-accuracy). For details on these rules, see the BoardSim User's Guide or online help.

Rules Apply Across All Simulations Run by the Board Wizard

All of the compliance rules you enter in the rules spreadsheet are automatically applied to all simulations run by the Board Wizard. For example, if you enable crosstalk simulation for the victim net stuck both low and high, Board Wizard will actually run four simulations (victim stuck low, aggressors switching high; victim stuck low, aggressors switching low; victim stuck high, aggressors switching high; victim stuck high, aggressors switching low). But the crosstalk compliance rule is checked against all of these simulations, and the worst-case crosstalk that occurs in any of them is checked against the rule.

All rules also apply for both the net for which they are entered and for all of the net's associated nets. A change made to the rules for any net in a group of associated nets changes the rules for the entire group. (For details on what is meant by "associated nets," see the BoardSim User's Guide or online help.)

Setting Values in an Entire Spreadsheet Column

Sometimes, when working in the Nets Spreadsheet, you will want to set every net's entry in a given column to the same value. For example, you may want to set every net's crosstalk threshold to 200 mV (a change from the default value), even if you don't intend to simulate every net.

To set the same value for every net in a spreadsheet column:

1. In the Nets Spreadsheet, click once on the heading button (the area that labels the column, at the top). This "selects" the column.

2. From the Column menu, choose Set Selected Column To. A dialog box opens.
3. In the dialog box, if the desired column value is numeric, type the new number; or if “binary” (i.e., true or false), click the appropriate radio button.
4. Click Apply. The dialog box closes and the column is filled with the new value. The value is displayed only for nets that are currently enabled for simulation.

Sometimes you may want to set an entire column back to its default value.

To reset every net in a spreadsheet column back to its default value:

1. In the Nets Spreadsheet, click once on the heading button (the area that labels the column, at the top). This “selects” the column.
2. From the Column menu, choose Set Selected Column To. A dialog box opens.
3. In the dialog box, click Apply Default. The dialog box closes and the column is filled with the default value. The value is displayed only for nets that are currently enabled for simulation.

There is also a way to set every entry in the *entire* spreadsheet (rather than just one column) back to default values.

To reset every entry in the entire spreadsheet back to default values (rather than just one column):

1. From the Sheet menu, choose Set Entire Sheet to Default.

All cells in the spreadsheet are set to their default values.

Setting Electrical Threshold for Crosstalk Simulations

Whenever BoardSim Crosstalk runs a crosstalk simulation (or any simulation involving coupled nets), it must judge which other nets are coupled to the selected/victim net. This process is described thoroughly in Chapter 6, section “How BoardSim Crosstalk Finds Aggressor Nets”; see it for details.

In the Board Wizard, the threshold for finding aggressor nets is always electrical, i.e., you specify to what mV level you're concerned about coupling, and the Wizard finds an appropriate set of aggressor net for each victim net it simulates.

When you enable crosstalk simulations in the Board Wizard, the crosstalk threshold for each net is specified by the crosstalk compliance rule you set for the net (i.e., the Crosstalk Max column in the Nets Spreadsheet). This means that the crosstalk compliance rule serves a dual purpose:

- it specifies at what amount of crosstalk on the victim net to generate a warning in the batch-mode report file
- it specifies the electrical crosstalk threshold (used for finding aggressor nets) for the victim net

This means that you can set different threshold levels individually for each net on which you enable crosstalk. However, often, for simplicity, you'll use the same level for all nets.

Editing the Nets Spreadsheet in an External Spreadsheet (e.g., Excel)

While the Board Wizard's Nets Spreadsheet has most of the features needed to comfortably edit and manage the settings for a board, you can optionally export its data to an external spreadsheet (like Microsoft Excel), edit it there, and re-import into the Board Wizard. You might want to do this, for example, to take advantage of certain features (like advanced sorting) the are not available in the Wizard's spreadsheet.

The Nets Spreadsheet's data is passed to external programs in the form of an ASCII .CSV (comma-separated values) file. Almost all spreadsheet programs can read .CSV files.

Exporting the Nets Spreadsheet to an External Program

To export the Nets Spreadsheet to an external spreadsheet:

1. In the Board Wizard's Nets Spreadsheet, from the Import/Export menu choose Export Sheet (Save as .CSV). A dialog box opens.
2. In the dialog box, type the name of the .CSV file into which you want to save the spreadsheet's data.
3. Click Save. The .CSV file is created.

To edit the data in an external spreadsheet program:

1. In the external spreadsheet program, import the .CSV file generated in by the previous steps. (See the program's documentation for details.)
2. Once the .CSV file is loaded, edit its contents. "Enable" columns in the data (e.g., the SI Enable column) use "0" to represent "disabled" and "1" for "enabled."
3. When you are finished editing, save the file back out in .CSV format.

Since many HyperLynx users own Microsoft Excel or Lotus 123 specifically, there is a short-cut for exporting the Nets Spreadsheet into Excel or 123.

To export the Nets Spreadsheet into Microsoft Excel or Lotus 123:

1. In the Board Wizard's Nets Spreadsheet, from the Import/Export menu choose Export Sheet (as .CSV) and Launch in Excel/123. A dialog box opens.
2. In the dialog box, type the name of the .CSV file into which you want to save the spreadsheet's data.
3. Click Save. The .CSV file is created, and Excel or 123 is automatically opened on the file.

If for some reason this automatic launching does not work on your PC, open the .CSV file manually in Excel or 123

Importing the Edited Data Back into BoardSim

Once you have edited the spreadsheet data in the external program and saved it back out into a .CSV file, you can import it back into BoardSim, for saving or further editing. .

To import the .CSV file saved from the external spreadsheet back into BoardSim:

1. In the Board Wizard's Nets Spreadsheet, from the Import/Export menu choose Import Sheet. A dialog box opens.

2. In the dialog box, type the name of the .CSV file you saved from the external spreadsheet. Click Open; the edited data appears in the Nets Spreadsheet.

Note: *An annoying feature of Microsoft Excel is that it always prompts to save a file into a fixed directory, rather than the directory from which you opened the file. Therefore, be careful to save the edited .CSV file back into your .HYP-file directory, if that's where you want it to be.*

Other Portions of the Board Wizard (see BoardSim User's Guide)

There are many other aspects to the Board Wizard, beyond those described here that relate to running detailed crosstalk and high-accuracy signal-integrity simulations. For details on other Board Wizard features, see the BoardSim User's Guide or the online Help.

Saving Compliance Rules

The Nets Spreadsheet saves your settings when you close it, unless you specifically tell it not to save them.

To close the Nets Spreadsheet and save changes to settings:

1. From the File menu in the spreadsheet, choose Close.
   OR

Click the close button in the upper-right corner of the spreadsheet window. A dialog box asks if you want to save your changes; click Yes.

To close the Nets Spreadsheet *without* saving changes to settings:

1. From the File menu in the spreadsheet, choose Cancel.
   OR
   Click the close button in the upper-right corner of the spreadsheet window. A dialog box asks if you want to save your changes; click No.

Finishing Net-by-Net Enabling and Rules

Once you are satisfied that the nets you wish the Board Wizard to analyze are all enabled and have appropriate rules entered, you can move ahead in the Board Wizard. If you later decide to modify or add to what you've entered, you can back up in the Wizard (or reopen it, if already closed) and makes changes.

To finish enabling nets and setting net-by-net compliance rules:

1. Once you have enabled all desired nets for analysis and entered net-by-net compliance rules, click the Next button. The Board Wizard moves to the next setup page.

Setting the Default IC Model

As described in previous sections, IC models must be in place (and driver buffer directions set) to get valid, accurate simulations. However, in some cases it is still possible to simulate even when not all IC models are loaded. This is particularly true of ICs missing from aggressor nets (missing ICs on victim nets cause simulations to fail completely). (For details, see section “Rules for Setting Driver-IC Direction” above.)

If the Board Wizard encounters nets which do not have IC models, it can sometimes use a “default” IC model to make judgments about whether the net is coupled to the selected net, etc. The characteristics of the default model are specified on the Batch-Mode Default IC Model Settings page.

For more information about how the default IC model is used by BoardSim Crosstalk to find coupled nets, see Chapter 6, section “The Default IC Model.”

Important! The default IC characteristics you specify in the Board Wizard are completely separate from those specified interactively, in the Crosstalk menu's Crosstalk Thresholds dialog box. However, for simplicity, you'll usually want to set the model's characteristics the same in both places.

To set the characteristics of the default IC model for batch mode:

1. From the Batch-Mode Default IC Model Settings page in the Board Wizard, run the following steps:

   In the Rise/Fall Time edit box, type the value (in ns) of the time in which the default driver IC switches high and low (0%-100%, not 10%-90% or 20%-80%). If the rise and fall times differ, enter the faster of the two. If you're not sure what value to use, 1.0 ns is a reasonable guess for today's fast ICs.

   In the Output Impedance box, type the driving resistance of the default driver. If you're not sure of the value, 5 ohms is a reasonable guess.

   In the Input Capacitance box, type the input capacitance of the default model assuming it stopped driving and acted as a receiver. If you're not sure, 5 pF is a reasonable guess.

Setting the Electrical Threshold for High-Accuracy Signal-Integrity Simulations

For crosstalk simulations, the electrical threshold for finding coupled, neighboring nets is set individually for each net, and is equal to crosstalk compliance rule value (Crosstalk Max column value) in the Nets Spreadsheet. See section “Setting Electrical Threshold for Crosstalk Simulations” above for details.

