



## High-speed Digital Board Design with Altium Designer

Istvan Nagy, Electronics Design Engineer, Blue Chip Technology

### Introduction

This is a tutorial for high-speed digital board design with Altium Designer. There are some design objects and rules that are not covered in this document since they are not important for understanding high-speed design constraints. There are different grades of high-speed design from the 8-bit USB microcontroller boards until the server computer motherboard designs. This article is trying to explain all possible trace length related challenges with solutions for all grades. This document mainly focuses on the Altium Designer release 10, and in few sections there are examples with Altium Designer 6.9 as well. This tutorial requires the reader to be familiar with the general PCB design with Altium Designer.

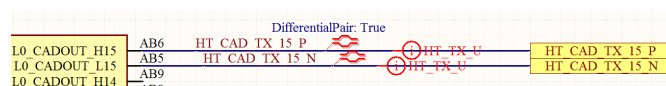
### Connection Objects

The connection objects define the connections in a circuit, group of connections, or the way of making the connections (eg differential-pair routing). Some of them are naturally created, and they are normally coming from the schematics like nets and buses. There are other objects that have to be manually defined. Some of these can be defined in the schematics level, while other can only be defined in the PCB design file. This chapter describes connection objects that have to be manually created, so we will be able to define the high-speed design rules properly based on them.

#### Differential Pairs:

For differential signals, we create differential pair objects. The best way is to create them in the schematics design. We give netnames for both

the positive and the negative net with „\_P” and „\_N” suffixes. We have to add net-class names to both nets, see later in the classes section. For the differential pairs, the net class assignment will be used for trace width rules. To add the diffpair symbol to a net, use „Place > Directives > Differential Pairs”, then attach the symbol to both nets, then doubleclick on the symbol and edit the „Name” field to give a unique name, for example the net names without the suffixes. Give the same diffpair name to both the \_P and the \_N net.



**Figure 1: Net name, Differential Pair and Net Class specification in schematics**

In the PCB layout file when we import the design the differential pair objects will be created. We can see a list of all differential pairs in the PCB Panel when we select the „Differential Pair Editor” in the panel’s top drop-down list. Here we can browse hierarchically, first select a diffpair class or „All” from the first list, then select the diffpair by its name in the second list, then we can see the two nets in the third list. In the panel we can specify what should happen when we select an object, by choosing from a small drop-down list on the PCB panel. The selection can have no effect („Normal”), or highlight („Dim”), or highlight and disable editing of another objects („Mask”). In the last two cases the selection can be cancelled by pressing the „Clear” button in the bottom-right corner of the Altium Designer window. The level of highlighting can be set in the „Mask Level”, which is next to the Clear button. In Altium Designer, the different „Panels” can be turned on or off in the right bottom corner of the editor, by clicking on the PCB-button.

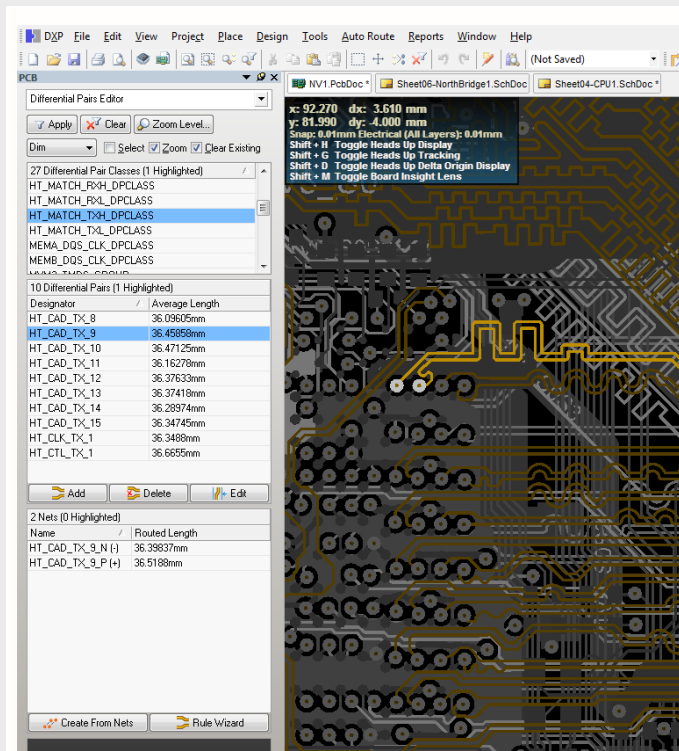


Figure 2: PCB Panel and Differential Pairs Editor

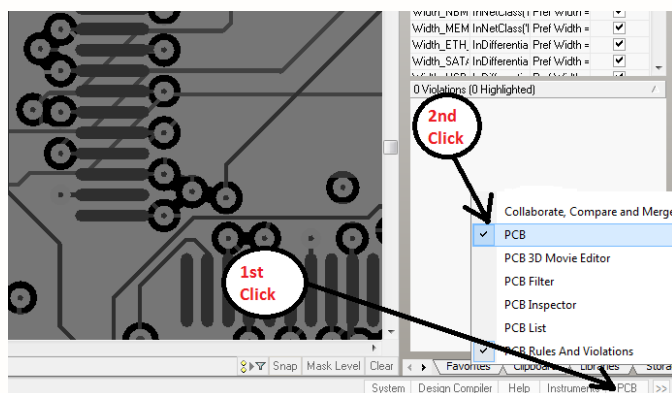


Figure 3: Altium Designer Panels

### From-Tos:

The „From-To“ objects define pin-to-pin connections or pin-pairs on multipoint nets. Based on these objects, we are able to control trace length related propagation delay from one specific pin to another specific pin or on a net, by setting up length or matching design rules where the „scope“ of the rule is a From-To or a From-To Class object. These objects can only be created in the PCB editor, not in the schematics level. On the PCB Panel, select the „From-To Editor“ in the top drop-down list. In this editor we can create the From-To objects, by first selecting a net in the first list, then selecting two component pins on the net from the second list, then pressing the button underneath the second list. The text on the button is selection-dependent, showing something similar to this: „Add From-To NETNAME (component1-pin1 : component2-pin2)“. We can also auto-generate a set of pin-pairs on a net by first selecting a net in the first list, then pressing the „Generate“ button, and select one topology option. Most of the cases the automatic option doesn't generate what we need. Unfortunately with the current version of Altium Designer we have to create the From-Tos on every net one-by-one manually. If we have a wide multipoint bus, then it can be very time consuming. On multipoint nets,

we have to create these objects on every important driver-receiver pin-pair, or in some cases based on a standard like the JEDEC DDRx memory module design specifications, known as JESD21C documents ([www.jedec.org](http://www.jedec.org)).

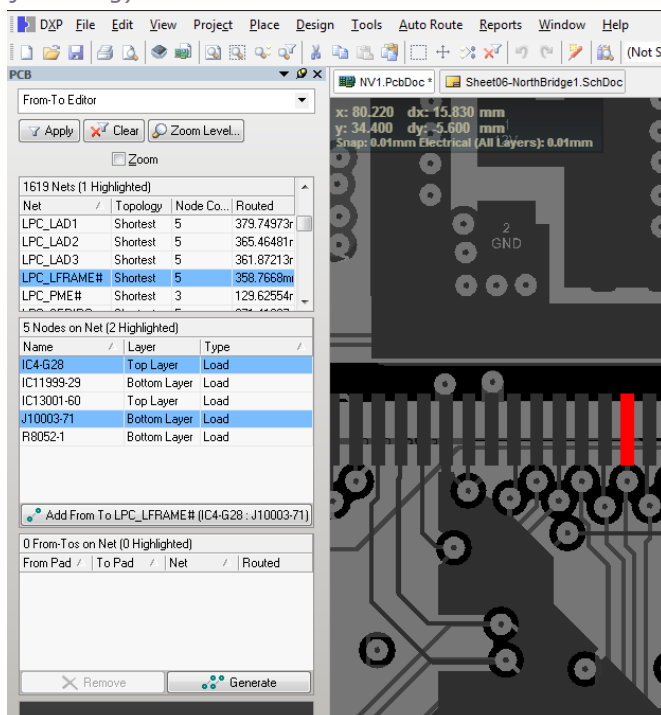


Figure 4: PCB Panel and From-To Editor

The importance of the From-To objects is the fact that we have to control the flight time on multi-point nets where the signal arrival time from a pin to another pin has to be controlled. We need net segment control normally between each driver or receiver device pair (static timing), and between terminations and active devices (signal integrity). Without From-Tos, Altium Designer would do total-etch-length control which is nothing to do with flight time on multipoint nets, it is only suitable for point-to-point nets where we have 2 components/pins on a net. In high-speed designs not only the signal integrity is important, but the static timing as well. To learn more about static timing on BCBs, refer to the two-part article: „Generalized I/O Timing Analysis“ in the Printed Circuit Design and Fabrication Magazine: <http://publish-it-online.com/publication/?i=49146> page 26, and <http://publish-it-online.com/publication/?i=50463> page 23.

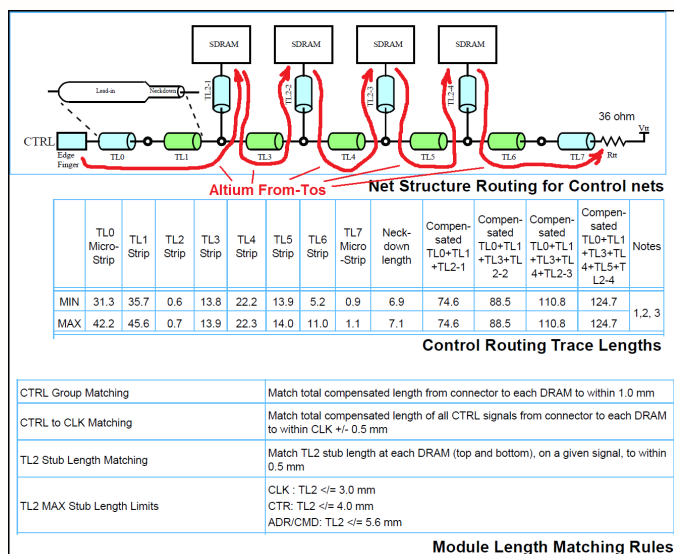


Figure 5: JEDEC specification for DDR3 memory module control signal group routing

## Buses:

If we work with wide parallel buses made of single-ended signals (like PCI or IDE), we can simplify our schematics by using „Bus“ objects. In this design guide we don't discuss „Bus“ objects any further, since they don't have much importance for trace length control design rules, and they are most of the times impractical to use as a scope of a Design Rule. We use them often for graphical aid in schematics design. For buses made of differential pairs (like PCI-Express or Hyper Transport) the Altium Designer „Bus“ object has not much use. For signal groups to be used in PCB design rules, we use class objects instead.

## Classes:

The classes are also design objects, they are groups of other design objects (connection objects or components or other types). For example Net Classes, Differential Pair Classes, From-To Classes, Component Classes. With using classes, we can define design rules that are valid on a class of objects. For example match the length of all signals in a net class, or set differential pair phase tolerance on a class of differential pairs. We can manually create or view/edit classes in the PCB Editor's „Design“ Menu „Classes...“ item. This opens the „Object Class Explorer“ window. Here we can create new classes and select objects from a list to add to the new class. Classes can overlap, which means for example that one net can be a member of two net classes, then we can refer to the two net classes in two different design rules. This creates an opportunity for complex multi-level matching structures like DDRx memory clock-strobe-data 2-level matching.

