A differential signaling system is one where a signal is transmitted down a pair of tightly coupled carriers, one of these carrying the signal, the other carrying an equal but opposite image of the signal. Differential signaling was developed to cater for situations where the logic reference ground of the signal source could not be well connected to the logic reference ground of the load. Differential signaling is inherently immune to common mode electrical noise, the most common interference artifact present in an electronic product. Another major advantage of differential signaling is that it minimizes electromagnetic interference (EMI) generated from the signal pair.

Differential pair PCB routing is a design technique employed to create a balanced transmission system able to carry differential (equal and opposite) signals across a printed circuit board. Typically this differential routing will interface to an external differential transmission system, such as a connector and cable.
It is important to note that while the coupling ratio achieved in a twisted pair differential cable may be better than 99%, the coupling achieved in differential pair routing will typically be less than 50%. Current expert opinion is that the PCB routing task is not to try and ensure a specific differential impedance is achieved, rather the objective is to maintain the properties required to ensure the differential signal arrives in good condition at the target component as it travels from the external cabling.

According to Lee Ritchey, a noted industry high-speed PCB design expert, successful differential signaling does not require working to a specific differential impedance. What it does require is:

- To set each of the routing signal impedances to half the incoming differential cable impedance.
- That each of the two signal lines is properly terminated in its own characteristic impedance at the receiver end.
- That the two lines should be of equal length, to within tolerances of the logic family and the circuit frequency used in the design. The focus should be on preserving the timing, match the lengths close enough to satisfy the skew budget of the design. Example length tolerances include: high-speed USB, length mis-match should be no greater than 150 mils; DDR2 clocks need to be matched to within 25 mils.
- Use the benefit of routing the two signals side-by-side to help achieve good quality routing of matched lengths, where required it is acceptable to separate to route around obstacles.
- Layer changes are acceptable, as long as the signal impedances are maintained.

For more information, refer to the article *Differential Signaling Doesn't Require Differential Impedance*, by Lee W. Ritchey, available from [http://www.speedingedge.com/RelatedArticles.htm](http://www.speedingedge.com/RelatedArticles.htm).

---

**Defining the Differential Pairs on the Schematic**

Differential pairs are defined on the schematic by placing a [Differential Pair directive](http://www.speedingedge.com/RelatedArticles.htm) on each of the nets in the pair. Note that the net pair must be named with net label suffixes of _N and _P. Placing a Differential Pair directive on each pair net applies a parameter to the net, which has a parameter Name of DifferentialPair and a Value of True.

Differential pair definitions are transferred to the PCB during design synchronization.
Place directives on the schematic to define differential pairs.

To apply directives to a net-type object, you place a Parameter Set object so that it touches the net object, as shown above. It is the presence of a specific parameter in that Parameter Set object that the software detects, causing it to display the Parameter Set as the required directive type. To learn more, refer to the Parameter Set object page.

**Using a Blanket Directive to Identify Multiple Pairs**

If there are a large number of pairs to be defined, an alternate approach is to also place a Blanket directive, this allows you to apply directives to multiple nets that are under the blanket (it is the Net Labels that the software detects). The blanket is then tagged by a single Differential Pair directive, as shown in the image below.

The image also shows that, as well as directing that the contained nets be defined as differential pair members (via the DifferentialPair parameter), the directive also instructs all contained nets to become members of the Net Class ROCKET_IO_LINES, and to also create a Differential Pair Routing rule. In the PCB editor, this rule will be scoped to target this ROCKET_IO_LINES net class.
A Blanket can be used to configure multiple nets as differential pair members.

**Transferring the Differential Pairs to the PCB Editor**

If you have placed Differential Pair directives on nets in the schematic, the default project options settings will result in the differential pairs members being created on the PCB. The following options in the *Options for PCB Project* dialog are used to configure this:

- **Comparator tab** - *Extra Differential Pairs* (then the Different Differential Pair checks for subsequent updates, and the Rules options if you are also creating/changing design rules)
- **ECO Generation tab** - *Add Differential Pair* (then the Change Differential Pair checks for subsequent updates, and the Rules options if you are also creating/changing design rules)
- **Class Generation tab** - *Generate Net Classes* (if you are also creating a Net Class to use to scope a PCB Differential Pair Routing rule)

**Viewing and Managing Differential Pairs on the PCB**

Differential pair definitions are viewed and managed in the **PCB panel**, set to **Differential Pairs Editor**. The image below shows the pairs that belong to the class RIODiffPairs. Pair V_RX0 is highlighted, the nets in this pair are V_RX0_N and V_RX0_P. The - and + displayed next to each member net name is a system flag, indicating if it is the positive or negative member of the pair.
Defining Differential Pairs on the PCB

Differential pairs can be defined on the schematic, and they can also be defined in the PCB Editor. To create a differential pair object in the PCB editor, click the Add button when the PCB panel is set to Differential Pairs Editor mode. From the resulting Differential Pair dialog, select existing nets for both the positive and negative nets, give the pair a name and click OK. Note that any two nets can be chosen as the differential pair members.
Quickly create pairs from the named nets.