If you have enabled high-accuracy signal-integrity simulations, the Board Wizard must also find coupled nets, similar to how it does for crosstalk simulations. However, a single, global electrical threshold is used for these simulations (in contrast to crosstalk simulations). This threshold is specified in the Wizard on the Batch-Mode Strength/High-Accuracy Threshold page. (For details on how coupled nets are found and what is meant by the “electrical threshold,” see Chapter 6, section “How BoardSim Crosstalk Finds Aggressor Nets.”)

To set the global electrical threshold for use during high-accuracy signal-integrity simulations:

1. From the Batch-Mode Strength/High-Accuracy Threshold page in the Board Wizard, in the Include Nets with Coupled Voltages Greater Than edit box, type the value of threshold.

Running the Board Wizard

Check Power Supplies Before Running

IMPORTANT! Before running the Board Wizard, you must check to see that all power-supply nets are correctly identified to BoardSim Crosstalk. (For details on editing the power-supplies list, see the BoardSim User’s Guide or online help.) If a power-supply net that connects to many other nets through components like resistors and capacitors is left unidentified, the Board Wizard will run for a very long time when it analyzes the net (the Wizard may appear to be “hung”). This is especially true if crosstalk or high-accuracy signal-integrity simulations are enabled.

Running the Wizard

To run the Board Wizard:

1. Run the preliminary steps in the Wizard to choose simulation types, enable specific nets for simulation, and set other parameters. See the preceding sections for detailed instructions.

2. Click ahead in the Wizard until the final, Batch-Mode Analysis page appears. Then click Finish. The Board Wizard begins running on your board, showing its progress as it runs. For large boards, the Wizard may take some time to run. If crosstalk and/or high-accuracy signal-integrity simulations are running, the Wizard may take many hours to complete.

3. When the batch run is completed, it automatically opens its report file in the HyperLynx Report File Viewer for viewing. If the report is very large, it may take a while to load even after the Wizard itself has stopped running.

Stopping the Board Wizard

You can force the Board Wizard to stop running before it has completed analysis of your entire board (e.g., if you remember that you set something up incorrectly, or if the Wizard is taking longer to run than you expected).

To stop the Board Wizard before it has completed its analysis:

1. Click the Cancel button. A message appears briefly, saying that the Wizard will stop after it finishes analysis of the present net.

When the Wizard has finished with the present net, it stops and opens its report for viewing. A warning message is written at the bottom of the report file, noting that you stopped the Board Wizard's analysis early. For details on the report, see below.

Viewing the Board Wizard's Results

When you run the Board Wizard, it writes its results into an ASCII report file called <*HYP_file_name*>.RPT, where <*HYP_file_name*> is the name of your board's .HYP file. When the Wizard has finished running and the report file is complete, BoardSim Crosstalk automatically opens the report for viewing in the HyperLynx Report File Viewer.

Changing the Name of the Report File

You can change the default name of the report file, if you want.

To change the name of the report file:

1. On the Batch-Mode Analysis page in the Board Wizard, just before clicking the Finish button to begin running the Wizard, type a different name in the Report File Name edit box.

Format of the Wizard's Report

The Wizard's report file (.RPT) is formatted to be easy to read and understand. It is divided into sections; a number of the sections relate to non-coupled simulations and are described in the BoardSim User's Guide (or online Help).

Results Table for Each Net

For crosstalk and signal-integrity simulations, the report file contains a concise table summarizing the simulation data for each net that was simulated. The table shows the following information:

"Nets coupled during crosstalk simulation"	If crosstalk simulations run, lists the aggressor nets that were included in simulation
"Nets coupled during high-accuracy simulation"	If high-accuracy signal-integrity simulations run, lists the coupled nets that were included in simulation
Device.Pin	A list of all the IC pins (driver and receiver) on the selected net and its associated nets; pins are named as<reference designator>.<pin name>
Dir	The pin's direction; "out" means the pin is driving, "in" means receiving; the suffix "df" is added to designate differential pins
Delay Rise	Minimum and maximum delays to each pin on the net, for a rising-edge transition of the driver (for details on how delays are calculated, see the BoardSim User's Guide)
Delay Fall	Same as Delay Rise, except data is for a falling-edge transition of the driver
Overshoot	The maximum overshoot, in the rising and falling directions, beyond the final DC value (for details on how overshoot is calculated, see the BoardSim User's Guide)

Crosstalk	The maximum crosstalk that occurred on the selected/victim net during any simulation in which the aggressor-net driver ICs toggled high and low; measured relative to the “stuck” victim net’s DC voltage
ERROR FLAGS	A section summarizing any signal-integrity violations, versus the compliance rules you entered, for the net and its associated nets; see description below and the legend at the top of the report file for details on interpreting this section
elapsed simulation time	A record of how many nanoseconds of simulation time was spent on analysis; extremely long times (100’s of ns) may indicate an IC-model or similar problem
max. rising overshoot allowed	The compliance rule you entered for maximum allowed overshoot on the rising edge
max. falling overshoot allowed	The compliance rule you entered for maximum allowed overshoot on the falling edge
min. delay allowed	The compliance rule you entered for minimum allowed pin delay, for rising and falling edges
max. delay allowed	The compliance rule you entered for maximum allowed pin delay, for rising and falling edges
max peak crosstalk allowed	The compliance rule you entered for the maximum allowed crosstalk; a peak value measured relative to the “stuck” victim net’s DC voltage

Searching in the Report for Crosstalk Violations

If a net (and its associated nets) violates any of the compliance rules you specified (see section “Setting Net-by-Net Compliance Rules” above for details on setting rules), a warning is issued in the net’s report table. There are two classes of warnings in the file; signal-integrity and crosstalk violations are considered “severe” warnings. XE "warnings, in batch-mode report" \t "*See* batch-mode crosstalk simulation, finding compliance warnings"

To search for warnings in the Board Wizard’s report file, using the Report File Viewer:

1. From the Report File Viewer’s Search menu, choose Find Warning.
   OR
   Click the yellow Find Warning button on the Viewer’s toolbar.
   The editor jumps to the next warning below the cursor.

To search for the next warning in the report file:

1. From the Report File Viewer’s Search menu, choose Find Next.
   OR
   Click the yellow Find Warning button on the Viewer’s toolbar.
   OR
   Press F3.

To search for only “severe warnings”:

1. From the Report File Viewer’s Search menu, choose Find Warning Severe.
   OR
   Click the red Find Warning Severe button on the Viewer’s toolbar.

Interpreting Violations

For nets that have crosstalk (or signal-integrity) violations, the ERROR FLAGS section of the net’s results table summarizes what kind(s) of violation(s) occurred. Each IC pin on the net has a violations line of its own; rising-edge and falling-edge violations are reported separately.

Types of Violations

The Board Wizard checks for the following kinds of crosstalk and signal-integrity violations:

delay violation	receiver-IC pin's maximum delay is longer than the net's compliance rule *OR* receiver-IC pin's minimum delay is shorter than the net's compliance rule (not checked for driver-IC pins)
threshold error	IC pin's signal level never reached the switching threshold
overshoot violation	IC pin's signal level exceeded the final DC level by more than the net's overshoot threshold
multi-threshold-crossing error	receiver-IC pin's signal level crossed the Vih or Vil threshold more than once during transition (not checked for driver-IC pins)
crosstalk violation	victim-net IC pin's peak crosstalk level exceeded the net's compliance rule

Note that three of these violations are checked against user-entered compliance rules (delay, overshoot, and crosstalk), while two are automatically checked against IC-model threshold values (threshold and multi-threshold crossing). Also, three (threshold, overshoot, and crosstalk) are checked at drivers and receivers, whereas two (delay and multi-threshold-crossing) are checked only at receivers.

For 3-state drivers that are 3-stated (i.e., set to be "off" rather than driving), only overshoot violations are checked for.

How Violations are Reported

The ERROR FLAGS section for each net shows which (if any) of these violations occurred during the Board Wizard's analysis. There is one group of flags for the rising switching edge and one for falling. If a violation occurs, a "code letter" is printed to the appropriate column; otherwise, the columns are marked with a hyphen ('—').

The following table shows the violation codes and what they mean (this information is also summarized at the top of the report file):

D	delay violation
T	threshold error
O	overshoot violation
M	multi-threshold-crossing error
X	crosstalk violation
—	no error

When looking at a net’s report data, you can quickly scan the ERROR FLAGS table and look for any column entries other than’—‘. Any letters in a column indicate a violation. Check the detailed numerical information to the right to see the details of the violation, e.g., by how much a delay was too long, etc.

Also, if any violation occurs, then the message ** Warning(Severe) ** is printed into the net’s section in the report. You can search on this message to quickly find nets with violations (see “Searching in the Report for Crosstalk Violations” above for the easiest ways to search).

About Negative Delays

If you are calculating delays driver-relative, it is possible for some receiver delays to be negative. This can occur especially at unterminated trace ends where the high impedance causes a doubling effect at the receiver and may cause it to cross its threshold before the driver crosses its threshold. (See the BoardSim User’s Guide for more details.)

Opening an Existing .RPT File

To view an existing Board Wizard report using the Report File Viewer:

1. From the Wizards menu, choose Board Wizard.
   OR

Click the Run Board Wizard button on the toolbar.
The Board Wizard dialog box opens.

2. In the View Previous Board Report area, click Open. The Report File Viewer is launched.

The Open button automatically causes BoardSim to search for a file named <*HYP_file_name*>.RPT, where <*HYP_file_name*> is the name of the currently loaded .HYP file.