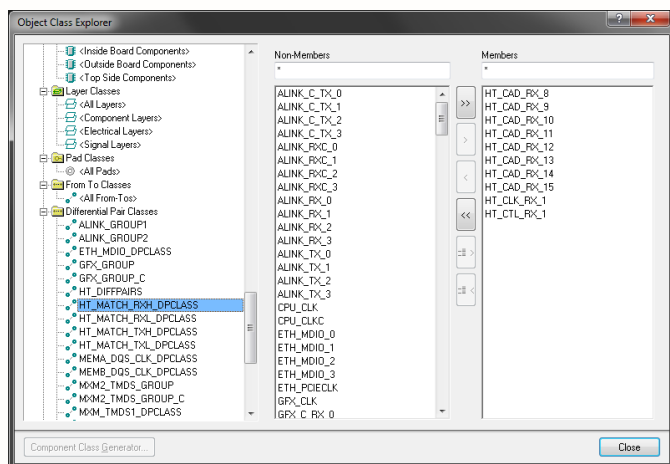


Figure 6: Object Class Explorer

Net Classes can be created in the schematics level as well. By simply using „Place > Directives > Net Class“, then attach the symbol to every net one-by-one that has to be in the Net Class, then doubleclick on the symbol and edit the „Name“ and the „Class Name“ fields. To make it work, the Schematics Editor's „Project > Project Options > Class Generation > Generate Net Classes“ menu item has to be enabled. Altium Designer 10 does not support Diffpair Class or From-To or From-To Class specification in the schematics level.

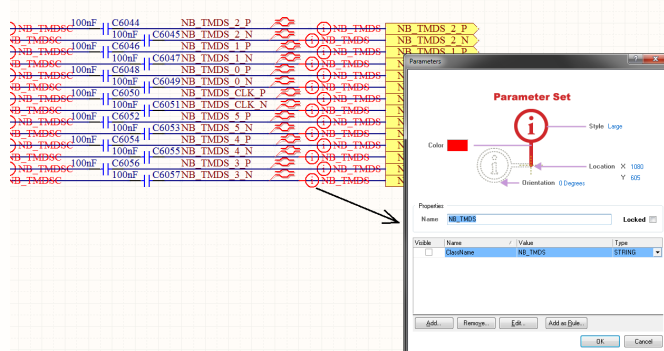


Figure 7: Net Class specification in schematics

If a series element breaks a high-speed net into two or more portions, then we have to specify net names, Net Classes, Differential Pairs, and Differential Pair Classes for all signals on both of the separate segments. Altium Designer doesn't treat these signal paths as single objects, like the Cadence Allegro would by creating „XNETs“. This way the objects and the trace length related design rules have to be created for both segments separately.

## Design Rules (Constraints)

In Altium Designer we normally specify PCB design rules or constraints in the Design Rules editor, in the „Design > Rules...“ menu item. For a PCB design, we have to set up various general design rules, but this document only describes the ones immediately related to high speed design considerations. There are several Altium Designer tutorials about using the other rules. Some design rules control interactive PCB editing parameters, others are checked (and objects marked if violated) by on-line DRC, while some other rules are checked by the manually-run Design Rule Check (DRC). For every rule type, we have to know how they are used/checked in the software. The trace width and spacing rules are based on manufacturing capabilities and impedance calculations. The length related rules can be based on a component manufacturer's (processor, FPGA, Graphics Processor...) design guide document, or based on a standard like the Jedec or PICMG standards, or they can be based on timing analysis or Signal Integrity simulations/calculations. Please refer to the article about this subject: „Generalized I/O Timing Analysis“ in the Printed Circuit Design and Fabrication Magazine.

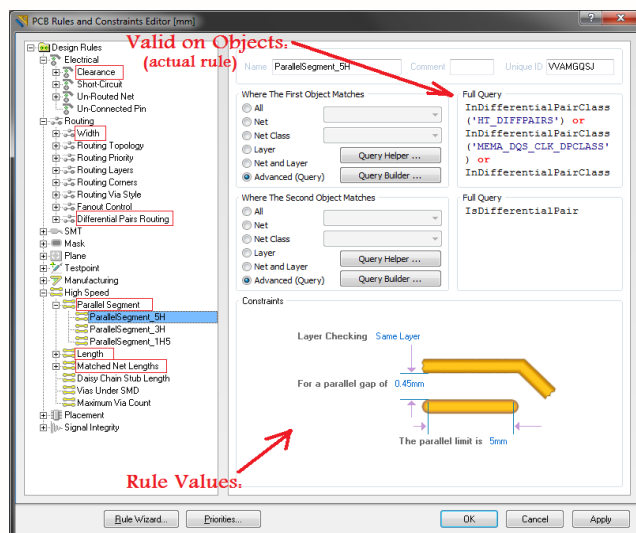


Figure 8: Design Rules Editor (the discussed rule types are framed with red)



With Altium Designer 10, we can specify PCB design rules in the schematics level as well, but a little bit differently than in the PCB editor. We add one PCB\_Layout directive (which can contain multiple rules) to a net, but the rules are without a „scope“. Instead of specifying the object to apply in the rule, the object is specified graphically by attaching the rule symbol to net or to multiple nets (copies of the same directive). We can place a rule directive from the „Place menu > Directives > PCB\_Layout“, then doubleclick on the symbol and doubleclick on the rule in the list, then click on the „Edit rule Values“ button. This opens a rule selector where we can create any PCB rule by doubleclicking on a category. We can add more rules to a directive symbol, by doubleclicking on the symbol on the schematics drawing, then press ADD, then type „Rule“ in the „Name“ field, press OK, then doubleclick on the new entry in the list and press the „Edit Rule Values“ button. Alternatively we can also do this by doubleclicking on the symbol on the schematics drawing, then press Add\_As\_Rule then click „Edit Rule Values“ then choose a rule from the tree-view. In this article we are using rules defined in the PCB Editor.

### Trace Width

Normally we put a group of nets based on characteristic impedance into a Net Class. Then we set up a „Width“ rule in the „Routing“ category for every net class separately, and also a default width rule for all other or non-impedance controlled traces. The most important use of Net Classes on high-speed boards is to group the signals for trace width specification, that will be used in interactive editing, autorouting and DRC.

### Spacing

Spacing in PCB design has two aspects: manufacturability and controlling crosstalk levels. For the first aspect, we set up a „Clearance“ rule in the „Electrical“ category, which is normally the minimum spacing that our PCB manufacturer recommends. The Differential Pair rules also contain a field called „Gap“, but this is related to the differential impedance of the diffpair.

In high-speed digital board designs we have to keep high-speed traces apart each other to minimize the crosstalk between them. Normally in the component/BGA breakout or fanout area or in small localized board areas next to vias or component we can not keep the desired separation. This way we can not set up a spacing rule based on the crosstalk-based recommendations, simply because the rule would be violated at too many places on the board with hundreds of DRC errors. The solution for the problem is to use the „Parallel Segment“ rule in the High-Speed category in the Design Rule Editor. This rule is a spacing rule which ignores short parallel segments, and only checks spacing if the two traces run too long in parallel. This is because the crosstalk does not manifest measurably on short parallel segments. The crosstalk noise voltage level is higher if the Aggressor and Victim tracks run parallel at a longer segment. To apply correctly, we should do signal integrity simulations with different spacing values, where the spacing to dielectric height ratio is what we are looking for. The actual space value does not guarantee crosstalk control, but this ratio does. The

dielectric height is the distance from a signal layer to the nearest ground plane layer. The rule has a parameter selector called „Layer Checking“. With this we can specify where the rule applies: „Same Layer“ only or „Adjacent Layers“ for dual stripline structures. Without the Adjacent Layers option the design engineer would have to check the spacing on adjacent layers manually.

The first rule is enforced in interactive editing and checked by the On-Line (real-time) DRC, while the second type is only checked by the manually-run DRC. Normally we apply both types of spacing rules on net classes. Sometimes we want to specify spacing (Parallel Segment rule) between differential pairs within a group, so we specify the rule's scope like „InDifferentialPairClass('HT\_DIFFPAIRS')“ for the „First Object“ and „IsDifferentialPair“ for the „Second Object“.

### Room

We can specify areas on the board where different rules would apply. These areas are called „Rooms“ and can be placed from the „Design > Rooms > Place\_Rectangular\_Room“ menu option. Then the Rooms can be named and Layer to-be-used-on specified, then referenced in the design rules (based on room name), so the rule will only apply within the area covered by the Room. This way we can for example specify that in the BGA breakout area all traces will be routed with minimal trace width, while outside the room the other width rules (for example the ones based on impedance / trace width calculations) will apply. The Room works with Trace width, spacing, manufacturing-related and many more Design Rules.

### Differential-Pair rule (spacing and uncoupled length)

The „Differential Pair Routing“ rule in the „Routing“ category has to be set up for every Differential-Pair Class separately or in groups (applied to a list of classes). The rule specifies the trace spacing („Gap“) between the positive and negative track within the differential-pair, and it is enforced in interactive editing when we use the Interactive Differential Pair routing option. It also specifies the maximum uncoupled length, but this one is only checked by the manually-run DRC. The result of the impedance calculations for a diffpair is a trace width and a spacing. The Differential-Pair rule only applies the spacing, so we need to create trace width rules as well, applied on the Differential-Pair Class or Net Class. We will also have to set up a „Matched Net Lengths“ rule to meet the phase tolerance requirements.

### Net Length Rule

With this rule, we can specify a minimum or maximum trace length for a group of signals (Net Class or Differential-Pair Class) or a group of From-Tos. We can find the „Length“ rule in the „High Speed“ category in the Design Rules Editor.

### Matched Net Length rule

With this rule we can control the trace lengths relative to each other in a group of signals, diffpairs or From-Tos. We can find the „Matched Net Lengths“ rule in the „High Speed“ category in the Design Rules Editor. An important parameter of the rule is the types of objects which between the rule will be applied. For

differential phase tolerance rules we enable only the „Check Nets Within Differential Pair” option, and leave the other two options disabled. If we want to match a group of nets like a lane on a memory interface, then we enable the other two options and disable the above mentioned one. If we want to match differential pairs to each other in a group, then we create a matching rule and enable only the „Check Between Differential Pairs” option. Don’t forget that for buses made of diffpairs, we must set up two separate matching rules for phase tolerance and group matching. Examples for differential buses (multi-lane or multi-pair) are x2-x16 PCI-Epress links, DVI, DMI, Hyper Transport buses and XAUI interfaces. Note that the „Tolerance” value in the rule is the maximum difference between the longest and shortest track, and not the +/-delta value from the average length.

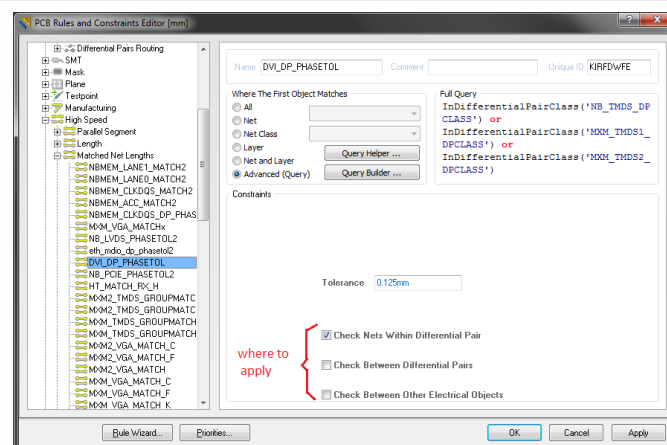


Figure 9: Matched Net Length Rule

### Stub Length rule

We can find the „Daisy Chain Stub Length” rule in the „High Speed” category in the Design Rules Editor.

