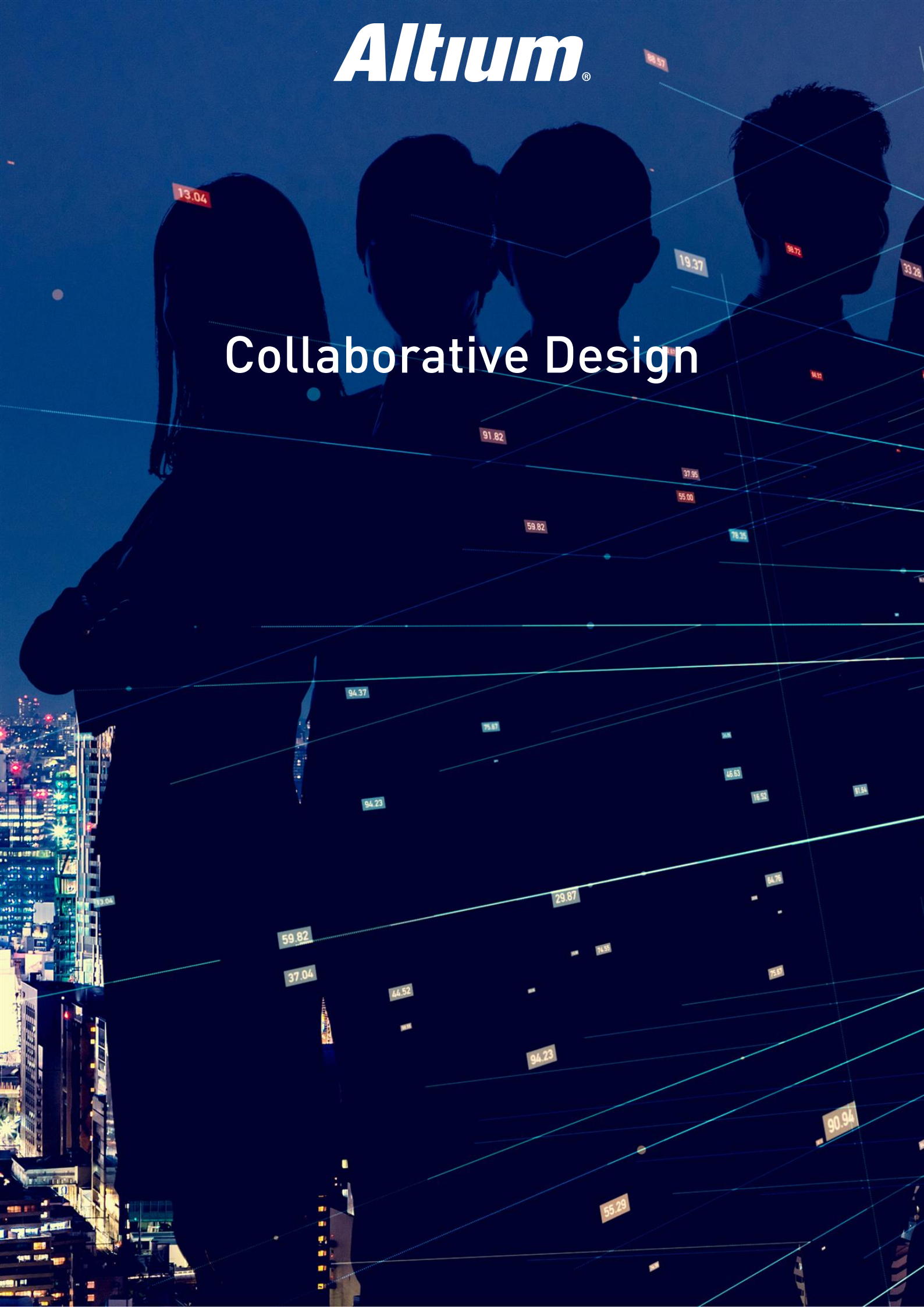


**Altium**<sup>®</sup>

# Collaborative Design





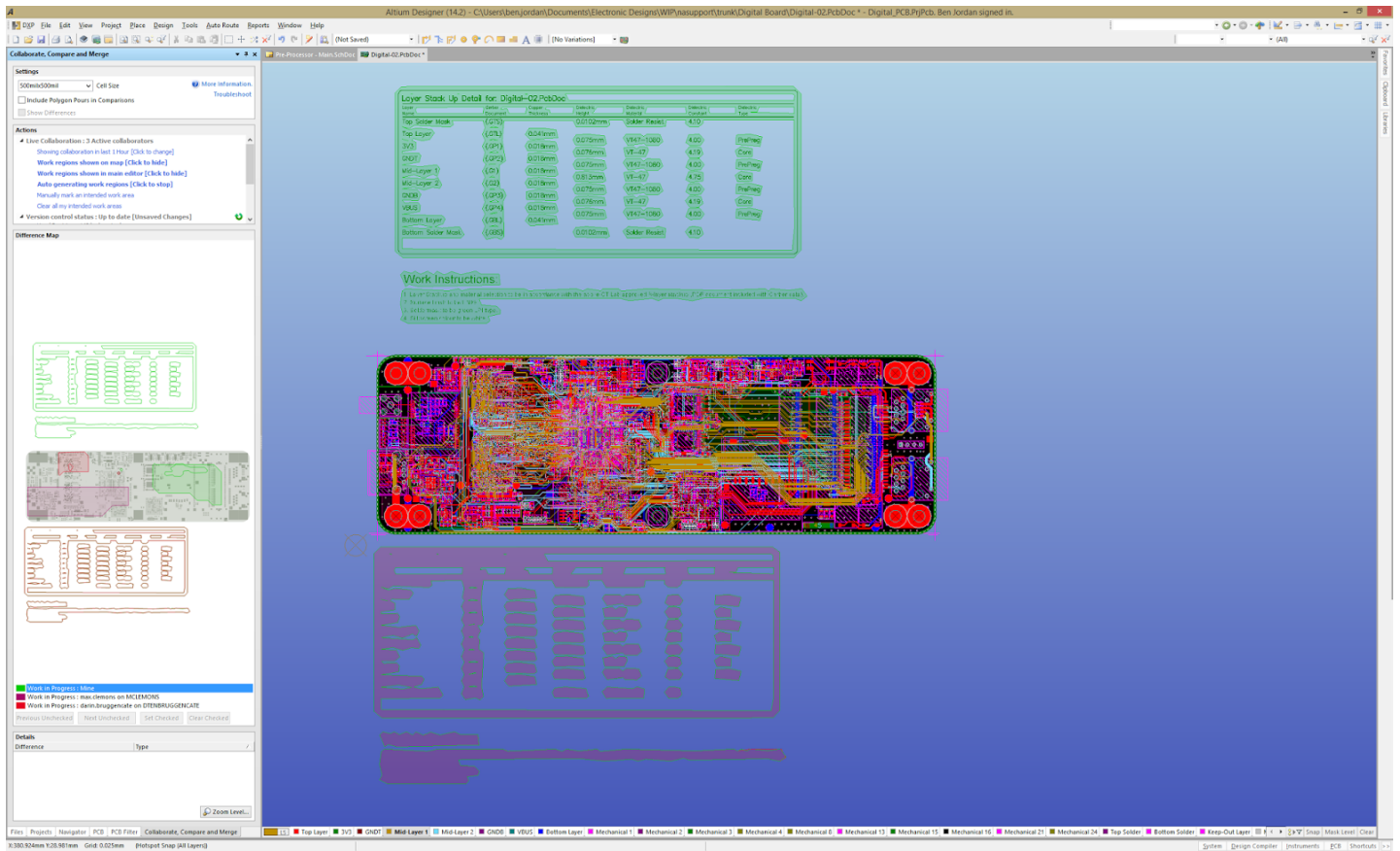
## COLLABORATIVE DESIGN

PCB Design is a complex field and every step you take to design and develop a board has its own set of best practices. But what will truly move the needle in your design process? Collaborative design tools that keep your team not only focused but in the flow. Altium has pioneered collaborative design for EDA tools because Altium PCB design tools start each iteration with ultimate productivity in mind. Join us in an exploration of collaborative PCB design and get an inside look at what has made Altium Designer the designer's EDA tool right from the very start.

Not 100% convinced? Join us as we discuss a variety of topics to help you decide if Collaborative Design is for you, including:

- Collaborative Design Part 1: Designing in Perfect Harmony
- Collaborative Design Part 2: Building a Better System
- Collaborative Design Part 3: Crossing the ECAD/MCAD Bridge
- Collaborative Design Part 4: True ECAD/MCAD Collaboration with IDX

## COLLABORATIVE DESIGN PART 1: DESIGNING IN PERFECT HARMONY



Electronic design isn't a solo gig anymore. Teamwork among dozens or hundreds of designers, engineers, suppliers, manufacturers, and any number of other people is essential for bringing a product to market. This blog series explores the ways to cut through the barriers and reduce some of the tedious back-and-forth dialog that slows design cycles to a crawl. In part 1, we'll take a look at what kinds of features make for effective collaboration tools, as well as some of the different ways people can collaborate on a design.