Chapter 10: Technical Background on Crosstalk, Coupled Transmission Lines, and Differential Signals

Summary

This chapter gives some technical background on the subject of crosstalk, coupled transmission lines, and differential signals, specifically:

- what crosstalk is and what causes it
- the difference between forward and backward crosstalk
- the electrical parameters that describe coupled transmission lines
- differences between coupled striplines and microstrips (propagation modes)
- differential signals, and differential and common-mode impedance
- how coupled transmission lines can be terminated

What is Crosstalk and What Causes It?

In the simplest terms, crosstalk is unwanted coupling of voltages and currents between neighboring conductors. On a PCB, the conductors are usually traces, although crosstalk can also occur in connectors, cables, and component packages. In the classic crosstalk scenario, when a signal is intentionally driven on one conductor, an unwanted signal also appears on a neighboring conductor — even though there is no conductive connection between the driven conductor and its neighbor.

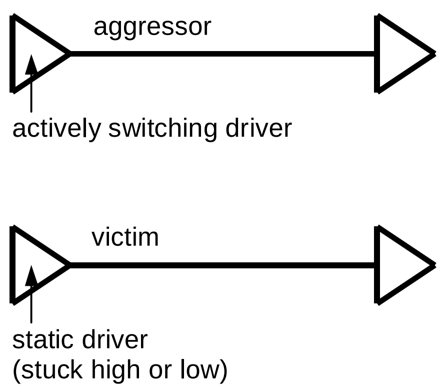
In the remainder of this chapter, conductors will usually be referred to as "traces," even though crosstalk can occur between any types of conductors.

"Aggressor" versus "Victim" Traces

In a crosstalk scenario, any trace that is intentionally driven (usually by a switching IC output buffer) and is therefore a potential source of crosstalk on other traces is called an "aggressor." Any trace that potentially receives unwanted crosstalk from an aggressor is called a "victim."

Note that victim traces are *not* undriven. Rather, the victim trace is usually in a static state, "sitting high" or "sitting low" when a nearby aggressor trace is actively switched, and an unwanted signal appears on the victim. See Figure 10-1. Because of reflection effects, the state of the victim trace's static driver is an important factor in the crosstalk waveforms that actually appear on the victim trace; see section "Reflection of Backward Crosstalk from Victim Driver IC" later in this chapter for details.

Figure 10-1: Aggressor and victim trace



Of course, it is possible to have a collection of traces (e.g., on a microprocessor bus) *all* of which are actively driven and *all* of which receive unwanted signal components from the other traces. In this situation, the distinction between aggressor and victim becomes blurred — each trace is both an aggressor and a victim.

Note that in differential signaling, if the differential pair is tightly coupled, then the two traces crosstalk with each other just like any two other coupled traces. However, it is not typical to use the terms "aggressor" or "victim" in a differential case, or even "crosstalk," because the coupled signals are actually *wanted*. "Crosstalk" usually refers to *unwanted* coupling.

What Causes Crosstalk?

When a signal travels down a trace, it is an electromagnetic wave that is propagating along the trace, from the driver end toward the trace's far end. At points along the trace which the wave has already reached, transient voltages appear and currents flow, in response to the wave's presence.

Note: *A misconception about propagation on a transmission line is that electrons in the conductor are traveling along the line at the propagation velocity. This is absolutely not true! Electrons in a conductor spend almost all of their time randomly colliding with atoms in the conductor lattice; the mean time between collisions is on the order of 10 femtoseconds (1/100th of a ps).As a result, conduction electrons have only a relatively tiny average forward velocity in the presence of a driving voltage. A typical electron "drift velocity" in a conductor is on the order of 1 foot/hour. Instead, what moves at the transmission line's propagation velocity is the electromagnetic wave that constitutes the actual signal on the line. Indeed, this wave is what you measure in the lab with an oscilloscope: a voltage waveform, which is really a measure of the electric field associated with the traveling electromagnetic wave.*

Electromagnetic waves consist of time-changing electric *and* magnetic fields. The fields are not confined to the inside of the trace that carries them — in fact, just the opposite: the fields' energy exists very predominantly *outside* the trace.

Therefore, as a signal propagates along one trace on a PCB, if there are other traces in the vicinity, they "see" the propagating signal's electric and magnetic fields. But according to Maxwell's equations (which define the behavior of all electromagnetic phenomena, except at atomic distance scales), time-changing fields induce voltages and currents in conductors — and thus the fields created by the propagation of a signal along one trace cause signals to appear on other nearby traces. This is crosstalk.

Forward and Backward Crosstalk

As a signal propagates along an aggressor trace, it causes an unwanted crosstalk signal to appear on a nearby victim trace (see "What is Crosstalk and What Causes It?" above for more details.) But exactly what kind of signal appears on the victim trace?

The crosstalk signal on a victim trace can be divided into two components: a *forward* signal and a *backward* signal. The following sections describe how these components are created, and what their properties are.

The "Speedboat Analogy"

The mathematics that governs crosstalk is somewhat complex. However, it's possible to describe a few of the qualitative features of forward and backward crosstalk using a physical analogy: a speedboat traveling across a lake.

When a speedboat travels through water, it disturbs the water in two ways. First, the boat tends to build up a "pile" of water in front of it; this "bow wake" travels along with the boat. Second, the boat leaves another wake behind it; this wake stretches out for a long distance behind the boat. See Figure 10-2.

Roughly speaking, the same thing happens on a victim net when a signal travels along the aggressor net. Two crosstalk components develop: a forward signal, which travels on the victim net just "in front" of the aggressor signal, and looks like a "piled-up" voltage; and a backward signal, which trails out behind the aggressor signal and stretches out in time. See Figure 10-3; compare to Figure 10-2.

Note: *Admittedly, this analogy can't be pushed very far before it breaks down. For example, the "bow wake" in front of a speedboat consists of water which is "raised up" by the boat, but flows around the moving bow — it doesn't really* *travel with the boat, as a forward crosstalk signal does with its aggressor signal. Also, when a boat travels fast enough, i.e., begins "planing," the bow wake becomes very small — there is no such analogy with crosstalk (unfortunately).*

Similarly, the backward wake left by a boat spreads in width, but doesn't really travel backward, like a backward crosstalk signal does. Also, a speedboat's backward wake is normally much larger in amplitude than its bow wake — not true for crosstalk.

Nevertheless, this analogy is a useful way to remember that forward crosstalk signals "pile up" and travel along with the aggressor signal, and that backward signals stretch out in time behind the signal.

Figure 10-2: Speedboat analogy for crosstalk signals

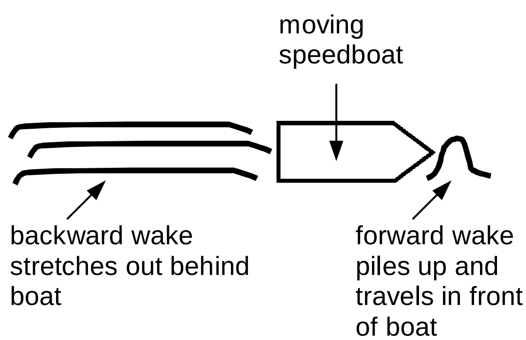
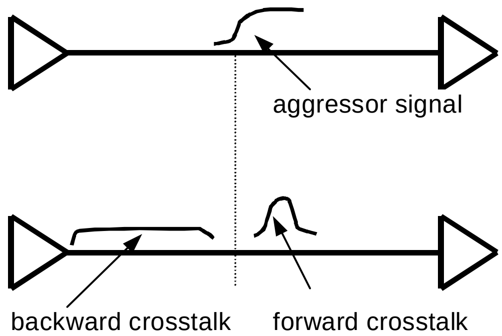


Figure 10-3: Forward and backward crosstalk signals



Details of Forward Crosstalk

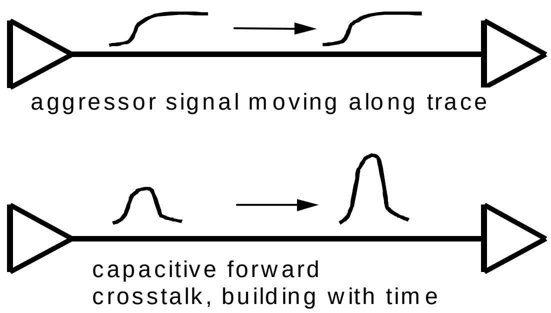
The “speedboat analogy” of the preceding section suggests only the crude details of how a forward-crosstalk signal behaves: that it travels along “in front” of the aggressor signal that creates it, and that it tends to be short and “piled up” rather than long and stretched out. This section gives more detail.

Forward crosstalk appears as a result of two *competing* coupling mechanisms, one capacitive and one inductive.

Capacitive Forward Crosstalk

As an aggressor signal travels down its trace, its time-varying *electric* field tends to generate on the victim trace a signal whose voltage polarity is the same as the aggressor’s, e.g., a rising edge on the aggressor trace creates a positive pulse on the victim trace. This crosstalk pulse travels ahead on the victim trace at the same speed (or set of speeds; see “Propagation Modes — Single-Dielectric versus Layered-Dielectric Traces” below in this chapter for details) as the aggressor signal travels on its trace. Therefore, the pulse doesn’t spread out in time, but rather keeps getting added to as the aggressor signal travels along and couples more and more energy onto the victim trace. See Figure 10-4.

Figure 10-4: Capacitive portion of forward crosstalk signal; same polarity as aggressor signal



Notice in Figure 10-4 that the forward crosstalk pulse is not a one-way ramp, like the aggressor signal, but rather an up-and-down pulse. This occurs because the crosstalk occurs only when the aggressor signal is *changing*, i.e., the crosstalk pulse's shape is related to the *derivative* of the aggressor signal's shape. The time duration of the forward pulse is therefore equal to the switching time of the aggressor signal.