What we need to setup:

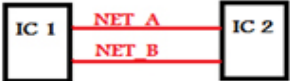
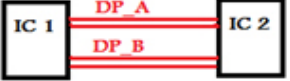

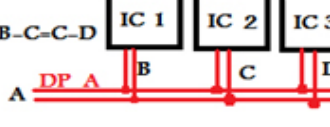
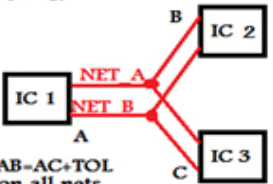
Case	Objects to create	Design Rules (on objects)
Trace Width	• Net Class or DP-Class	• Width rule (on Net Class)
Spacing	• Net Class or DP-Class	• Clearance rule (on Net Class(es)) • Parallel Segment rule (on Net/DP Class). If we have a multi-lane differential-pair-bus, then we have to set the rule based on DP-Class, to avoid overriding the DP P/N spacing from the DP rule.
Differential Pair	• Diffpair • DP Class • Net Class	• Diffpair rule (on DP Class) • Width rule (on Net Class) • Matched Net Lengths rule (Phase tolerance, on DP-Class. „Check Nets Within Differential Pair”=On, others=Off)
Min/Max Trace Length	• Net/DP/From-To list or Class	• Net Length Rule (on Net Class)
Match nets on Point-to-point bus 	• Net Class	• Matched Net Lengths rule (on Net Class) • Width rule (on Net Class)
Match diffpairs on a multi-lane Point-to-point bus 	• Diffpair • Net Class • Differential Pair Classes, one for each matching group	• Diffpair rule (on DP Class) • Width rule (on Net Class) • Matched Net Lengths rule (Phase tolerance, on DP-Class. „Check Nets Within Differential Pair”=On, others=Off) • Matched Net Lengths rule (Lane-to-lane, on DP-Class. „Check Nets Within Differential Pair”=Off, others=On)
Match Net Segment Lengths 	• From-Tos: every controlled segment on every net is a separate From-To. • From-To Class: Same segment on different nets. every controlled segment is a separate From-To Class.	• Matched Net Lengths rule (on From-To-Class). One separate rule for each controlled segment. • Length Rule (on all segment's From-Tos together). This sets min/max length for all segments.
Match Diffpairs on Segments 	• Diffpair objects (DP, DP Class, Net Class) • From-Tos: every controlled segment on every net is a separate From-To. • From-To Class: Same segment on different nets	• DP rules: DP, Width • Matched Net Lengths rule (Phase tolerance, on From-To-Class. „Check Nets Within Differential Pair”=On, others=Off). One separate rule for each controlled segment. • Matched Net Lengths rule (Group Match, on From-To-Class. „Check Nets Within Differential Pair”=Off, others=On). One separate rule for each controlled segment. • Length Rule (min/max) on all segment's From-Tos together).
Match Net Segments on balanced Tree topology  AB=AC+TOL on all nets	• From-Tos: From-Tos are defined in the Master device to each Slave direction, not between the Slaves. • From-To Class: All nets in group from the master to all slave devices is one From-To Class.	• Matched Net Lengths rule (on the one and only From-To-Class). • Length Rule (on all segment's From-Tos together). This sets min/max length for all master-to-slave connections.

Table 1: Objects and Rules

## Impedance rule

Altium Designer can set trace widths based on impedance specifications, but this uses built-in analytical equations, which is inaccurate for most real high-speed designs. This way we should really use an external field-solver-based impedance calculator program to calculate trace widths and differential pair separation. The Polar Instruments SI8000 or the freeware TNT-MMTL are suitable for this. Note that the two program's results deviate by 0-5 Ohms in some cases, and the Polar software is widely accepted in the industry. If we want to use AD's impedance calculator, then we just click „Characteristic Impedance Driven Width” in a „Width” design rule.

## Interactive editing

During interactive editing or routing we have to measure the trace lengths real time or quasi real time. The Altium Designer's on-screen Length Meter pops up during „Interactive Length tuning” or during „Interactive Differential Pair Length tuning” which shows net lengths in real-time. Unfortunately this only works on total net length, not on segment (From-To) lengths, and it does not pop up during routing or during sliding/stretching of trace segments.

The PCB Panel shows lengths in quasi real time, which means that we tune/slide/route then press ESC to exit the editing mode, then the measured length values get updated. If we match a group of nets or differential pairs, we want to see the length of the other traces in the group as well. The PCB Editor Panels, the PCB-Panel, the PCB-List-Panel and the PCB-Rules&Violations Panel can help in this. The last one was introduced recently, in earlier versions of Altium Designer this was a tab on the PCB-Panel, but now we can see both panels in the same time on the screen. It makes sense to make the used panels always visible so we can see them during editing. If we have more than one display then we can put these panels on the second screen as floating panels and use the main screen for the PCB drawing.

### Signal Routing and Length tuning (simple Point-to-point net length control)

We can control trace lengths by choosing routing path, deleting then re-routing segments, sliding segments or using the „Interactive Length tuning” or „Interactive Differential Pair Length tuning” features from the Tools menu.

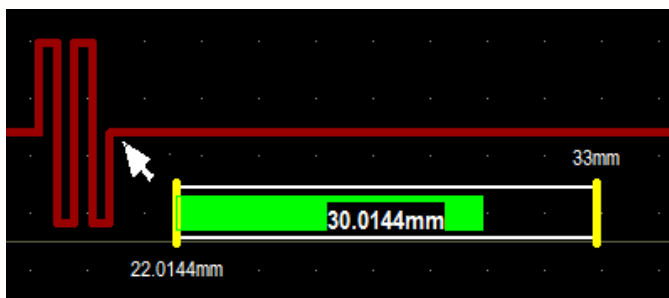


Figure 10: On-screen Length-Meter (Gauge) during „Tuning”

During Length Tuning modes the On-screen Length-Meter pops up on the screen to show net length in real time. The tuning parameters such as Reference Net selection, Rule selection, Meander parameters

can be set up during tuning modes when we hit the TAB-key.

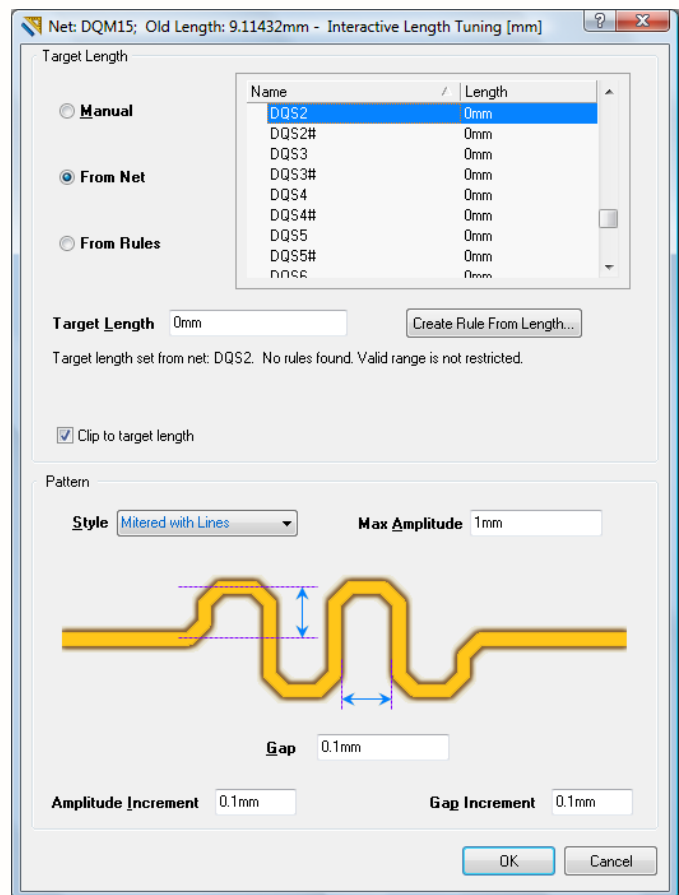


Figure 11: Tuning Parameters

Very often we want to control the trace length by re-routing or sliding segments. During this the Length meter doesn't pop up, so we have to measure the lengths using the PCB panel. This is not real time, so the way of using it is to route/slide/tune then exit routing mode (pressing ESC) then the length data on the PCB panel gets updated. When we are sliding a segment, the value gets updated after letting the trace go after sliding. If we have to adjust the length of a net then select „Nets” on the top of the panel, if we tune a differential pair then select „Differential Pairs Editor”, for Net Segment length control we select the „From-To Editor”. We can also hand route then run DRC then modify all tracks that are violating a matching rule.

In Altium Designer 10 the Rules&Violations panel lists all the rules and related violations with violation details (eg. deviation from preferred length). If we fix a violation then it disappears from the list. In earlier versions like Altium Designer 6.9 the Rules&Violation was a tab on the PCB Panel and it didn't display violation details, so we had to use it together with the PCB List panel to see the Violation details by setting: „Non-Masked objects” and „Include Only = Violations” on the top of the PCB List panel, then the details got updated as we started the trace editing.

Matching reference: By default, Altium Designer treats a matching group as a range from the shortest net to the longest net, and not as a reference-net +/-delta. During Tuning we can set a reference net (after hitting the TAB key), but otherwise there will be an automatic referencing based on actual length.

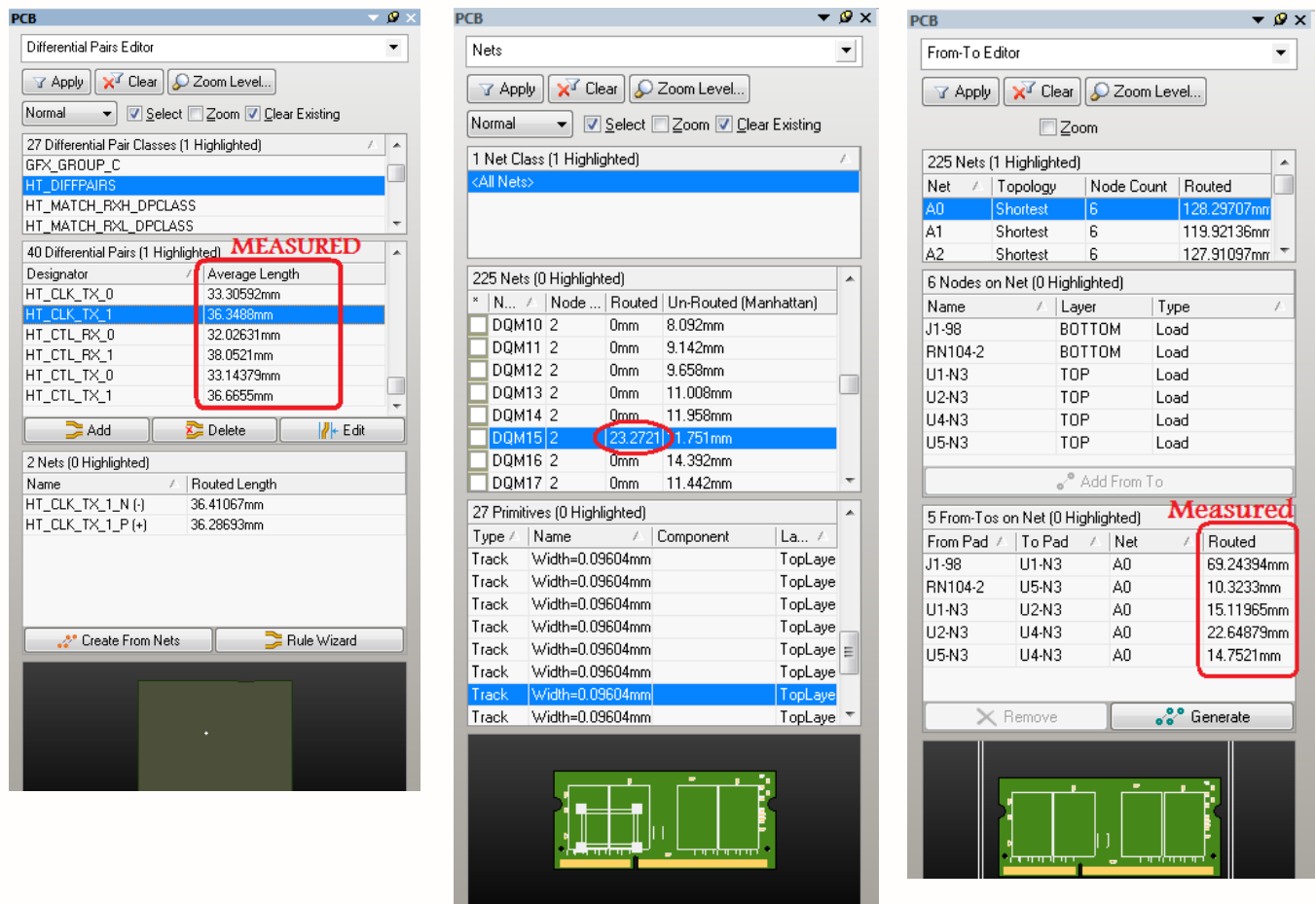


Figure 12-14: Measuring lengths on PCB Pane: Nets tab, Diffpairs Editor tab and From-To Editor tab.