If you are pairing nets that have a consistent naming scheme (they have a common prefix and a consistent positive/negative suffix, for example TX0_P and TX0_N), then you can use the Create Differential Pairs From Nets dialog. Click the Create From Nets button in the PCB editor panel to open the dialog, then use the filters at the top of the dialog to show net pairs, based on existing net names.

Defining a Differential Pair Class

Often there is more than one differential pair that needs to be targeted by design rules. In this situation you can define classes of differential pairs, clustering them into logical groups. Classes are defined in the Object Class Explorer dialog (Design » Classes).
Using xSignals with Differential Pairs

If your differential pairs have series components in the signal path, you might find it worthwhile creating xSignals. An xSignal is a designer-defined signal path between 2 nodes - they can be 2 nodes within the same net, or they can be 2 nodes in different nets. Using an xSignal, you can define the signal path so that it includes the net on either side of the series component, as well as the series component. Route length calculations for xSignals includes the length of the path through the series component, as shown by the thin line that is displayed when an xSignal is selected in the xSignals mode of the PCB panel.

These differential pairs have been defined as xSignals, the route length includes the series component.

xSignals can be used to scope design rules, including the Matched Length and Length rules. To learn more, refer to the article Defining High Speed Signal Paths with xSignals.

Applicable Design Rules

To interactively route a differential pair, create and configure the following two design rules in the PCB Rules and Constraints Editor dialog (Design » Rules):

- **Differential Pairs Routing** - defines the routing width of the nets in the pair, the separation
(gap) between the nets in the pair, and the overall uncoupled length (the pair becomes uncoupled when the gap is wider than the Max Gap setting). Set the scope of this rule to target objects that are a differential pair, eg. IsDifferentialPair, or InDifferentialPairClass('All Differential Pairs'). Note that the Gap setting is used during placement, but not during design rule checking. During design rule checking, the distance between the nets in a pair is tested by the applicable Electrical Clearance design rule.

- **Electrical Clearance** - defines the minimum clearance between any two primitive objects (e.g. pad-pad, track-pad) on either any net, the same net or between different nets. Set the scope of this rule to target objects that are members of a differential pair, eg InDifferentialPair.

**Important Note:** as you route a differential pair, the nets in the pair will be separated by the Gap setting defined in the applicable Differential Pair Routing design rule. However, during design rule checking, all electrical objects will be tested using the applicable Electrical Clearance design rule. If the nets in the pair are placed closer together than the minimum setting allowed by the Electrical Clearance design rule, you will need to add an additional Electrical Clearance design rule targeting the differential pairs, allowing them to have a clearance equal to the Diff Pair Routing Gap setting.

**Setting the Scope of the Design Rules**

The scope of the design rule defines the set of objects that you want the rule to be applied to. Since a differential pair is an object, you can use queries like the following examples to scope the rule to target differential pairs:

- **InAnyDifferentialPair** - is the object in any differential pair.
- **InDifferentialPair('D_V_TX1')** - targets both nets in the differential pair named D_V_TX1.
- **InDifferentialPairClass('All Differential Pairs')** - targets all nets in all pairs belonging to the differential pair class called All Differential Pairs.
- **(IsDifferentialPair And (Name = 'D_V_TX1'))** - targets the differential pair object that has a name of D_V_TX1.
- **(IsDifferentialPair And (Name Like 'D*'))** - targets all differential pair objects whose name starts with the letter D.

**Using the Differential Pair Wizard to Define the Rules**

Click the **Rule Wizard** button in the Differential Pairs Editor (PCB panel) to walk you through the process of setting the required design rules. Note that the scope used for the created rules will depend on what was selected when the Rule Wizard button was clicked - if one pair was selected the rules will target the nets in that pair, but if a differential pair class was selected then the rules will target the nets and all pairs in that class.

**Routing a Differential Pair**

Differential pairs are routed as a pair - that is, you route two nets simultaneously. To route a differential pair, select **Interactive Differential Pair Routing** from the Place menu. You will be prompted to select one of the nets in the pair, click on either to start routing. The animation below shows a differential pair being routed.
Differential pairs are routed simultaneously. As you route the differential pair the standard conflict resolutions modes are available, including Walkaround, Push, Hug and Push, and Ignore obstacles. Use the:

- **SHIFT + R** shortcut to cycle through the conflict resolutions routing modes
- **Shift+Spacebar** shortcut to cycle through the available corner styles:
  - 45 degree corner
  - 45 degree arc in corner
  - 90 degree corner
  - 90 degree arc in corner
- The * key on the numeric keypad to switch layers
- The 5 shortcut to cycle possible via patterns
- **Shift+F1** shortcuts to display all of the available shortcuts.

In the arc in corner modes, press the < key to decrease the maximum arc radius, and the > key to increase the maximum arc radius. The arc size can be changed interactively by moving the cursor, the setting defines the maximum allowable arc radius, which is displayed on the Status bar when you are routing.

Differential pair routing is essentially the same as single-net interactive routing. Read the [Interactive Routing](#) article to learn more about routing.