The height of the crosstalk pulse depends on how strongly the two traces are coupled capacitively; the coupling strength, in turn, depends on all of the details (geometric and material) of the PCB cross section in which the traces lie. Calculating the coupling strength is difficult; in LineSim/BoardSim Crosstalk, that job is performed automatically by the built-in field solver (see Chapters 3 and 8 for details).

The crosstalk pulse tends to grow in height proportionally to the length over which the two traces are parallel, i.e., the longer the traces run side-by-side, the higher the crosstalk pulse is. However, there is a limit to this effect; after a while, the amplitude of the pulse tends to approach a limiting value. This occurs because the aggressor signal slowly loses energy to the victim trace, and also because the victim trace couples back to the aggressor.

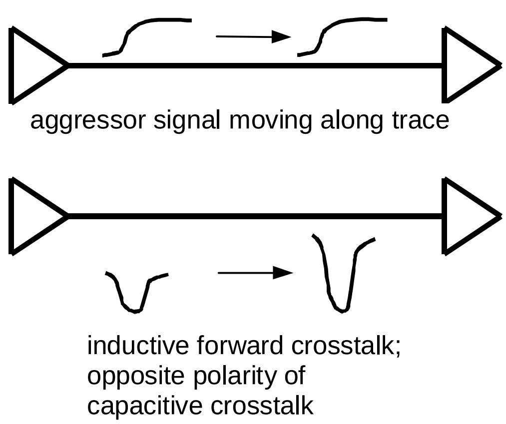
The height of the crosstalk pulse also tends to increase with the slew rate of the aggressor signal, i.e., the faster the driver signal switches (and the further it swings), the higher the crosstalk pulse is. This is the reason that faster-switching driver ICs tend to generate more crosstalk.

Inductive Forward Crosstalk

As an aggressor signal travels down its trace, its time-varying *magnetic* field also generates a crosstalk pulse; this is the inductive component of forward crosstalk. In fact, everything described in the preceding section about capacitive forward crosstalk — generates a pulse on the victim trace, pulse length equals aggressor switching time, pulse height is approximately proportional to trace length and to aggressor slew rate — is true of inductive crosstalk.

But there is one major difference: the polarity of inductive forward crosstalk is *opposite* that of capacitive forward crosstalk. See Figure 10-5. This means that in the forward direction, the capacitive and inductive components of crosstalk are *competing*, i.e., they tend to cancel each other out. If the capacitive and inductive coupling strengths are exactly equal, then no forward crosstalk will occur at all.

Figure 10-5: Inductive portion of forward crosstalk signal; opposite polarity of aggressor signal and capacitive crosstalk



In practice, you would rarely see perfect cancellation between the capacitive and inductive components of forward crosstalk. But for many cross sections, the forward crosstalk is indeed fairly small, and reverse crosstalk becomes the major concern. This is often the case especially for traces on stripline layers (i.e., between two planes), because the capacitive coupling is usually enhanced. But there's really no way to know for certain without simulating.

Notice that if you *do* see a forward crosstalk pulse, you can tell from the polarity of the pulse whether your traces are more capacitively or inductively coupled. If the pulse has the same polarity as the aggressor signal that created it, capacitive coupling dominates; if the pulse has the opposite polarity, inductive coupling is stronger. (On PCBs, the inductive coupling is usually stronger.)

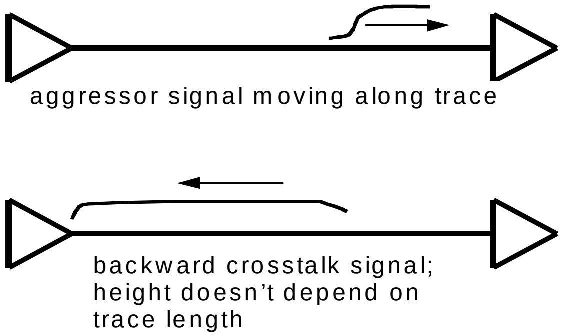
Details of Backward Crosstalk

Backward crosstalk is caused by the same physical mechanisms as forward crosstalk — time-changing electric and magnetic fields from the aggressor signal, which induce both a capacitive and inductive signal on the victim trace — but there are several major differences between the backward and forward signals.

The biggest difference is the time duration of backward signals. Forward crosstalk signals are short pulses which last only as long as the switching time of the aggressor signal. This occurs because forward signals travel at the same speed and *in the same direction* as the aggressor signal.

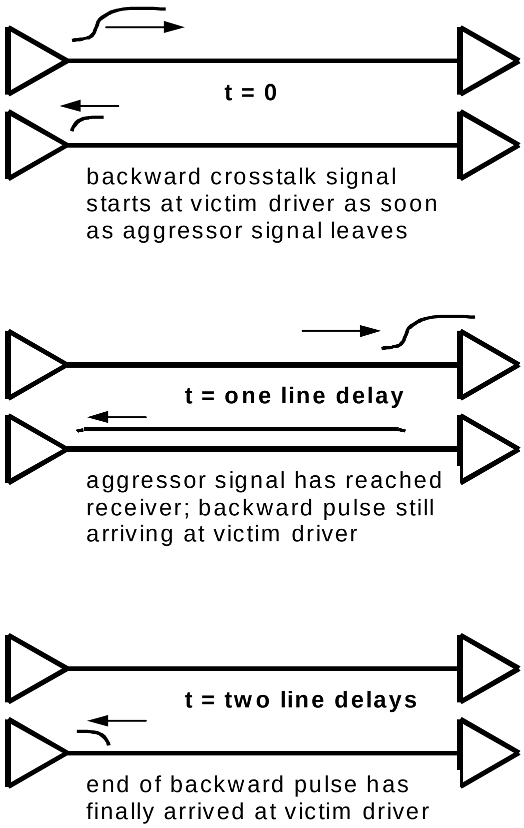
But backward crosstalk is launched in the *opposite direction* of the aggressor signal's travel. Therefore, backward crosstalk does not "pile up" like forward crosstalk does; rather, it "flows out" behind the aggressor signal, and forms a long pulse. (The "speedboat analogy" may help make this clearer; backward crosstalk corresponds to the long wake *behind* the boat.) See Figure 10-6. Unlike with forward crosstalk, the height of the backward pulse is not related to the trace length.

Figure 10-6: Shape of backward crosstalk signal



In fact, the time duration of a backward crosstalk pulse is twice the delay length of the aggressor trace. To see why this occurs, consider Figure 10-7. Suppose you are watching the backward pulse from the vantage point of the victim-trace driver IC. You see the backward pulse start as soon as the aggressor signal leaves the driver; when the aggressor signal reaches the far end of the aggressor trace, it is still generating a backward pulse, and the far end of it doesn't reach you until another line-delay's-worth of time. Hence the total duration of the pulse is *two* aggressor-trace line delays.

Figure 10-7: Length of backward crosstalk signal



Reflection of Backward Crosstalk from Victim Driver IC

Of course, you generally don't care about crosstalk at a driver IC — it's receiver ICs that matter. So in Figure 10-7, why do we even care about the backward pulse? The answer is that driver ICs, because they are normally low-impedance, reflect rather than absorb crosstalk signals. So the backward crosstalk pulse that reaches the victim-net driver IC in Figure 10-7 will reflect back toward the receiver IC. This reflection of backward crosstalk is why it's critical in crosstalk simulations to model the victim-trace driver IC. In LineSim or BoardSim simulations, you would usually do this by applying an IC model to the victim-trace driver location, then setting it to be "stuck low" or "stuck high" (i.e., driving, but not actively switching). See Figure 10-8.

Since the driver IC is almost always lower in impedance than the trace itself, the reflection off the driver usually causes the backward crosstalk pulse to invert.

Note: *Because many driver ICs have different impedances when driving high than when driving low, it is often important to check crosstalk waveforms with the victim-trace IC model in both states, stuck high* ***and*** *stuck low. However, the low-side impedance of a driver is usually the same as or less than the high-side impedance, so if you simulate in only one of the two stuck states, "stuck low" is the best choice (it usually generates a maximum reflection).*

Polarity of Backward-Crosstalk Components

Unlike with the capacitive and inductive components of forward crosstalk, the backward components have the same polarity. This means that backward crosstalk never cancels itself out like forward crosstalk sometimes does. The polarity of both backward components is the same as that of the aggressor signal, e.g., a rising edge on the aggressor trace creates a positive capacitive and inductive backward pulse. The height of the total backward pulse is the sum of the two component heights.

However, remember that if you observe the backward pulse at the victim-trace receiver (assuming the IC positions in Figure 10-7), you are looking at a signal that has been reflected *and inverted* by the victim driver IC. So you often observe backward crosstalk as having the opposite polarity of the aggressor signal. See Figure 10-8.