It's pretty rare nowadays to see an electronic design project done entirely by a single person, or even by a small, centralized team. More commonly, a large team of designers and engineers stretched out across the globe are working together to meet this goal. And as you can guess, coordinating and collaborating on a single design this way is no small task. The question is: how do we cut through the practical barriers to allow true design collaboration?

To clarify, true collaboration means that designers are able to work together on a single project as they choose, whether this is in parallel across different areas of a design, or in tandem, comparing and merging as necessary to complete a single product. Of course, realizing true collaboration requires a set of capable tools, which are characterized by some notable features: visibility, differencing, and merging.

### I Can See Clearly Now

Having insight into what other team members are doing gives a perspective of the design as a whole. This is especially true when several people are working in parallel, where visibility into each design area can help prevent them from stepping on each other's

# COLLABORATIVE DESIGN

toes.

The obvious analogy for this is a map showing the locations of various people. In place of a map however, we have a high-level view of the design project, and instead of each person's location, we have the changes they've made to the design. The usefulness of this feature in a collaboration tool is entirely dependent on just how responsive it is. Do designers see others' changes in real-time? Or only after committing their own changes to the final design?

## EBONY AND IVORY

It's not enough just being able to see what everyone else is doing. Collaboration really doesn't work unless the changes made by each designer are compared against one another, as well as previous iterations of the project, to see how the design has developed. For a collaboration tool, this boils down to differencing, that is, comparing design changes and recognizing their differences.