The DRC also uses automatic referencing. This automatic referencing will be mentioned in the Rules&Violations list and on the PCB List Panel in the violation details. The referencing (eg. „...Differential Pair B Actual Length Difference against diff pair E is: 0.82...” in the violation details) looks similar to this: {A,B,C,D} → E, E → A.

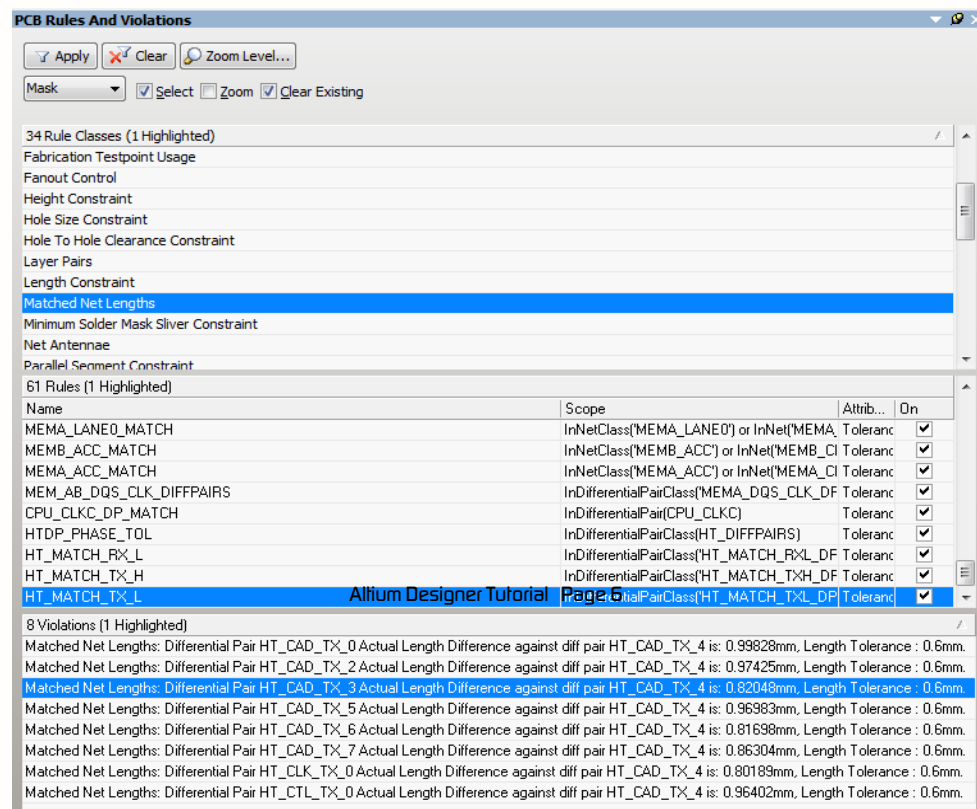


Figure 15: PCB Rules & Violations Panel.



### Point-to-point Net matching

After we set up the appropriate design objects and design rules, we can route all our tracks on a bus or interface. We should route the tracks with reasonable spacing to allow for space around the tracks for lengthening them. Finally we should run the DRC to update the violation list and fix the violations one by one. If we have a group of point to point nets to be matched then the violations will be on the tracks that are furthest away from the rest of the group. We can try to first shorten the longest tracks in the group. To do this, we select the net group (eg. Net Class) on the PCB Panel's Nets tab and then sort the member nets by length, then pick the few longest ones and try to change them to be shorter. After this we have to lengthen the shortest traces until all will be within range.

We can achieve this in different ways. The simplest is to hand calculate the minimum length limit based on the longest trace, then use the PCB Panel to select shorter tracks and lengthen them until all will be longer than the minimum. We can also use the Rules&Violations panel instead of the PCB Panel to select Violations and lengthen the tracks related to the violations until the violations disappear one-by-one, finally the list will be empty. Another way is to just try to lengthen all nets in the group by using „Interactive Length tuning“ and use the on-screen length meter to see when the meter's colour becomes green. The Interactive PCB Editor clamps the traces at the rule's maximum length. (if it's set to do so after hitting TAB) This way the tuning is fast and we don't need to verify values or do hand calculations, only watch for the colour of the Length Meter bar. Sometimes this doesn't work when the Length meter picks up the wrong minimum and maximum lengths (it might be a bug or a setting in a hard to find place), so then we have to use one of the first two methods.

### Diffpair Phase tolerance matching

The differential pairs need to be balanced, so the length of the negative net relative to the positive net of the same pair has to be matched. If the pair is unbalanced, then we get signal integrity problems, like eye-closure, increased common-mode noise due to „mode transformation“, increased crosstalk and increased EMI. We set up a matching rule where the „Check Nets Within Differential Pair“=On, others=Off. We have to route the differential pairs with the Interactive Differential Pair Routing option, although it might create a little bit unbalanced diffpairs. We have to fix the unbalance close to the trace-ends by re-routing, tuning or sliding segments of the shorter net and watching the result on one of the panels. The PCB Panel Diffpairs Editor tab shows the length of the nets in the DP separately, the Rules & Violations Panel (PCB List Panel in earlier Altium Designer versions) shows the violation details if any of the diffpairs are unbalanced.

We have to select the appropriate matching rule on the Rules & Violations Panel to see the related violations. The phase tolerance requirement is specified in processor datasheets, design guides or standards. For example for PCI-Express 1.1 we usually use +/-0.125mm phase tolerance.

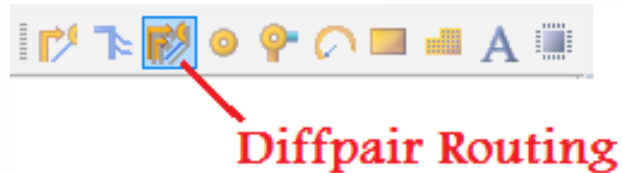


Figure 16: PCB Standard Menu

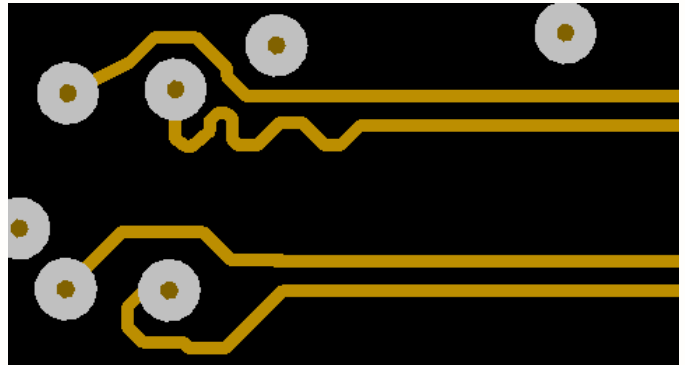


Figure 17: Manually balanced Diffpairs

### Diffpair-to-Diffpair matching

There is a need for diffpair to diffpair matching in multi-lane high-speed serial interfaces like PCI-Express, DVI, DMI, XAUI or Hyper-Transport. Some interfaces have built-in de-skew circuits so they can allow for a more loose matching (e.g. PCIe needs +/- 3° with some controllers), while others without this circuit need tighter matching (eg. Hyper Transport 1.0 needs +/-1mm lane-to-lane matching).

The design of these interfaces have to be done similarly to the net group matching. First we set up the objects then the rules (matching rule with „Check Nets Within Differential Pair“=Off, others=On.), then we hand route the diffpairs with Interactive Differential Pair Routing. After this we fix the unbalance within the pairs then we match the diffpairs to each other within the group. To measure the diffpair length we should use the PCB Panel's Differential Pairs Editor. We can adjust the length of a DP by deleting and rerouting a segment, by selecting the same segment on the P/N nets and slide them together, or by using the „Interactive Differential Pair Length Tuning“ in the Tools menu. The best way to check when they are matched is to select a diffpair class in the DP-Editor on the PCB Panel, sort the diffpairs in the list by length, then hand calculate the minimum length, then tune the nets that are shorter than the minimum length until they are longer than the minimum, but still shorter than the maximum.



Figure 18: Tuned diffpairs

### Net Segment length control

If we have to control the propagation delay from a particular pin to another particular pin on a multi-point net, then we have to create a design object describing that specific pin-pair. Sometimes we have



to control the length or delay on a daisy-chain or fly-by bus on the segments between the devices as well. These rules come from timing analysis or signal integrity analysis, other times it is standardized and described in a standard (like JESD21C) or in a processor's motherboard design guide. In the second case the timing and signal integrity analysis was done by those people who wrote the document, so the motherboard designer only has to follow the pre-created design rules. Typical examples are the different DIMM memory card designs, memory-down configurations, multiple-DIMM/channel designs, DDR1/2 DIMM-based designs, Compact-PCI card designs, multi-device PCI-X systems...

The net segments are represented by the From-To objects in Altium Designer. Normally we apply a design rule on a group („Class“) of objects, so we have to create the From-To Classes in the Object Class Explorer (Design>Classes) window. First we have to create the From-Tos for every affected net one by one in the PCB Panel's From-To Editor tab. When we set up the design rules, we apply them on From-To Classes, and during interactive editing we use the PCB Panel's From-To Editor to check the lengths of the pin-pair (From-To) connections. First we hand-route the tracks to follow the specified topology on all tracks in the group, then we tune the appropriate track segments while we check the From-To lengths on the PCB Panel. If we have absolute rules on the net group then it's easy to see the values on the PCB Panel, but if we have to match the segments then we have to run the DRC then use the Rules&Violations Panel to see if there are violations.