Figure 10-8: Backward crosstalk reflecting off of victim driver IC and inverting

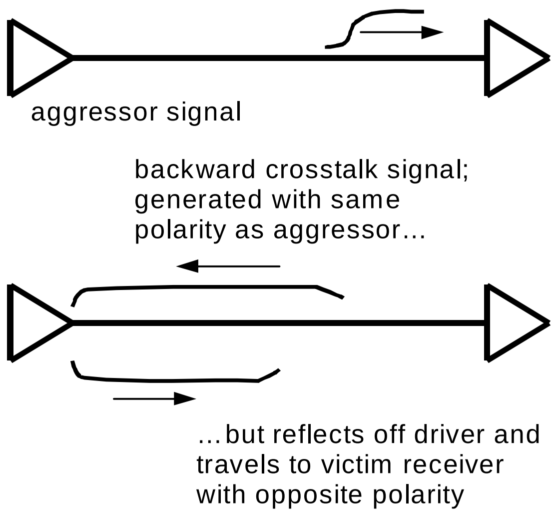
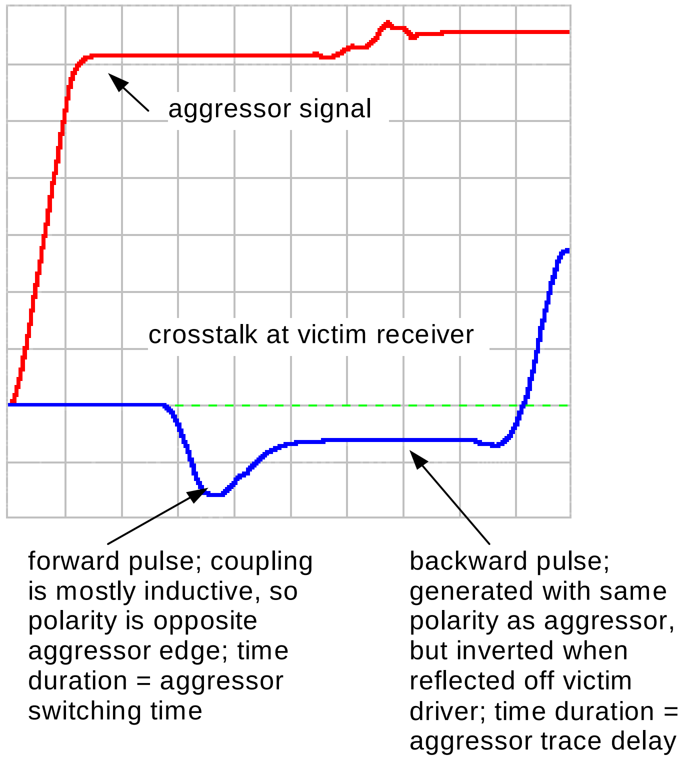


Figure 10-9 shows a "classic" crosstalk waveform at the victim-trace receiver IC, for two traces on a microstrip stackup layer (for which inductive coupling normally dominates capacitive). The trace and IC topology is assumed to be as Figure 10-8 (i.e., aggressor and victim drivers on same end, receivers on same end). The waveform interpretation is shown in the figure.

Figure 10-9: "Classic" crosstalk waveform at victim-trace receiver IC



Electrical Parameters of Coupled Transmission Lines

A trace on a PCB can be treated as a mathematical construct called a "transmission line," provided that the distance between the trace and a ground-return structure is small compared to the wavelength of the signals propagating along the trace. For PCBs with solid ground/Vcc planes, the transmission-line model is usually very accurate. (It may not be for boards with no planes — single- or double-sided PCBs — or for boards on which the plane layers are "cut" or otherwise compromised.)

Transmission-line theory does not deal explicitly with electric and magnetic fields. Rather, a transmission line is defined in terms of the capacitance and inductance that is distributed uniformly along the length of the line. However, there is a close link between these circuit quantities and the underlying field theory: capacitance is basically a measure of how much electric field is produced when a given quantity of charge is placed on a conductor, and inductance measures the amount of magnetic field that “links” a circuit when a given current flows in the circuit’s conductor.

Uncoupled Transmission Lines

The behavior of a single, uncoupled transmission line (providing that loss mechanisms are ignored) is fairly straightforward. The line’s total capacitance (C) and total inductance (L) combine to create a propagation delay that is given by:

$$tpd = \sqrt{L \cdot C} \qquad (if \;\; L \; and \;\; C \; are \; total \; values)$$

where *L* and *C* are the *total* inductance and capacitance of the transmission line.

If you are using L and C per unit length instead, then the above equation must be multiplied by the length of the transmission line:

$$tpd = \sqrt{L \cdot C} * l \qquad (if \;\; L \; and \;\; C \; are \;\; per - unit - length \; values)$$

where *L* is the inductance per unit length of the transmission line, *C* is the capacitance per unit length of the line, and *l* is length of the line.

The distributed C and L also create a property called “characteristic impedance,” which determines the ratio of voltage to current that flows in each direction along the line. Characteristic impedance plays a central role in how the line responds to an initial driver-IC impulse (i.e., how much voltage “steps” into the line) and how the line generates reflections at its ends.

The characteristic impedance (Z_0) is given by:

$$Z_0 = \sqrt{L/C}$$

Note: *In the equation for characteristic impedance above, you can use either total or per-unit-length values of L and C.*

In both of these equations, note that C and L are single numeric values. For example, for a typical 8-inch microstrip transmission line, total C = 16 pF and total L = 84 nH.

Coupled Transmission Lines

Capacitance and Inductance

The mathematics for coupled transmission lines bears strong resemblance to the theory of uncoupled lines, but there are some important — and surprising — differences. First, the quantities C and L become *matrices* instead of being single numbers. For example, the same microstrip line referred to in the previous section, if another trace is placed in parallel with it, 8 mils away, has the following capacitance matrix:

$$\mathbf{C} = \begin{bmatrix} 16.3\text{pF} & -2.72\text{pF} \\ -2.72\text{pF} & 16.3\text{pF} \end{bmatrix}$$

At first glance, you might think this result is wrong, because some of the capacitance values are *negative*. It is correct, however.

Each diagonal value in the matrix represents the capacitance of the corresponding trace when that trace is charged to 1V and all other traces are grounded. The off-diagonal values in each column of the matrix represent the capacitances between the 1-V trace and the other traces. But since the 1-V trace is positively charged, all other traces accumulate negative charge; and since capacitance is defined as the ratio of charge to voltage (Q/V), their matrix capacitance values are also negative.

***Note:** If the concept of negative capacitance bothers you, just ignore the negative signs and concentrate on the magnitude of the values. The negative signs are important in the mathematical formalism of coupled transmission lines, but intuitively not of much use. LineSim/BoardSim preserves the negative signs in its field-solver output reports because if the capacitance matrix is transferred to another tool (e.g., SPICE), the negative values must be used.*

Similarly, the inductance matrix for the two side-by-side microstrip traces is:

$$\mathbf{L} = \begin{bmatrix} 82.6\text{nH} & 23.0\text{nH} \\ 23.0\text{nH} & 82.6\text{nH} \end{bmatrix}$$

Per-Unit versus Absolute Values of C and L

In the equations given in the section "Uncoupled Transmission Lines" above, **C** and **L** were *total* values, for the entire length of the transmission line. It is more traditional (especially with field solvers) to make **C** and **L** *per-unit-length* values, e.g., to give **C** in units like pF/m and **L** in nH/m. When specified this way, you must multiply **C** and **L** by the length of a particular line (or set of coupled lines) in order to get total **C** and **L**.

Given in per-unit-length terms, the matrices shown in the preceding section become:

$$\mathbf{C} = \begin{bmatrix} 80.2\text{pF/m} & -13.4\text{pF/m} \\ -13.4\text{pF/m} & 80.2\text{pF/m} \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} 406\text{nH/m} & 113\text{nH/m} \\ 113\text{nH/m} & 406\text{nH/m} \end{bmatrix}$$

***Note:** The values in these matrices result from dividing the values in the total L and C matrices by the length of the traces, 8 inches, and then converting inches to meters.*

Characteristic Impedance

A pair of coupled traces also has a characteristic impedance, but as is the case with **L** and **C**, $\mathbf{Z_0}$ is also a matrix quantity. For example, for the pair of traces described above, the characteristic-impedance matrix is:

$$\mathbf{Z_0} = \begin{bmatrix} 72.1\Omega & 16.1\Omega \\ 16.1\Omega & 72.1\Omega \end{bmatrix}$$

For an uncoupled transmission line, the impedance Z0 gives the ratio of voltage to current flowing in either of the two directions on the line (i.e., either forward or backward). For example, if you consider one end of the transmission line, if it is carrying a voltage V_f toward you, then you can find the current I_f which is traveling toward you from the expression:

$$V_f = Z_0 I_f$$

Note: *It is not true that the ratio of total voltage to total current on a transmission line at any point is equal to Z_0. Rather, this equation holds separately at every point for both the forward and backward waves traveling on the line. But the total voltage is the sum of these waves, and the ratio V_{total}/I_{total} is not equal to Z0.*

However, when you initially drive into a line, before any reflections have had a chance to return to the driver, V_{total}/I_{total} does equal Z_0, since V_{total} and I_{total} consist of only one wave. Then when a reflected wave comes back, this relationship ceases to be true.

For a coupled transmission line, the same equation holds, but the quantities are now all matrices:

$$\mathbf{V_f} = \mathbf{Z_0} \mathbf{I_f}$$

The matrix nature of the characteristic impedance causes the lines to exhibit crosstalk. For example, suppose you drive a 100-mA pulse into the near end of one of the coupled microstrip traces we've been discussing, but no current into the near end of line 2. You might expect a voltage to appear on line 1, since you forced current to flow in it, but no voltage on line 2 since you didn't drive it. But according to the equation above, since the lines are coupled, this is not the case:

$$\mathbf{V} = \mathbf{Z_0}\,\mathbf{I} \;\Rightarrow$$

$$\begin{bmatrix} V1 \\ V2 \end{bmatrix} = \begin{bmatrix} 72.1 & 16.1 \\ 16.1 & 72.1 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0 \end{bmatrix} \;\Rightarrow$$

$$\begin{cases} V1 = (72.1)(0.1) + 0 = 7.2V \\ V2 = (16.1)(0.1) + 0 = 1.6V \end{cases}$$

Even though you forced current only into line 1, a 1.6-V signal appeared on line 2. This occurred because the lines are coupled; the matrix impedance causes signals to appear on both lines. Note that it is the *off-diagonal* terms in $\mathbf{Z_0}$ that embody the coupling.