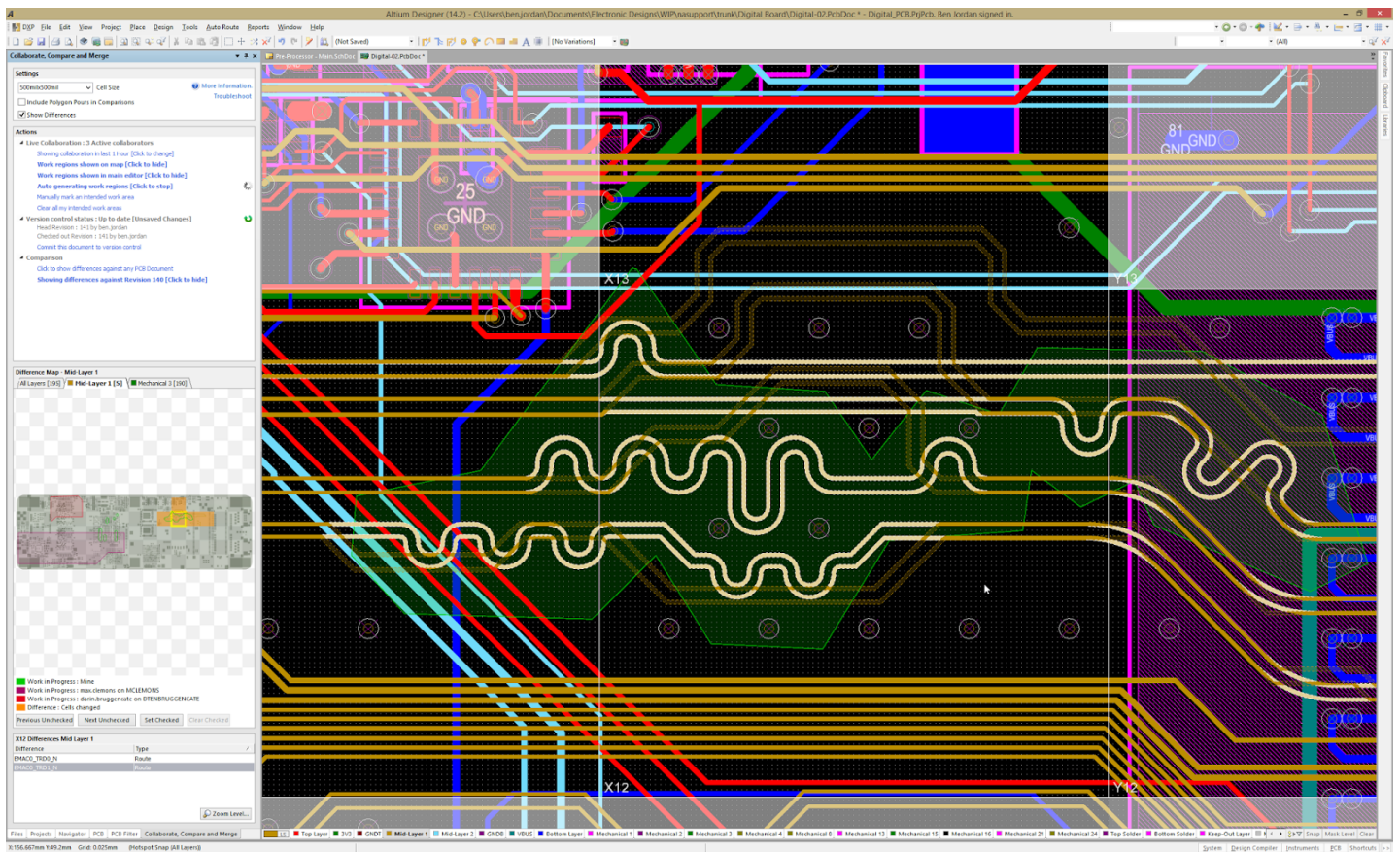


Figure 2: Seeing the differences between a design and its previous versions helps tracks how it has developed.

Differencing in collaboration should work much like it does in a Version Control System, with a bit of added intelligence. As mentioned, design collaboration can be separated by how designers interact with each other. This means prior to starting the project, they can be assigned a specific area to work on, either based on physical location or functional area. Splitting up a project in this way allows designers to work in parallel, and makes identifying differences a much more manageable task.

# COLLABORATIVE DESIGN

## We Can Work It Out

The final step to true collaboration is merging the changes made by each designer into a single master copy. Assuming tasks have been split up beforehand, merging is then just a matter of conflict resolution in the overlapping areas.