If we want to use the „Interactive Length Tuning“ feature, then we have to set a manual target length, like 99999mm which is surely longer than the net total length, since the Interactive tuning and on-screen Length Meter only measures a net's total length, they don't support From-Tos. We can also set a calculated target length: „actual\_total\_length + (required\_segment\_length - actual\_segment\_length)“ and tune until it clamps. Another problem is that the violation doesn't disappear from the list until we re-run the DRC, and the violation details also does not get updated until we re-run the DRC. Basically it doesn't work with On-Line DRC. This way nothing tells in real-time if the From-To length meets the rules or not. Basically the best method is to hand-calculate trace segment lengths and use the PCB Panel's From-To Editor to quasi-real-time measure the From-To lengths while comparing the measured lengths with our paper or Excel-based calculations. This might be very time consuming. It might be a good idea to first hand-route the board, then check all from-to lengths on the PCB Panel, then calculate absolute min/max length constraints for all segments, so we can easily compare the PCB panel's values with our Excel sheet during editing to see if the rules were met, without using automatic DRC. We can transform a matching rule into a min/max absolute rule by measuring the longest net in the group (that's the „max“), then set a rule for all From-Tos in the group. Since the DRC with min/max absolute length rules ignore From-Tos (bug in Altium Designer), there is not much point in setting these rules into the Altium Designer Design Rules Editor, so better to only handle them in an external Excel sheet. The matching rules/violations

on From-Tos also don't work very well, they don't work with on-line DRC but only with the normal DRC. In some older version of AD the creation of From-To Classes may not work. Instead of creating From-To Classes we can alternatively list all From-Tos in the Design Rules scope field (InFromto('name1') or InFromto('name2')...).

### Package Length

There isn't a really good support for this in Altium Designer. For large BGAs the signal routing lengths inside the package are usually not matched, each signal is routed at a different random length. The information is usually provided in an Excel spreadsheet or in a design guide or datasheet. The point in net length control is to ensure proper timing margins at the chip receiver circuits and NOT at the BGA balls, the signal arrival at the BGA ball has no significance. This way we need to control the die-pad to die-pad routing length by tuning the motherboard routing length. For this the design software has to take into account the in-package routing lengths as well. Other programs like the Cadence Allegro have this feature built-in as „Pin Delay“ for library components.

We can design a board by using matching rules only if the package length mismatch is smaller than the allowed length range on a particular bus. This way we have to tighten the design rules to compensate for the package length mismatch. If the mismatch is bigger than the allowed range of our original design rules, then we have to control every net length one-by-one, setting up Length rules for each net on the bus separately. This can mean a hundred rules for a bus. There is a way to speed this up a littlebit, by exporting rules into a text file from the Design Rules Editor, edit it with Notepad or Excel (copying rules and changing the referenced net in them), and then importing the file back into the Design Rules Editor. In the DRE we can right click on a rule category and select „Export Rules“ or „Import Rules“.

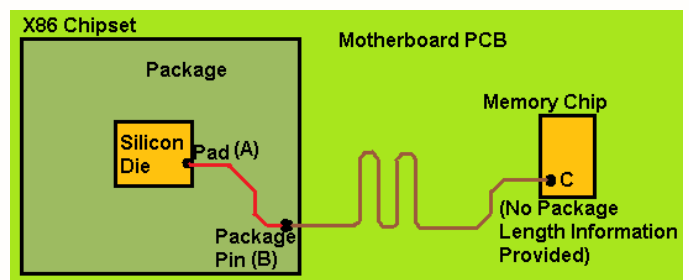


Figure 19: Package Length

### External Trace Length processing (Excel)

We can export trace lengths from the PCB Editor's PCB List Panel, or from the PCB Panel into Excel. The way to do it is to select/copy/paste like we do between two Excel sheets. This might be useful when we want to check the trace lengths on a bus against a processor manufacturer's trace length calculator. Intel, AMD and others usually provide a Trace Length Calculator for their processors and chipsets.

Unfortunately the opposite way, importing design rules from Excel into the PCB List Panel does not work. Even changing the rules values on the List panel doesn't work. This might be a bug in the software, or just not supported. We can import rules into the Design rules editor from a file, but the text

in the file has to be in a very specific format. If we create a rule in Altium Designer DRE and export into a text file, then edit the file in a text editor (or in Excel) then import it back into the DRE, then we can have lots of similar rules quickly created or updated with values.

Exporting From-To details also doesn't work properly, since from the PCB Panel we can only select/copy the From-Tos that belong to one net, and on the PCB List Panel the From-To routed lengths are not displayed. Also the net report does not contain From-To details.

## Typical Examples

### Differential Hyper Transport bus design

Embedded clock interfaces (like PCI-express or DVI) usually have De-Skew circuits built-in, so they only need loose matching, while for example clock forwarding interfaces (e.g. Hyper Transport 1.0 or XGMII) need tight matching between diffpairs.

In a DC-coupled x16 Hyper Transport link between an AMD Athlon-II processor and the AMD north bridge, we have four matching groups. Half link in one direction is one group, so we have groups named TX\_upper, TX\_lower, RX\_upper and RX\_lower. Within each group we need to match the diffpairs within +/-0.6mm. For this, we have to create four diffpair classes, and one separate Matched Net Lengths rule for each diffpair class. We need one more Matched Net Lengths rule which is applied on all the four classes, and it sets the phase tolerance matching at +/-0.125mm with „Check Nets Within Differential Pair“=On, others=Off.

Steps:

- ▶ Create diffpair classes, one for each matching group. The matching groups are specified in the processor's design guide or in the datasheet.
- ▶ Create rules (diffpair rules, width rules, matching rules for phase tolerance and group matching).
- ▶ Route diffpairs with Interactive Differential Pair routing.
- ▶ Run DRC.
- ▶ Verify if all the rules are working, if not then re-create the rules. Normally there will be a few violations on every rule, so if there aren't any then it's suspicious that the rule is not working.
- ▶ Match P/N nets for phase tolerance using the Rules&Violations panel.
- ▶ Match diffpairs within groups using the PCB Panel's Diffpairs Editor.
- ▶ Run DRC again, check if the phase tolerance is still matched, if not then fix them again.

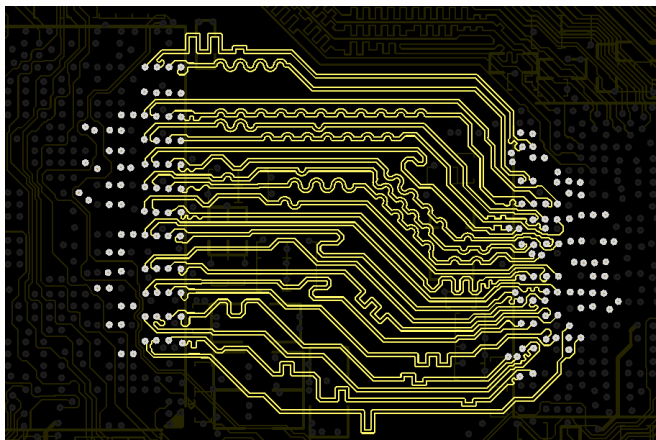


Figure 20: Hyper Transport Bus (one of the 3 used layers)

### DDR3 SODIMM-based motherboard design

For a DIMM-based memory interface, we have to create lots of separate matching groups, route both differential and single-ended signals. Normally the original design rules or constraints are specified in the processor or chipset's datasheet or motherboard design guide. We have to translate those constraints into Altium Designer design rules. The chip manufacturers usually define a few design rules: DQ/DM to DQS matching, ACC to CLK matching and DQS to CLK matching. These are specified for example as  $L_{dqs} = L_{clk} \pm \Delta$ , or maybe asymmetrically as  $\{L_{clk} - \Delta_1 \dots L_{clk} + \Delta_2\}$ . In this section there is a step by step method for the memory channel design, although other methods could also be suitable.

Steps for one 64 or 72-bit memory channel:

- ▶ Set up design objects: Net classes in schematics, one net class per lane in data bus (8 DQ signals and one DM signal), one net class for ACC bus, one net class for clocks one for strobes (DQS). Set up diffpair classes in PCB for strobes, and other class for clocks.
- ▶ Fan out the DIMM and the processor to dogbone vias.
- ▶ Route tracks in parallel until close to the processor. Try to use one layer.
- ▶ Route tracks from the processor's fanout dogbone vias to the routed tracks. To see clearly, set „View>Connections>Hide\_All“, then „View>Connections>Show\_Component\_Nets“ then click on the DIMM. Probably we will find that we can not connect all of them, so try to connect as many as we can, normally around half of them. The signals that we couldn't connect have to be moved to another layer, then we have to be able to connect all. If we still can't succeed then we have to swap a few signals between the two layers. We should also distribute the traces on the used layers evenly to maximize spacing between them. Now all signals are routed between the DIMM socket and the processor.
- ▶ Set up design rules:
  - One Matched Net Lengths rule for each lane on the databus (applied on: LaneK\_CLASS, DQSk\_P, and „Check Nets Within Differential Pair“=Off, others=On), One lane's DQ and DM signal are in one Net Class, specified in the schematics.
  - One Matched Net Lengths rule for the clock diffpairs phase tolerance („Check Nets Within Differential Pair“=On, others=Off). All the clocks are in a diffpair Class, so the rule is applied on the class.
  - One Matched Net Lengths rule for the strobe diffpairs phase tolerance („Check Nets Within Differential Pair“=On, others=Off). All the strobes are in a diffpair Class, so the rule is applied on the class.
  - One absolute Length rule for the Address/Command/Control (ACC) bus. These signals are in one Net Class specified in the schematics. The rule is applied on: ACC\_CLASS. The min/max rule values will be calculated later.
  - One absolute Length rule for the DQS-CLK matching. The rule is applied on: DQS\_CLASS. The min/max rule values will be calculated later.
- ▶ Display a list of all signals on the channel in the PCB Panel's Nets tab, select all related Net Classes, then sort them based on length. Find the few longest ones and try to re-route them to be shorter. Some processor datasheets specify a maximum memory bus length, so check if all our signals are shorter than that. (Those processors have a „simple“ read data capture circuit.)
- ▶ In the list of nets note the length of the longest and the shortest net ( $L_{max}$  and  $L_{min}$ , and separately for the ACC bus as well). Choose the length of the clock signals based on three equations:  $L_{clk} \geq 1.02 * L_{max} - \Delta_{L\_clk\_dqs}$ , and  $L_{clk} \geq 1.02 * L_{max\_ACC} - \Delta_{L\_clk\_acc}$ , and finally  $L_{clk} \geq \text{already\_routed\_clock\_length}$ .
- ▶ Tune both clock diffpairs to the chosen length within the given CLK-to-CLK tolerance (or just +/-0.1mm). Use Interactive Differential Pair Length Tuning, hit TAB key and set manual target length. Also make sure that the phase tolerance matching rule is satisfied. After this, make sure that the clock traces are not modified anymore.

- ▶ Set a value for the ACC bus absolute Length design rule:  $L_{min} = L_{clk} - \delta L_{clkacc}$ ,  $L_{max} = L_{clk} + \delta L_{clkacc}$ .
- ▶ Set a value for the DQS absolute Length design rule:  $L_{min} = L_{clk} - \delta L_{clk dqs}$ ,  $L_{max} = L_{clk} + \delta L_{clk dqs}$ .
- ▶ Match DQS signals for phase tolerance.
- ▶ Match DQS signals with CLK by re-routing segments and using Interactive Differential Pair Length Tuning and using the absolute Length rule. During tuning hit the TAB key to set reference length to be the CLK0\_P net. We can have the PCB Panels Diffpair Editor on the screen, CLK and DQS diffpair classes enabled to manually oversee and verify net lengths how far/close they are to the calculated ranges. Tune DQS until they are longer than  $L_{min}$ .
- ▶ Set DQS length more accurately for every lane, by selecting the lane's nets (the net class of DQ/DQS) sort them by length, and set  $L_{dqs} = L_{max} - \delta L_{dq dqs}$ . Lengthen DQS to this new length. Do this for each lane separately.
- ▶ Check if the DQS and CLK phase tolerance rules are violated or not. Fix if needed.
- ▶ Lengthen the shorter tracks in every lane by re-routing segments or using Interactive Length Tuning. During tuning hit the TAB key to set reference length (the appropriate DQS pair's positive net). The lengthening stops when we reach the right range, then proceed to the next signal in the lane. To easily find the related signals, use the PCB panel's Nets tab and select the appropriate Net Class of the actual lane, and enable „Dim“ or „Select“ on the header of the PCB Panel. On the PCB Panels Nets tab we can manually verify net lengths how far/close they are to the calculated ranges.
- ▶ Tune the ACC bus similarly to the data bus lanes, but set the reference signal to be CLK0\_P.
- ▶ Run the DRC to check if the whole bus is tuned, and also verify if the rules were working, for example by deleting a long segment from a track should create a violation in a matching rule.