Symmetry of Matrix Parameters

All of the matrices that define coupled-transmission-line parameters (e.g., **C**, **L**, $\mathbf{Z_0}$) are symmetric, i.e., element a_{ij} = element a_{ji}. (Another way to say this is that the matrix "reflects" across the diagonal.) This occurs because each off-diagonal element describes the coupling between a pair of traces, and this coupling has to be symmetric.

For example, in a capacitance matrix, element C_{12} represents the capacitance between trace 1 and trace 2 in some coupled region. Suppose C_{12} = 50pF/m. Then C_{21} must also be 50pF/m, because it makes no sense that "looking" in one direction between a pair of traces you would find one capacitance value and looking in the opposite you would find another.

One the other hand, the diagonal elements in the parameter matrices can all have different values. This occurs because these values represent the "self" or "to-ground" values of each trace. In the example we've been using in this section, the two diagonal values happened to be equal, because the two traces were physically symmetric. For example, when both traces were on the same layer and had the same width, we found

$$\mathbf{Z_0} = \begin{bmatrix} 72.1\Omega & 16.1\Omega \\ 16.1\Omega & 72.1\Omega \end{bmatrix} \qquad (\textit{both traces } 8 \textit{ mils wide})$$

But suppose we halve the width of trace 2. Then we find

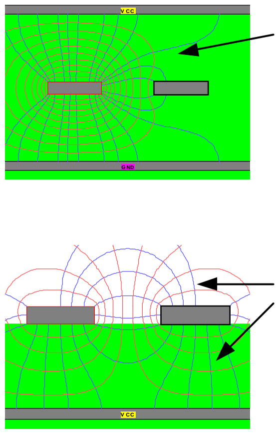
$$\mathbf{Z_0} = \begin{bmatrix} 72.5\Omega & 18.9\Omega \\ 18.9\Omega & 90.8\Omega \end{bmatrix} \qquad (\textit{one trace } 8 \textit{ mils wide}, \textit{ one } 4 \textit{ mils})$$

Note that the diagonal values now differ from each other. Each trace has a different impedance “to ground”; since trace 2 is narrower, it has less self-capacitance and more self-inductance than trace 1, and therefore has a higher diagonal impedance (90.8 ohms versus 72.5 ohms). But the off-diagonal coupling impedance is the same in both position Z_{12} and Z_{21}, as it must be.

Propagation Modes — Single-Dielectric versus Layered-Dielectric Traces

The behavior of coupled transmission lines changes depending on whether the lines are in a stripline configuration, where the lines’ electric and magnetic fields “see” only a single dielectric value, or in a microstrip or buried-microstrip configuration where the fields penetrate dielectrics with two or more permittivities. See Figure 10-10.

Figure 10-10: Single-dielectric versus layered-dielectric cross sections



When coupled transmission lines are in a single-dielectric configuration, they behave much like single, uncoupled lines: each line has the same propagation velocity, which is related in a simple way to the speed of light:

$$v = {}^{c}\!/\!\sqrt{\varepsilon_r} \qquad \textit{(for striplines)}$$

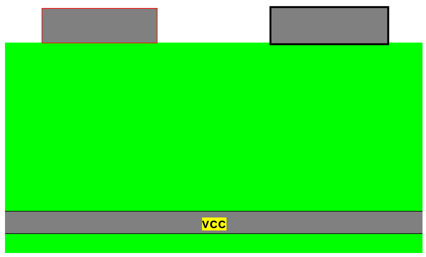
where v is the propagation velocity on each transmission line, c is the speed of light, and ε_r is the permittivity (i.e., dielectric constant) of the PCB's dielectric material. For example, coupled traces on a stripline layer of a PCB built from FR-4 with dielectric constant of 4.3 each propagate signals at v = 0.48c (i.e., at 48% of the speed of light).

However, the situation changes in an interesting way if the same traces are moved to a microstrip or buried-microstrip layer so that the trace's fields exist in two dielectrics, FR-4 *and* air. Now, the traces will support multiple propagation modes, each with a different propagation velocity. The next section describes this effect in detail.

Multi-Speed Propagation

Consider the pair of coupled microstrip traces shown in Figure 10-11. Since each trace, when it carries a signal, will generate electric and magnetic fields that exist in both FR-4 and air, it's fairly obvious that the propagation velocity can't be given by the simple expression of the previous section — after all, there are two different dielectric constants involved now, not one as with a stripline.

Figure 10-11: A pair of coupled microstrip traces



One reasonable guess as to the actual behavior of the pair might be that each trace propagates a signal at one speed that is based on some sort of average dielectric constant (something between 4.3 and 1.0). Another is that there is a continuum of speeds ranging from the one predicted by the permittivity of FR-4 to the speed in air. But neither is correct.

Instead, *two speeds* exist, each associated with a distinct "propagation mode" for the pair of traces. Each trace carries some amount of energy in *each* mode, i.e., if you drive a signal down trace 1, it will propagate some portion of the signal energy in mode 1 at speed 1, and the remaining energy in mode 2 at speed 2. Trace 2 also propagates signals in both modes.

This behavior is true not only of coupled pairs, but of sets of coupled traces of any size. In general, if there are *N* coupled traces in a multi-dielectric configuration, the traces will support *N* distinct propagation modes, each trace carrying a mixture of all modes.

Example 1: A Pair of Coupled Microstrip Traces

As an example, consider the pair of microstrip traces shown in Figure 10-11. The traces are 8 mils wide, 8 mils apart, and 10 mils above a ground plane on a layer of dielectric with permittivity 4.3.

When a signal is sent down either of these traces, it propagates partly in one mode, and partly in another. The two modes have the following characteristics:

Trace	**Propagation Mode**	**Speed (as a percentage of light speed)**	**Percentage of Energy Traveling in This Mode**
1 or 2	1	63.7%	50%
1 or 2	2	56.6%	50%

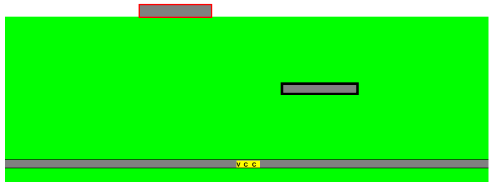
Note that both traces carry half of a propagating signal's energy in mode 1 and mode 2; this occurs because of the geometric symmetry of the cross section in Figure 10-11. If the traces were in a stripline configuration with the same dielectric material, signals would travel at 48% of the speed of light; here, with the mixture of FR-4 and air, both modal speeds are higher (64% and 57% of *c*).

Example 2: A Pair of Traces in an Asymmetric Configuration

In the preceding example, the symmetric breakdown of signal energy (50% in each mode, for both traces) was due to the symmetry of Figure 10-11's cross section. If the geometry is asymmetric, then the distribution of energy across modes also becomes, in general, asymmetric.

Consider the geometry of Figure 10-12. It is the same as for Figure 10-11, except that a second "buried microstrip" trace layer has been added to the stackup, and trace 2 has been placed on the new layer. Now the two propagation modes have the characteristics in the following table.

Figure 10-12: A pair of traces in an asymmetric cross section



Trace	Propagation Mode	Speed (as a percentage of light speed)	Percentage of Trace's Energy Traveling in This Mode
1	1	59.6%	91.3%
1	2	49.7%	8.7%
2	1	59.6%	14.9%
2	2	49.7%	85.1%

Here, the breakdown of signal energy into the propagation modes is significantly different for one trace versus the other. Trace 1 carries energy mostly in mode 1, which has the higher propagating speed (60%of *c*); this is sensible because trace 1 lies partly in air. Trace 2 carries energy mostly in mode 2, which is the slower mode (50% of *c*); again, this seems reasonable, because trace 2 is buried in dielectric and less of its fields are in air.

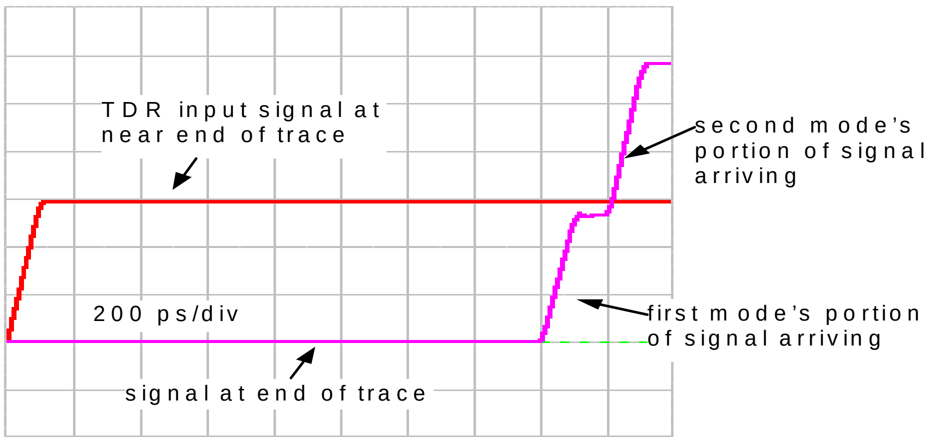
Signal Dispersion

The fact that coupled traces in a layered dielectric support multiple propagation speeds means that a signal is at least slightly distorted when it travels down such a trace. In particular, some portion of the signal will arrive before others, resulting in a "stair-step" effect.