### Matching the Lengths of Differential Pairs

Differential pairs are often used in high speed design due to their inherent immunity to noise, and the fact that they simplify the challenge of providing a high-quality return path for the signals. However, like single-sided signals, their lengths must be managed to ensure the signal timing requirements are met.
During differential pair routing, the length of each of the two nets in the pair is displayed on the Status bar, and also in the Heads-up display (\textit{Shift+H} to toggle on/off). The length values displayed in the \textit{PCB} panel are updated when you drop out of routing a pair.

Matched Length and Length Design Rules

\textbf{Length} and \textbf{Matched Length} design rules can be defined to ensure that the flight time and skew timing requirements are met. As well as being used during a design rule check (DRC), these rules are also used during interactive length tuning.

The Matched Length design rule detects the longest pair targeted by the rule scope, and uses the \textbf{Average Length} value of that pair as the reference to compare the other targeted pairs, requiring their lengths to be within \(+\) or \(-\) the Tolerance defined in the rule. The \textbf{Average Length} value is shown in the \textbf{Differential Pairs Editor} in the \textit{PCB} panel.

Within-Pair and Between Pair Design Rules

It is likely that you will have matched length requirements between the pairs, and also within each pair.

To manage this you create suitable Matched Length design rules:

1. Define a Matched Length design rule that applies across (between) the pairs. Scope the rule to apply to the required pairs, as shown in the image below on the left. Note that the \textbf{Check Between Differential Pairs} checkbox is enabled.

2. Define another Matched Length design rule that applies within the pair, this rule ensures the lengths of the two nets within each pair are within tolerance. The rule will also be scoped to apply to the required pairs, as shown in the image below, on the right. This rule should have a
higher priority than the between pairs rule. Note that the **Check Nets Within Differential Pair** checkbox is enabled.

Two matched length rules have been defined for the diff pairs - one between pairs (left image, the second within pairs (right image).

**Length Tuning a Differential Pair**

The length of pairs, and the nets within each pair, are tuned using the two length tuning commands. To tune the lengths:

1. The length of a differential pair can be accurately tuned using the **Interactive Diff Pair Length Tuning** command (**Tools** menu). During length tuning you can use shortcuts to interactively adjust the accordion style and size, or press Tab to open the **Interactive Length Tuning** dialog. In the length tuning dialog, the target length is defined:
   - **Manually**, enter the value in the **Target Length** field
   - **From a user-selected routed diff pair**
From the applicable Length and Matched Length design rules.

2. To tune a net within a pair, use the **Interactive Length Tuning** command (**Tools** menu). If you attempt to tune the longer net in the pair, the message **Target Length Shorter than Old Length** will appear.

![Diagram showing tunable lengths of differential pairs](image.png)

Refer to the **Length Tuning** article to learn more, where you will find a detailed list of the shortcuts that can be used to change the accordion style, amplitude and pitch. The article also explains how the software decides which rule settings to obey when there are overlapping settings in the Length and Matched Length design rules.

If you do not get tuning accordions appearing during length tuning, press **Tab** and check that the settings in the **Pattern** section of the dialog are sensible. For example, if the **Max Amplitude** value is very large, the software may be unable to place an accordion.

### Modifying Differential Pair Routing

During routing there will be many instances where you need to change some of the existing routing, for example you might not be happy with the pad exits and wish to re-shape them (as shown in the animation below). While you can change the existing routing using a drafting type approach of clicking and dragging track segments, it is often easier to simply re-route.

To do this, select the **Place » Interactive Differential Pair Routing** command, then click anywhere on the existing routing. Proceed to route the new path, coming back to meet the existing routing where required. This will create a loop with the old path and the new path, when you right-click or press **Esc** to terminate the route, the redundant segments are automatically removed, including any redundant vias. Differential pair routing is slightly different from single net routing, during single net routing the last segment is hollow, (the look-ahead segment), this segment is not placed when you click. Differential pair routing does not include look-ahead segments, so when you click you will place all visible segments. Position the cursor to ensure that there are no redundant segments.

If you are adjusting differential pairs by manually dragging track segments, you can either push one
pair member with the other, or drag each independently.

Using the loop removal feature to interactively re-route a differential pair - simply re-routing along a new path, the old routing loop is automatically removed.

Pairs can also be modified by dragging one net so that it pushes the other (Shift+R during dragging to cycle the mode)

To learn more, refer to the [Modifying the Routing](#) article.

**Removing a Length Tuning Accordion**

Differential pair length tuning accordions are created from multiple short track and arc segments. These can be manually selected and deleted, however it is usually more efficient to simple re-route over the top of them, the loop removal feature will then remove the redundant track/arc segments.
Using the loop removal feature to remove length tuning accordions.

**See Also**

- Interactive Routing
- Length Tuning
- Controlled Impedance Routing
- Modifying the Routing
- High Speed Design in Altium Designer
- Defining High Speed Signal Paths with xSignals

---

☑️ Thanks to Robert Feranec of the FEDEVEL Academy ([www.fedevel.com](http://www.fedevel.com)) for the use of the iMX6 Rex development board in images on this page ([http://www.imx6rex.com/](http://www.imx6rex.com/)).

---

**Source URL:** https://www.altium.com/documentation/display/ades/((Differential+Pair+Routing))_AD