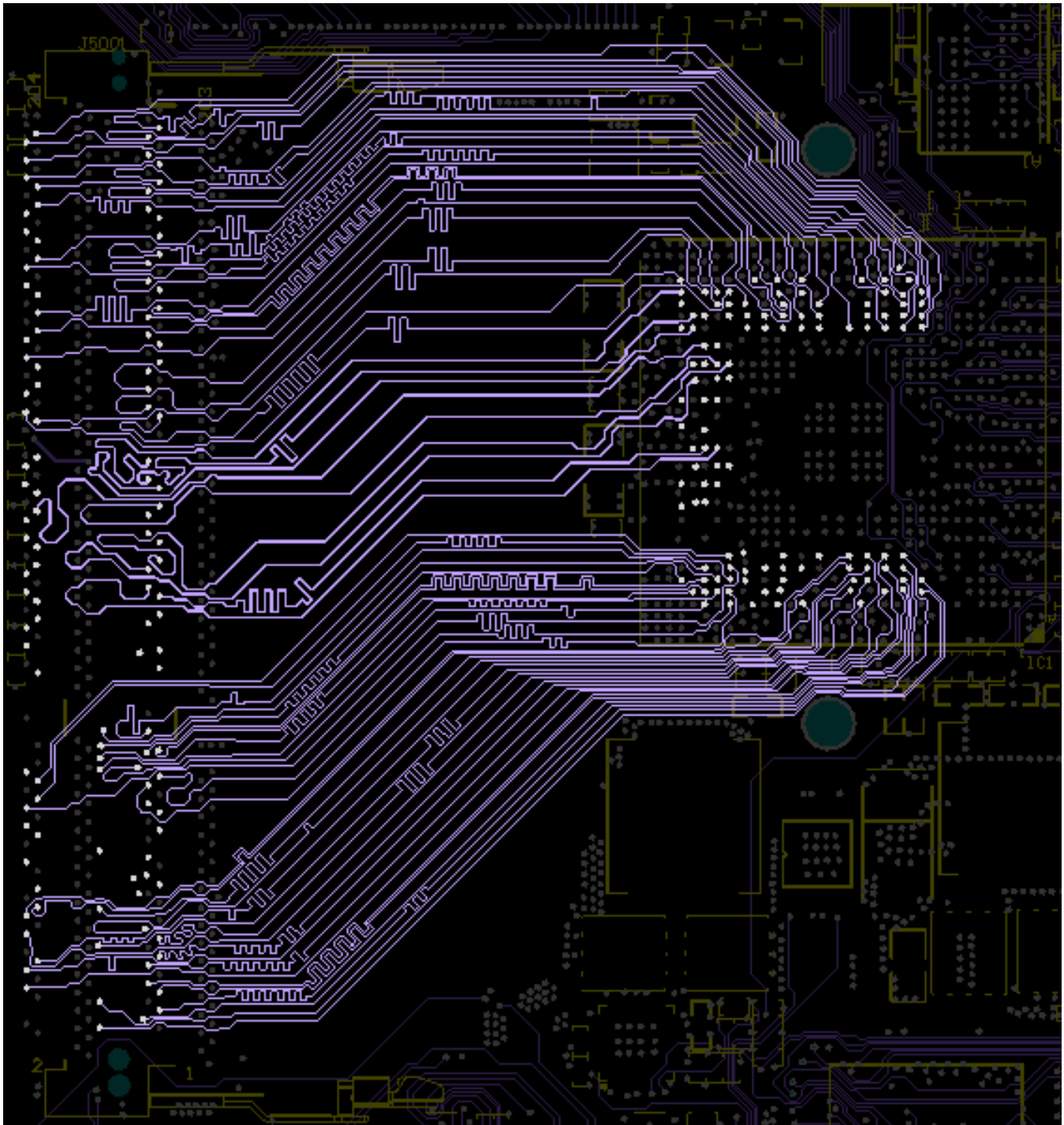


Figure 21:DDR3 SODIMM socket routed to a processor (one of the 2 used layers, 2/channel)



## DDR3 Memory single-chip Interface design

This is basically the same as the DIMM-based memory interface design.

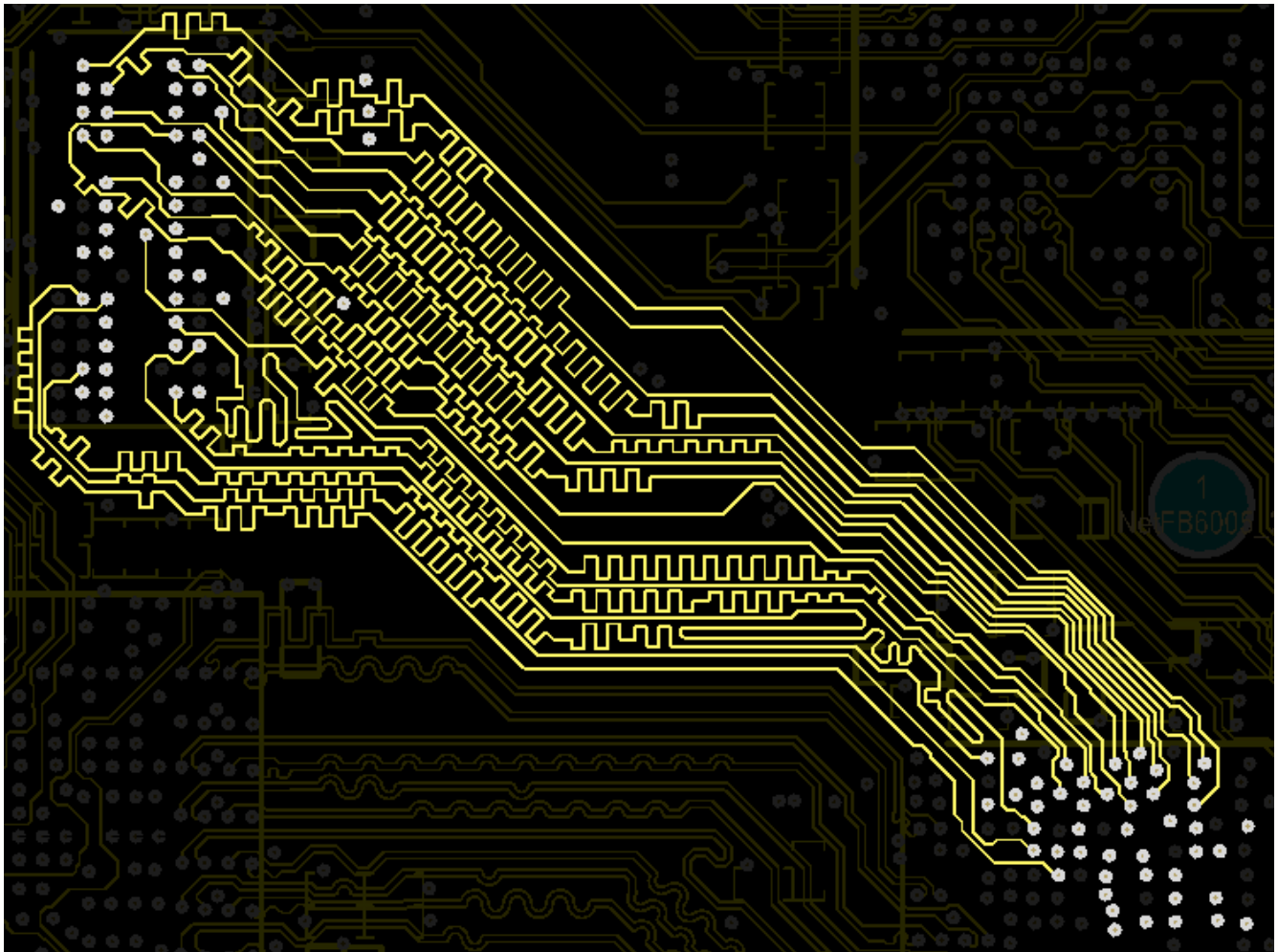


Figure 22: Single-chip memory interface (one of the 3 used signal layers)

### DDR3 SODIMM memory card design

If we have more than one memory chip, then the address bus is not point-to-point anymore and it requires From-To based design rules. In this example we are focusing on the Address/Command bus design on a DDR3 SODIMM memory modules, based on the JESD21C 4.20.18 document, using the Raw-Card-C configuration. The Jedec standard specifies track segment lengths with both absolute and matching rules. The data bus (DQ, DM, DQS) routing can be done in the same way as a cpu-to-DIMM motherboard design, but the absolute length rules also have to be satisfied. The CLK to DQS matching is indirectly specified in the Jedec specifications, and its different than the a cpu-to-DIMM rules: they specify absolute length rules for all signal groups. On DDR3 DIMMs the DQS/DQ/DM length is a lot shorter than the CLK/ACC signal lengths to any of the DRAM chips. Since DDR3 uses „Fly-By” topology, the CLK/ACC length to the different DRAM chips (Fro-To) is different. On DDR1/2 designs they used balanced tree topology where the CLK/ACC lengths to the different DRAM chips were matched.

The Jedec standards specify net segmenths that we have to translate into From-To design rules, for example in the figure-24 the From-Tos are:

TL0+TL1+TL2 (card-edge to first DRAM chip),  
TL2+TL3+TL2 (first chip to second), TL2+TL4+TL2,  
TL2+TL5+TL2, TL2+TL3+TL4+TL5+TL2 (first chip to last), and TL2+TL6+TL7 (last chip to termination).

A simple way of designing DIMMs or Memory-Down is when we copy the placement and the routing from the Jedec reference design. These reference designs are normally provided on the Jedec website for free, but in Cadence Allegro format. We have to import the Allgero file into Altium Designer on a temporary board file, then copy/paste the routed tracks onto our board, but the components have to be there already in the right locations to allow the tracks to pick up the correct nets.