For example, if a TDR (time-domain reflectometer) drives a fast edge into the one of the traces shown in Figure 10-11 (the symmetric microstrip example), and the signal is probed at both the TDR output and the trace's terminated end 12 inches away, the resulting waveforms are as in Figure 10-13.

Note that the input waveform is a nearly perfect ramp, but by the time this signal reaches the end of the trace, it has broken noticeably into two components, one of which arrives faster than the other. If you measure the end-of-the-line waveform carefully and compare it to the propagation data in the table for "Example 1" above, you'll see that the percentage of signal in each mode and the difference in arrival times matches the table's data well.

Figure 10-13: TDR waveforms for the one of the traces in Figure 10-11's cross section



From a practical standpoint, several things should be pointed out about this dispersive effect:

- we drove only *one* of the traces in the pair to produce this waveform; doing so stimulated both propagation modes, and resulted in a "split" signal; however, had we driven in differential or common mode only, we would have excited only one of the propagation modes, and the entire signal would have traveled at a single speed and arrived without dispersion — see "Differential/Common Modes and Propagation Speeds" below for more details

- unless you're running with very fast driver edges, even if you do drive in a "non-pure" mode (i.e., not purely differential or common mode) you probably won't observe the effect; the waveforms above come from a TDR with a 100-psec rise/fall time

- the difference in propagating speeds is rarely wider than shown in the preceding example; the typical range between fastest and slowest modal speeds is 10%-20%
- transmission lines have multiple ways of dispersing a signal; the effect described in this section is only one of them

Differential Signals

Differential signaling is becoming increasingly important in electronics. Differential methods have been in use for many years, of course, but there has been renewed interest recently as new high-speed IC technologies have sought to push bit rates into the hundreds-of-MHz range. Examples of new IC technologies that use differential signaling include LVDS ("low-voltage differential signaling") and various PECL-like CMOS-based families. These kinds of devices are becoming particularly important in telecommunications, networking, and high-speed computer applications.

Proponents of differential signaling cite a number of benefits, probably the most-important of which are immunity to external noise and reduced generation of radiated emissions. However, detailed "philosophies" of differential design vary considerably. For example, some designers prefer to couple their differential traces strongly; other advocate weak coupling.

However, many newer differential-IC technologies require a certain line-to-line terminator to achieve proper DC biasing of the differential output buffers. Since this same resistor is also the differential terminator for the trace pair, requiring a certain resistor value for DC bias essentially forces you to design your traces to have a matching differential impedance. (Thus, you may not have the luxury of choosing how tightly coupled your lines are.)

Differential Traces in LineSim and BoardSim

It is important to realize that differential traces in LineSim and BoardSim are not treated in the field solver or simulation engine as a special case. Rather, the programs handle *any* set of coupled traces in the same way, whether there be two nicely balanced differential traces in the cross section, or two highly asymmetric traces, or five coupled traces.

However, if your cross section contains only two traces, LineSim/BoardSim recognizes that you may be designing a differential pair, and automatically changes its impedance display and field-solver output report (LineSim) or coupling-region-viewer impedance display (BoardSim) to include differential impedance and other parameters of interest for differential design. For details on what information is displayed, see Chapters 3 and 8.

Differential and Common Modes

The Concept of “Propagation Modes”

Conceptually, the term “propagation mode” refers to a manner in which signals are arranged on a set of traces in order to propagate the signals. A “basis set” of propagation modes is a collection of modes that could be used in some mixture to create any arbitrary set of real signals on the traces. In the section “Multi-Speed Propagation” earlier in this chapter, the propagation modes discussed were a set that have the added physical significance of each mode propagating energy at a different, unique speed (a phenomenon that occurs only with layered dielectrics; see section “Multi-Speed Propagation” for details).

In differential signaling, designers typically conceive of a pair of modes called “differential mode” and “common mode.” The differential mode is one in which if one trace carries the voltage +V, the other trace carries –V (i.e., the two traces always carry opposite voltages). The common mode is one in which if one trace carries +V, the other also carries +V.

Note that it is conceptually possible to describe *any* pair of real signals traveling on the two traces as some mixture of these two modes. For example, a mostly differential signal that had a small common-mode component to it could be constructed by mixing 80% differential mode with 20% common mode.

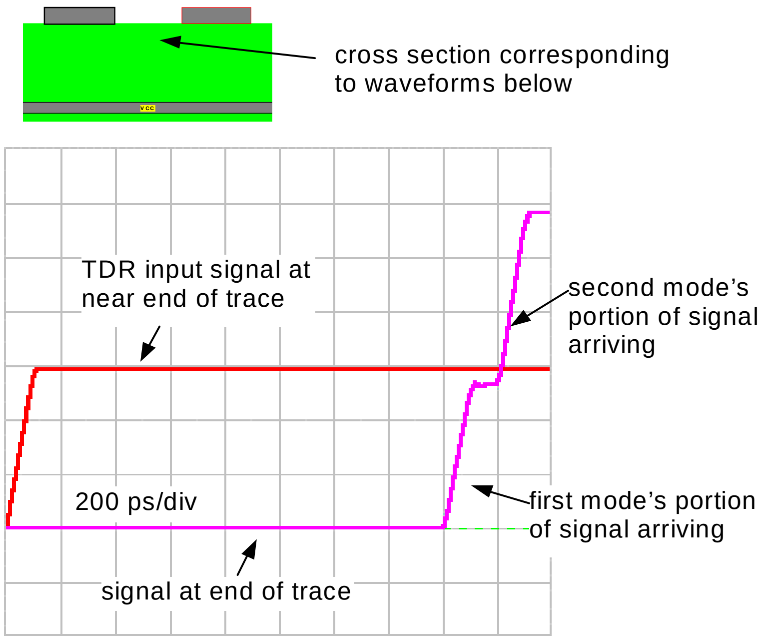
Differential/Common Modes and Propagation Speeds

For two-trace microstrip and buried-microstrip configurations in which the traces are symmetrically arranged (i.e., each trace is on the same layer, has the same width and thickness, etc.), it turns out that the mode set that describes the two propagation speeds and the differential/common mode set coincide, i.e., they're the same. Thus, for symmetric trace arrangements, driving purely differential signals means that only one mode is stimulated, and only one propagation speed results. The same is true for driving in common mode (both traces carrying the same rather than the opposite polarity), except that the other speed results.

Figure 10-15 repeats Figure 10-11 and Figure 10-13 to help illustrate this point. The cross-section picture shows two microstrip traces in a symmetric configuration, i.e., the two traces share the same layer, are the same thickness and width, etc. — generally, they can't be distinguished from each other except that one is on the left and the other on the right. This means that if these traces are driven purely differentially or purely in common mode, only one propagation speed will result.

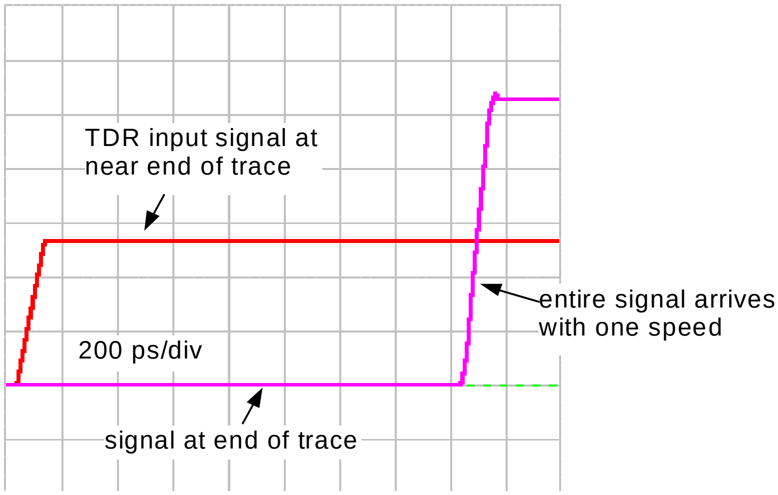
The waveform in Figure 10-15 (same as Figure 10-13 earlier) shows what happens if one trace is driven and the other not. This is not a "pure" mode: differential mode corresponds to driving the traces with signals [+V,-V] and common mode means driving [+V,+V], but here we're driving [+V,0]. So we would expect a mixture of differential and common mode to be excited, and to see part of the signal arriving with one velocity and a part with the other — exactly as Figure 10-15's waveform shows.

Figure 10-15: A symmetric microstrip trace pair, driven with a mixture of differential and common modes



Now suppose we drive instead differentially, which should produce a "pure" mode and propagate at only one speed. Figure 10-16 shows the resulting waveform.

Figure 10-16: Same cross section as Figure 10-15, but driven differentially



Indeed, as expected, the entire signal does arrive does arrive with one propagation speed, and therefore one delay.

Thus, you can clearly see one benefit of driving a pair of coupled traces differentially: if the traces are microstrips or buried microstrips (i.e., located in layered dielectrics), the dispersion which would normally result from driving the traces in an arbitrary manner is eliminated.

However, it is important to note that this benefit is achieved only if the traces are symmetric (i.e., interchangeable geometrically); otherwise, differential mode will not correspond to a single-speed propagation mode. The safest way to achieve this symmetry is to route two traces of the same width and thickness together on the same stackup layer.

Again, it should be noted that these considerations apply only to traces in a layered-dielectric configuration (microstrips or buried microstrips). For striplines, all propagation is always at a single velocity.