Signal Group	Matching Rules
CLK to CLK# Matching	Match TLx segment by TLx to segment to within 0.1 mm
CLK Pair to CLK Pair Matching (Pair to Pair - Avg. Length)	Match total compensated length from connector to each DRAM to within 0.25 mm
CTRL Group Matching	Match total compensated length from connector to each DRAM to within 1.0 mm
CTRL to CLK Matching	Match total compensated length of all CTRL signals from connector to each DRAM to within CLK +/- 0.5 mm
ADR/CMD Matching	Match total compensated length from connector to each DRAM to within 1.0 mm
ADR/CMD to CLK Matching	Match total compensated length of all ADR/CMD signals from connector to each DRAM to within CLK +/- 0.5 mm
TL2 Stub Length Matching	Match TL2 stub length at each DRAM (top and bottom), on a given signal, to within 0.5 mm
TL2 MAX Stub Length Limits	CLK : TL2 <= 3.0 mm CTR: TL2 <= 4.0 mm ADR/CMD: TL2 <= 5.6 mm
CLK First to Last Length	The maximum length from the first DRAM and the last DRAM = 166 mm
Neckdown Length	5.0 mm <= length <= 10.0 mm Match to within +/- 1.0 mm

**Note 1** All length matching is done using velocity compensated stripline equivalent lengths.  
**Note 2** The velocity compensation ratio of 1.1 will be used (MS length / 1.1 = SL equivalent length)  
**Note 3** The neckdown length is the trailing portion of the lead-in section, which is routed at 0.1 mm width  
**Note 4** Maximum first to last length can be calculated by subtracting length to the first DRAM from the length to the last DRAM  
**Note 5** Via compensation is not required  
**Note 6** All clocks are length matched, segment by segment and have matched via count

Figure 23: Jedec DDR3 SODIMM general routing guidelines

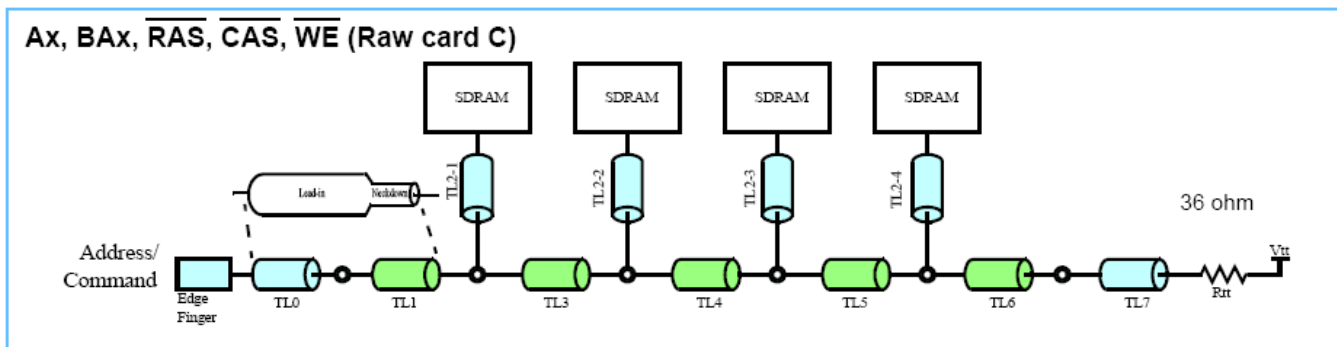


Figure 22 — Net Structure Routing for Address and Command Raw Card C

Table 29 — Trace Lengths for Address and Command Net Structures Raw Card C

	TL0 Micro- Strip	TL1 Strip	TL2 Micro- Strip	TL3 Strip	TL4 Strip	TL5 Strip	TL6 Strip	TL7 Micro- Strip	Neck- Down Length	Compens- ated TL0+TL1 +TL2-1	Compens- ated TL0+TL1 +TL3+TL 2-2	Compens- ated TL0+TL1 +TL3+TL 4+TL2-3	Compens- ated TL0+TL1 +TL3+TL 4+TL5+T L2-4	Notes
MIN	25.0	23.3	0.5	13.7	22.2	13.8	2.7	0.9	6.9	74.6	88.5	110.8	124.7	1, 2, 3
MAX	55.8	51.3	2.0	14.0	22.3	14.0	15.2	1.2	7.1	74.6	88.5	110.8	124.7	

**Note 1** All distances are given in millimeters.

**Note 2** Recommended Trace-Trace spacing : min0.2 (Lead-in section), min0.2 (Loaded section, between vias is min 0.1)

**Note 3** In the column of compensated length, Micro-strip line length is compensated as Equivalent Stripline Length (= Microstrip Length / 1.1), e.g.: (compensated TL0+TL1+TL2-1) = TL0/1.1 + TL1 + TL2-1/1.1

Figure 24: Jedec DDR3 SODIMM Address / Command routing guidelines (abs. rules)

We will use manual absolute rule checking because of the bugs in Altium Designer related to From-To-based rule support. We can set up matching rules for the DRC-based verification, since the DRC still works with From-To-based Matched Net Length rules.

To easily see which nets to adjust, we should have them in one net class, and highlight (dim) them through the PCB Panel's Nets tab. Although this doesn't help us in finding them in the net list on the From-To Editor. We can also highlight the actually edited net through the From-To Editor, by selecting the net in the net list, then right click and click on „Zoom Selected“. After this, the editing of the tracks is disabled, so we have to press clear, then find and highlight the same net on the graphical editor using CTRL+click. Because of the no-support for From-Tos in the On-Screen Length-Meter, we have to do hand calculations to convert the From-To adjustment to total net length adjustment.

#### Steps:

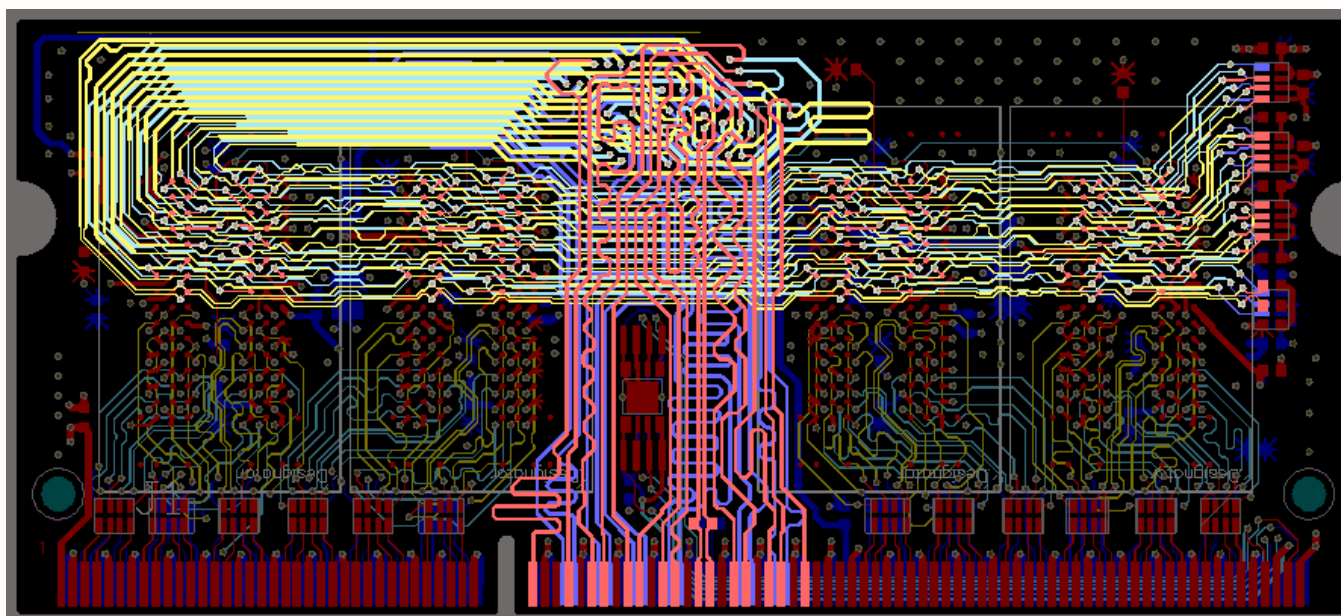
- ▶ Hand-route the bus and follow the specified topology. Check the Jecdec reference board design (Allegro) file for the topology, placement, fanout, routing and layer usage. A free Cadence Allegro Viewer can be used for this. Make sure to break the track segments at every branch to avoid From-To length measuring errors (see the chapter about Bugs and Workarounds).
- ▶ Set up the From-Tos in the PCB Panel's From-To Editor, for all specified pin-pairs on all nets in the group. These are the DIMM card edge to first DRAM chip's pin, the segments between the DRAMs, from the last DRAM to the termination, and from the first DRAM until the last DRAM (clock only).
- ▶ Check the routed From-To lengths (one-by-one on the PCB Panel), if they are longer than the Jecdec absolute trace length specifications. If they are longer, then try to re-route them to be shorter. If we can not shorten all tracks, then we have to set a new target length for the ACC bus. For a DDR3 memory address bus we can increase a few segment (card edge to first DRAM, last DRAM to Rtt) lengths by a few millimeters. This will deviate from the standard, but if we check the different Raw Card topologies, then we would see that there are big differences in these segment lengths. The important thing is to match the different nets on the same segment, so use a longer target length but for all signals in the group. If we increase the ACC-bus (from card edge to first DRAM From-To) and the data bus by the same amount, then the effect is like if the DIMM socket on the motherboard was further away from the processor/chipset, which is completely allowed by the

processors/chipsets. This might have signal integrity effects, although the static timing will be correct.

- ▶ Also check if the From-To were measured correctly by the program or not. There is a bug in the program that needs this workaround. The net total length should be around the same as the sum of the From-To lengths (stubs are measured twice). If it isn't, then we have to re-route segments until this is fixed. See the section about bugs and workarounds.
- ▶ For each net one-by-one tune (lengthen) the routed segments to be within the Jecdec absolute length requirements, while measuring the length on the PCB Panel's From-To Editor. Since the PCB Panel's length display is not really real-time, we have to use the On-Screen Length Meter in the following way: Check how much longer (extra) the From-To has to be (extra = desired - actual), then check initial total net length (in the first list on the From-To Editor), hand calculate the „initial\_total\_net\_length + extra = new\_total\_net\_length“, then use the Interactive Length Tuning (set manual target length to the calculated value by hitting TAB) to tune the appropriate segment until we get the calculated value for the total net length. After this we verify that whether the From-To length is as it was specified in the Jecdec specs or not. We can select a net on the From-To Editor to be able to find it on the board, but it only selects the pins, not the routed tracks, so we have to press the „Clear“ button in the lower right corner to enable the access to the tracks.
- ▶ Set up From-To Classes and Matched Net Length rules based on the general routing guidelines.
- ▶ Run the DRC to see if all the From-Tos are properly matched.

#### DDR3 Memory-Down design

The „Memory-Down“ is the design technique where we design a complete DIMM memory on to the motherboard, so we don't need to use DIMM sockets, all the memory chips will be soldered on the motherboard. In a Memory-Down configuration, we basically attach the design rules of a processor-to-DIMM design guide with the DIMM design guide from the Jecdec specifications. The rules are given in a format like offset\_from\_reference+/-delta (relative) or offset+/-delta (absolute) or min/max range (absolute) format. The motherboard design guides normally provide relative rules, where the length of a signal group is specified relative to another signal or group (eg. ACC-bus to CLK), while the DIMM design rules are provided as absolute rules in min/max ranges. We have to transform both constraints to offset +/-delta description, then sum the offsets and the deltas separately. This way we get constraints from chip/die pin to chip/die pin. Because the Altium



**Figure 25: JEDEC standard-based DDR3 SODIMM (Routed on four signal layers, ACC/CLK highlighted. We can see that the ACC-bus enters the card in the middle of the edge-connector, then routed up, then changes layer and routed to the left, then routed to the first DRAM chip, then to the second, to the third, to the last DRAM chip, then finally to the termination resistors. This way the Jecdec standard net topology and trace segmenth length rules can be satisfied by length tuning.)**



Designer's From-To support has some issues, we work with min/max design rules for every From-To (see above). This way we have to convert the summed constraints into min/max range absolute rules. If we copy a Jedec DIMM layout into our motherboard design, then we only have to set up the From-Tos on the ACC/CLK from the processor pins until the first DRAM chip on the chain, since the other segments will not change and they are properly designed by the Jedec members. We have to calculate the CLK-to-DQS matching as well.

Steps:

- ▶ Convert the Jedec ACC-bus trace lengths into Offset +/- Delta rules (Figure-24):
  - $\text{Offset} = (\text{length\_min} + \text{length\_max}) / 2$
  - $\text{Offset\_from\_reference} = \text{Offset} - \text{offset\_of\_reference\_signal}$
  - $\text{Delta} = \text{length\_max} - \text{Offset}$
  - The reference signal for the ACC bus is the CLK0 diffpair's positive net.
- ▶ Sum the motherboard design guide's constraints with the Jedec constraints. For the ACC bus (or separately for addr/comm, and control bus):
  - $\text{Offset\_from\_reference} = \text{Offset\_from\_reference\_Jedec} + \text{Offset\_from\_reference\_CPU}$
  - $\text{Delta} = \text{Delta\_Jedec} + \text{Delta\_CPU}$ .
- ▶ Hand route the whole bus based on the specified topology drawing, then measure CPU-to-first DRAM chip From-To length on all signals on the ACC bus and clocks. Take the longest one and calculate new\_offset = length\_max\_routed - Delta\_ACC. Calculate absolute length rules for the ACC bus (or separately for addr/comm, and control bus):  $\text{Length\_min} = \text{new\_offset} + \text{Offset\_from\_reference} - \text{Delta}$   $\text{Length\_max} = \text{new\_offset} + \text{Offset\_from\_reference} + \text{Delta}$
- ▶ Manually check the From-To lengths against the calculated rules on the PCB Panel's From-To Editor during routing, as it was described in the above section about designing a DIMM memory card.
- ▶ Set up an absolute length design rule for the databus based on the processor's matching rules (+/-delta\_L\_clkdqs\_cpu) and the Jedec CLK0\_P(to first DRAM)/DQS(to first DRAM) length difference ( $\text{offset\_L\_clkdqs\_jedec} = \text{L\_clk0\_P} - \text{L\_dqs0\_P}$ ). The DQS absolute Length design rule:  $\text{Length\_min} = \text{L\_clk0\_P} - \text{offset\_L\_clkdqs\_jedec} - \text{delta\_L\_clkdqs\_cpu}$   $\text{Length\_max} = \text{L\_clk0\_P} - \text{offset\_L\_clkdqs\_jedec} + \text{delta\_L\_clkdqs\_cpu}$
- ▶ Tune/design the databus in the same way as we did for the DIMM-based motherboard design. The only difference now is the way of specifying the DQS length range relative to the CLK length, in the previous step. Remember, we have to set up matching rules for every lane on the databus, measure lengths in groups, tune DQS to longest DQ then tune other DQ/DM to DQS...

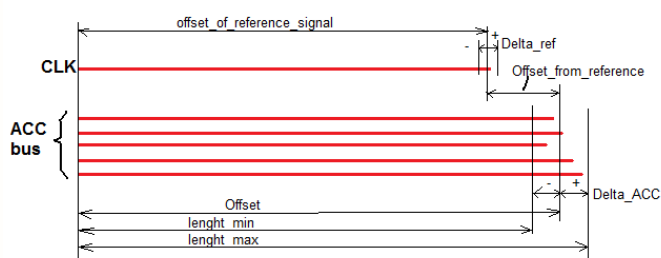


Figure 26: Relative and absolute length rules

If we need to connect two memory chips to a DSP or to a graphics processor, then we can do it similarly to a 64-bit Memory-Down design, but simply cut off the not needed part of the DIMM topology. This means for example for a 32bit interface, that we cut off the second half of the x16 DRAM chips (seen on the Fly-By chain) and the track segments between them.

## Bugs and workarounds

1.) The Violation list on the Design rules&Violations panel (previously on the PCB Panel's Rules&Violations

tab) does not get updated in case of diffpair to diffpair matching rules, or in case of From-To based matching rules. It needs us to run the DRC. It gets updated only during net to net matching. Fortunately the PCB Panels's diffpair length and From-To length display gets updated, so we can use that to check if the routing already meets the requirement or not. The only downside is that we have to see if the numbers are within range, we may have to note numbers on a piece of paper and use a calculator to calculate the min/max range of the particular net, the program will not tell in real-time if the matching rule is met or not.

2.) Sometimes the DRC and the On-Line DRC ignore some matching rules or it ignores few of the objects in the rule. Because of this bug, we have to make sure that every rule is „alive“, for example by deleting a trace segment and checking if a new violation shows up. If it doesn't violate the matching rule, then we have to delete the rule and create a new one with a new name.

3.) Sometimes the Length Meter displays a minimum and maximum limit that are incorrect and unrelated to the actual trace lengths or rules. This may because it picks up a wrong target length. The best thing is to make sure by hand calculation about the minimum and maximum limits for every trace and ignore the Length Meter's displayed limits. We might have to set manual target length using TAB. We can also hand calculate min/max lengths then set a rule for the range, finally set a fixed target length (in the middle of the calculated range) during length tuning. If there are two matching rules, then sometimes the Length Meter picks up only the first rule.

4.) Lack of „Package Length“ support in Altium Designer: We can design a board by using matching rules only if the package length mismatch is smaller than the allowed length range on a particular bus. This way we have to tighten the design rules to compensate for the package length mismatch. If the package mismatch is bigger than the allowed range of our original design rules, then we have to control every net length one-by-one, setting up Length rules for each net on the bus separately. This can mean a hundred rules for a bus. There is a way to speed this up a littlebit, by exporting rules into a text file from the Design Rules Editor, edit it with Notepad or Excel copying rules and changing the referenced net in them, and then importing it pack into the Design Rules Editor. In the DRE we can right click on a rule category and select „Export Rules“ or „Import Rules“.

5.) Displaying From-To Length: It works only on the PCB Panel, it doesn't work on the On-Screen Length Meter or on the Heads-up Display. This way we have to use the PCB Panel during editing. If we want to use the „Interactive Length Tuning“ feature with From-To-based rules, then we have to set a manual target length, like 99999mm which is longer then the net total length, since the Interactive tuning and on-screen Length Meter only measures a net's total length, they don't support From-Tos. The Interactive tuning and on-screen Length Meter only measures a net's total length, they don't support From-Tos. Another problem is that the violation doesn't disappear from the list until we re-run the DRC, and the violation details also does not get updated until we re-run the DRC. This way nothing tells in real-time

if the From-To length meets the rules or not. Basically the best method is to hand-calculate trace segment lengths and use the PCB Panel to quasi-real-time measure the From-To lengths while comparing the measured lengths with our paper or Excel-based calculations. This might be very time consuming.

6.) The DRC sometimes doesn't find any violations on absolute Length Rules applied on From-Tos. If we need to use absolute length rules, then we have to check violations manually using the PCB Panels' From-To Editor and an Excel sheet or a piece of paper.

7.) The From-To Editor sometimes displays an incorrect length for the From-Tos. For example it displays correctly for most of the nets, but for a few ones the From-To lengths seem to be longer than they should be. Normally for these nets the sum of all From-To lengths is a lot longer than the net total length which is a good way to finding these nets. The cause of this error is that Altium Designer is not able to measure partial track segment lengths. To make sure that Altium Designer will measure correctly, we should route the tracks in a way that the tracks break at every branch. See the drawing below. If this still doesn't fix the length problem, then we have to delete then re-route a segment or the whole net manually. Sometimes by simply re-routing a 5mm segment, the measured From-To length changes by 50mm. This is another bug related to the From-To length measurements in Altium Designer.

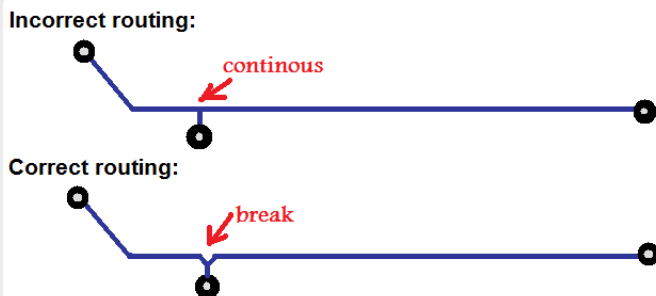


Figure 27: Routing Effect on From-To measuring correctness

## Non-existing features

For real high-speed designs, a few important features should be implemented in Altium Designer:

1) Package length support for large BGA devices like PC chipsets and large FPGAs. This one is the most important missing feature. For large BGAs the signal routing lengths in the package are usually not matched, each are routed at a different random length. The information is usually provided in an Excel spreadsheet or in a design guide or datasheet. The point in net length control is to ensure proper timing margins at the chip receiver circuits and NOT at the package/BGA balls, the signal arrival at the BGA ball has no significance. This way we need to control the die-pad to die-pad routing length by tuning the motherboard routing length. For this the design software has to take the in-package routing lengths as well. Other programs like the Cadence Allegro have this feature built-in as „Pin Delay“ for

library components. In Altium Designer we could have a new parameter for pins in the PCB or Schematics library components, defining the package length. This has to also be spreadsheet editable, since in a usual case we would have to copy this parameter for thousand-pin components.

2) Importing design rules from Excel spreadsheet. For example PC chipsets usually have a trace length calculator that takes the package lengths and other parameters into account and tells the min/max range for every net on a bus. We could import this spreadsheet into Altium Designer as one rule/net rather than manually creating 80 rules for a DIMM memory interface. Exporting net lengths work fine from the PCB Panel's Nets tab by using simple copy/paste Windows function.

3) Automatically generate From-Tos for all nets in a group or Net/DP Class, based on user defined topology. Now for a wide bus we have to create the From-Tos for every net one by one. The Cadence Allegro has this capability, where we create pin-pairs (like From-Tos) on one net, then create a rule that contains the topology relative to component designators, then apply the rule for all nets on a group of signals.

4) Length Meter improvements needed: Display From-To lengths well. Display the Length meter during routing and sliding as well, since now it pops up only during length tuning. If there are multiple rules on a net, then display multiple length meters on the screen in the same time.

5) Color coding of length parameters on the PCB panel. If a trace or From-To or diffpair length meets all associated rules then the text noting its length should become green, otherwise red. Since the on-screen length meter does not support all editing modes and rules, we normally use the PCB panel to measure length. It would be nice if we didnt have to compare the numbers from the PCB panel with our calculations, but the program could tell if lengths meet the rules or not.

6) Exporting From-To details also doesn't work properly, since from the PCB Panel we can only select/copy the From-Tos that belong to one net, and on the PCB List Panel the From-To routed lengths are not displayed. Also the net report does not contain From-To details. The improvement needed would be to add a column into the PCB List Panel that shows the routed length of the From-Tos.

7) The Rules and Violations Panel should show the number of violations for every rule. Now we have to click on a rule to see how many violations are there for that particular rule. If we have lots of rules then we have to check them one by one to make sure that there are no more violations. We should be able to see these at once in one list or table.

8) PCB List improvements: A filter based on user defined text with wildcards could help in seeing the objects that we want to see during editing and hiding the remaining few hundred objects that are unrelated to the actual editing. For example when we want to adjust From-To objects on different nets, the PCB Panel only shows From-Tos that belong to one net.

To speed up the tuning on buses, we should really see same segment From-Tos on different nets in the same time. The PCB list should also display the From-To detail, called „routed“. With Altium Designer 10 there is no user selectable column for this detail. If the PCB List did display the From-To length and total net length as well, then we could do quick and easy Excel calculations for all From-Tos with copy/paste from/into-Excel, this way speeding up the design process significantly. We can sort objects on the PCB List, but as soon as we start some editing on the PCB, this sorting changes (lines get mixed up) and it becomes hard to find the lines that we were checking.

9) When we select a From-To on the From-To Editor, it should highlight the related track segments, and editing should not be disabled. In Altium Designer 10, it only highlights the associated component pins, and the track editing/tuning is disabled. We have to press Clear, then find the same net on the PCB which is not easy to do.