Differential and Common-Mode Impedance

In differential signaling, another commonly encountered concept is that of "differential impedance" and "common-mode impedance." There are various ways of motivating the definitions for these impedances, but the most practical from a signal-integrity viewpoint is based on the discussion of propagation modes in the preceding sections (see "Differential and Common Modes" above):

For a pair of ***symmetric*** *coupled traces,*

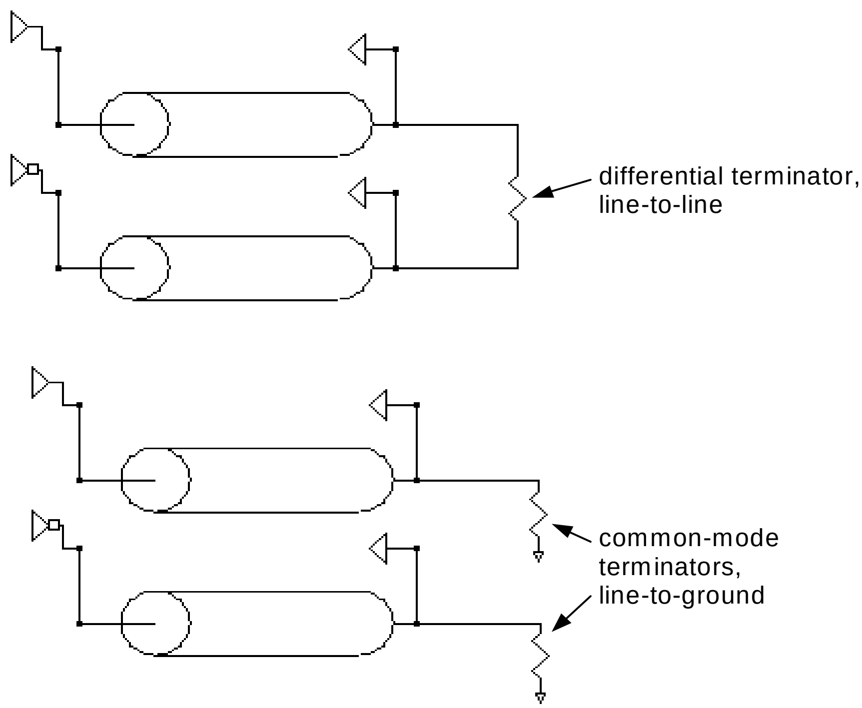
- *the differential impedance is the trace-to-trace resistance that will properly terminate a pair of signals driven in differential mode*
- *the common-mode impedance is the trace-to-ground impedance (for each trace) that will properly terminate a pair of signals driven in common mode*

For asymmetric traces, these impedances are still useful as terminators, but they will not function as well as for symmetric traces because asymmetric configurations introduce multiple propagation speeds into both differential and common modes. For details, see "Differential/Common Modes and Propagation Speeds" above.

LineSim and BoardSim automatically display differential impedance in the Edit Coupling Regions dialog box (LineSim) or coupling-region viewer (BoardSim) when you're working with a two-trace coupling region. The values of differential and common-mode impedance are also given in the field solver's report file (LineSim). See Chapters 3 and 8 for details.

Figure 10-17 illustrates the use of these impedances for terminating purposes.

Figure 10-17: Differential and common-mode terminators



These definitions explain why differential trace pairs are often terminated with only a single resistor, line-to-line. If the traces are indeed driven with “pure” differential signals, nothing else is required for perfect termination. However, if the actual driven signals contain a mixture of differential and common modes, the common-mode portion will *not* be terminated by a line-to-line resistor.

Generally, the only termination which can guarantee proper termination of a pair of traces given non-ideal signals is a three-resistor network that simultaneously implements both the differential and common-mode impedances. See “Terminating Coupled Transmission Lines” below for more details.

Relationship of Impedances to Characteristic-Impedance Matrix

The values of differential and common-mode impedance are derived from the trace pair’s characteristic impedance matrix. (For details on the Z_0 matrix, see “Characteristic Impedance” above.)

If the traces are in a symmetric configuration (same width, thickness, distance from a ground plane, etc.), then the following relations hold:

$$Zdiff = 2(Z_{11} - Z_{12})$$
$$Zcomm = Z_{11} + Z_{12} \qquad (\textit{for symmetric traces})$$

If the traces are asymmetric, then the expression for differential impedance becomes:

$$Zdiff = Z_{11} + Z_{22} - 2Z_{12} \qquad (\textit{for asymmetric traces})$$

Again, the asymmetric case will not terminate perfectly with this value because differential mode will excite two propagation speeds; see “Differential/Common Modes and Propagation Speeds” above for details.

It should be emphasized that whenever you are working with a two-trace coupling region, LineSim and BoardSim calculate the differential and common-mode impedances automatically for you, so you should never need to make these calculations manually.

Terminating Coupled Transmission Lines

The preceding section (see “Differential and Common-Mode Impedance” above) discussed termination of differential trace pairs specifically. In that context, the concepts of differential and common-mode impedance are useful.

However, it is possible to draw more-general conclusions about the termination of coupled transmission lines. These concepts can be extended, for example, to an arbitrary number of lines, and have some interesting properties relative to eliminating crosstalk at line ends.

Termination into the Characteristic-Impedance Matrix

If a single, uncoupled transmission line is terminated into its characteristic impedance, i.e., into a resistance equaling the line's Z_0, then no reflections will be generated from the end of the line.

For a set of coupled transmission lines, nearly the same statement can be made: that terminating into an array of resistors that synthesizes the impedance $\mathbf{Z_0}$ will perfectly terminate the lines. Note that because, for coupled lines, $\mathbf{Z_0}$ is a matrix quantity, the terminator required to implement it is an *array* of resistors. (For details on matrix impedances, see "Characteristic Impedance" above in this chapter.)

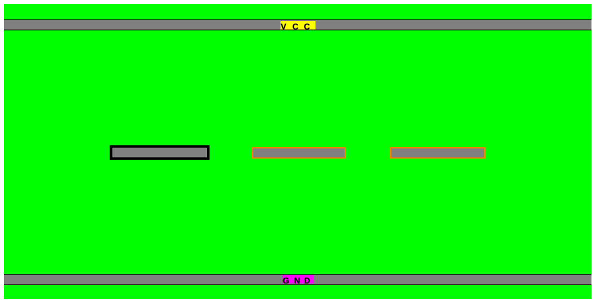
Also, the required resistors do *not* have the values in the $\mathbf{Z_0}$ matrix, rather *together* in a network, they implement the impedances in $\mathbf{Z_0}$. The array consists not only of resistors from each transmission line to ground, but also from line to line. LineSim and BoardSim calculate the proper resistances for the termination array; look for the section in the field solver's detailed report (LineSim) or coupling-region viewer's Impedance pane (BoardSim) called "Optimal Terminator-Resistor Array." For details on opening and using this file, see Chapters 3 and 8.

Such an array of resistors has fairly remarkable properties. First (as is the case for an uncoupled line), the array will eliminate reflections from each of the line ends. More surprisingly, the array will also cancel any crosstalk that appears at the line ends.

Example: Perfectly Terminating a Three-Trace Cross Section

Figure 10-18 shows an example cross section containing three coupled traces.

Figure 10-18: Cross section with three coupled traces



For this cross section, the characteristic-impedance matrix is:

$$\mathbf{Z_0} = \begin{bmatrix} 57.2\Omega & 19.8\Omega & 7.7\Omega \\ 19.8\Omega & 55.6\Omega & 19.8\Omega \\ 7.7\Omega & 19.8\Omega & 57.2\Omega \end{bmatrix}$$

To terminate the traces "into" $\mathbf{Z_0}$, a resistor array must be constructed such that an observer "looking" from each trace end would see the appropriate diagonal impedance to ground; and from trace-to-trace, would see the appropriate off-diagonal values.

In this case, the properly constructed resistor array is:

$$\mathbf{R_{term}} = \begin{bmatrix} 79.0\Omega & 142.7\Omega & 3692\Omega \\ 142.7\Omega & 111.6\Omega & 142.7\Omega \\ 3692\Omega & 142.7\Omega & 79.0\Omega \end{bmatrix}$$

This termination, if fully implemented, would have 79.0Ω, 111.6Ω, and 79.0Ω to ground from the ends of traces 1, 2, and 3 respectively; 142.7Ω line-to-line between the ends of traces 1 and 2, and 2 and 3; and 3692Ω line-to-line between traces 1 and 3. However, because 3692Ω is so large compared to the diagonal line impedances, this resistor could be omitted without effect.

If this termination is implemented, then the set of coupled traces is as perfectly terminated as possible: line-end reflections are eliminated for any set of signals driven down the traces, and crosstalk at the end of the lines is canceled.

Admittedly, such a termination is “expensive” from a component-count standpoint. It also may not be possible owing to the small value (in this case, 142.7Ω) of some of the line-to-line resistors, which might cause too much current to flow between drivers on different traces when the drivers are in opposed states. (The line-to-ground resistances could also present too heavy a load to individual drivers.) If these problems exist but it is still desired to use the terminator, AC coupling (through the addition of capacitors) may help (adds still more components, though).

In spite of possible implementation difficulties, the resistor-array terminator is a potentially powerful weapon against reflections and crosstalk. In certain critical situations, where tolerance for over/undershoot, ringing, excessive delays, or crosstalk is very low, array terminations may prove quite valuable.

Chapters 3 and 8 contain additional information about termination arrays. See sections “Optimal Terminator-Resistor Array.”

Terminating a Differential Pair with a Resistor Array

The array termination for a pair of coupled traces is a three-resistor terminator, consisting of one resistor line-to-line and two (one for each line) line-to-ground. Together, these resistors implement a combined differential/common-mode terminator. Thus, the array properly terminates any kind of signal driven down the lines: differential, common-mode, or a mixture of the two.

Index