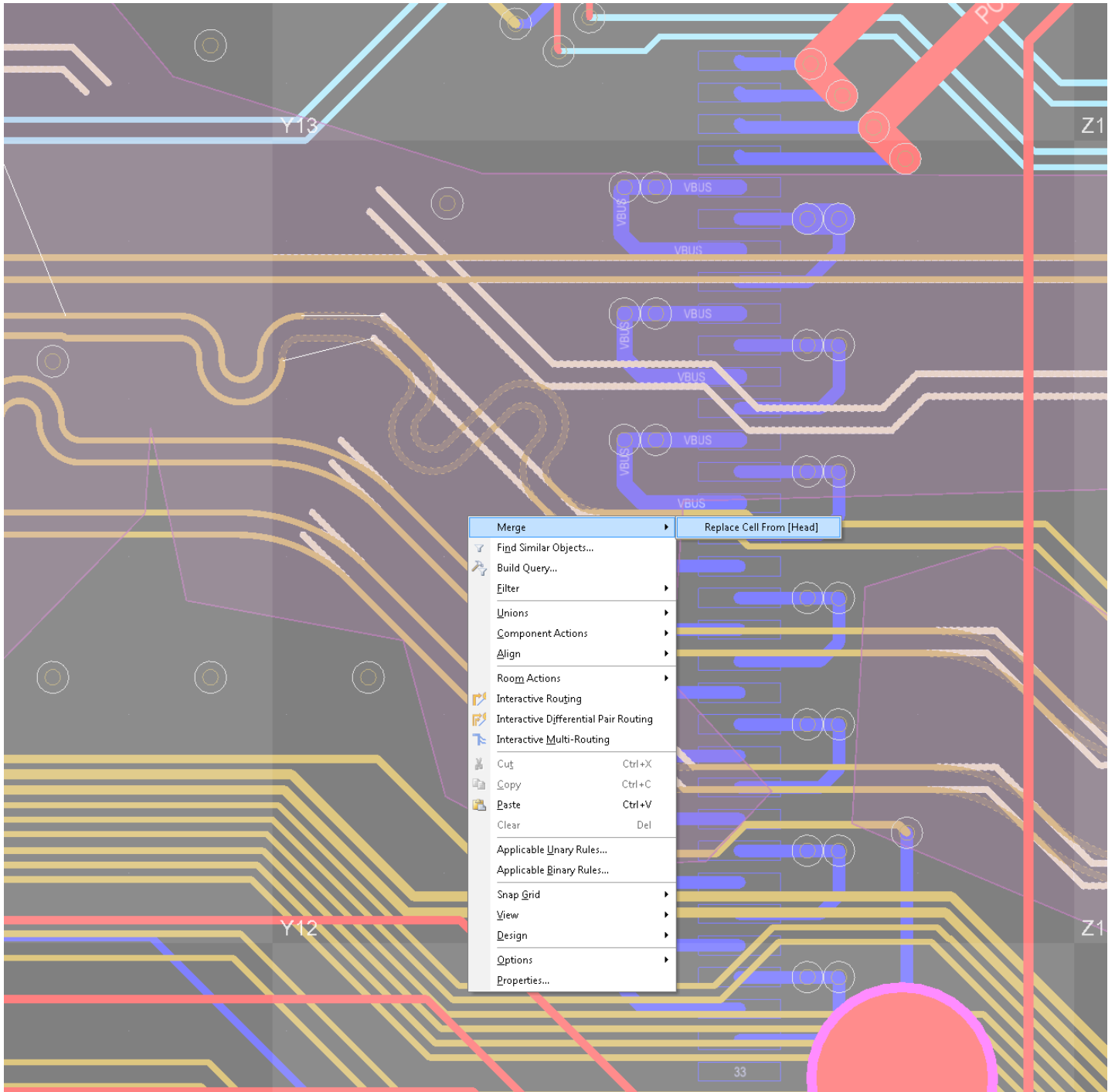


Figure 3: Merging is the final step in collaboration, where differences are resolved and combined into a single design for everyone.

Conflict resolution during collaboration again works just like a Version Control System. If a designer runs into a conflict when

# COLLABORATIVE DESIGN

---

committing changes, they can proceed by either abandoning the changes they've made, or overwriting others' work with their own. How granular this conflict resolution is directly relates to how useful the collaboration tools are in general. For instance, being able to examine, add, or remove individual primitives from overlapping areas can greatly enhance the effectiveness of collaboration tools.

## Come Together

Design collaboration doesn't have to stay in a single domain. Work can be done between people in different locations, across different pieces of software, or even in entirely different organizations!

- PCB - Most likely, this is the scenario that comes to mind first, but is largely the most limited view of collaboration. It's pretty common to see a situation where multiple designers are working together to layout and route the same board at the same time.
- SCH/PCB - Also a common scenario, collaboration across the schematic and PCB is an essential part of the design flow. While this doesn't traditionally fall under the term collaboration, it still represents multiple designers working on the same project. The difference is that this type of collaboration happens across two domains, even though they are usually part of the same software package.
- ECAD/MCAD - Taking the concept to higher level of abstraction, this type of collaboration also takes place across two domains, but is usually achieved by two different software packages. The key to this is how well those programs interface with each other.
- Supply Chain - The ultimate form of collaboration happens across several different domains, and across multiple organizations. Working with component suppliers, engineers, board designers, fabrication and assembly houses, as well as any other relevant parties requires effective and useful collaboration tools.

Whether or not you're aware of it, design collaboration is happening in one form or another during your product cycle. Having a useful collaboration toolset can help save some of that time-consuming back and forth dialog between designers. In the next part of this blog series, we'll get into the gritty details about collaboration and some of the best practices for using it.

# COLLABORATIVE DESIGN PART 2: BUILDING A BETTER SYSTEM

The effectiveness of design collaboration is mostly determined by the structure and organization of the system back-end. For the second part of this blog series, we'll take a look at the best structure and practices for version control, and how this really forms a good basis for collaboration. Enter in the lifecycle management and enhanced administrative functionality of a real design release, and you have an even better foundation.

From my [previous blog](#) on design collaboration, it's clear that good tools have strong capabilities for visibility, differencing, and merging. The features and interactions seen by end users are just the tip of the iceberg though, and really, the back end of a collaboration system determines how effective it can be. A well-organized core structure provides the foundation for seamless collaboration.

It's almost mandatory in a software development environment to enact a strict organizational structure. There are just too many different people potentially working from several revisions of code, and without a system in place to control it all, every development cycle would be a mess. Since version control is so pervasive as an organizational file structure in that domain, it only makes sense to use it as a jumping off point for our electronic design collaboration system..

Version Control: A Not-so-Brief Introduction

Anyone familiar with a version control system knows about trunks, branches, merges, and tags.

A trunk represents the main line of development for a design or project, with the latest revision on the trunk called the head revision. From that, branches are split off as various changes are made by different designers and for different areas of the project. So it's possible (and actually unavoidable during collaboration) to have several parallel lines of development, and at some point, these need to be consolidated to reach a final design or milestone during development.

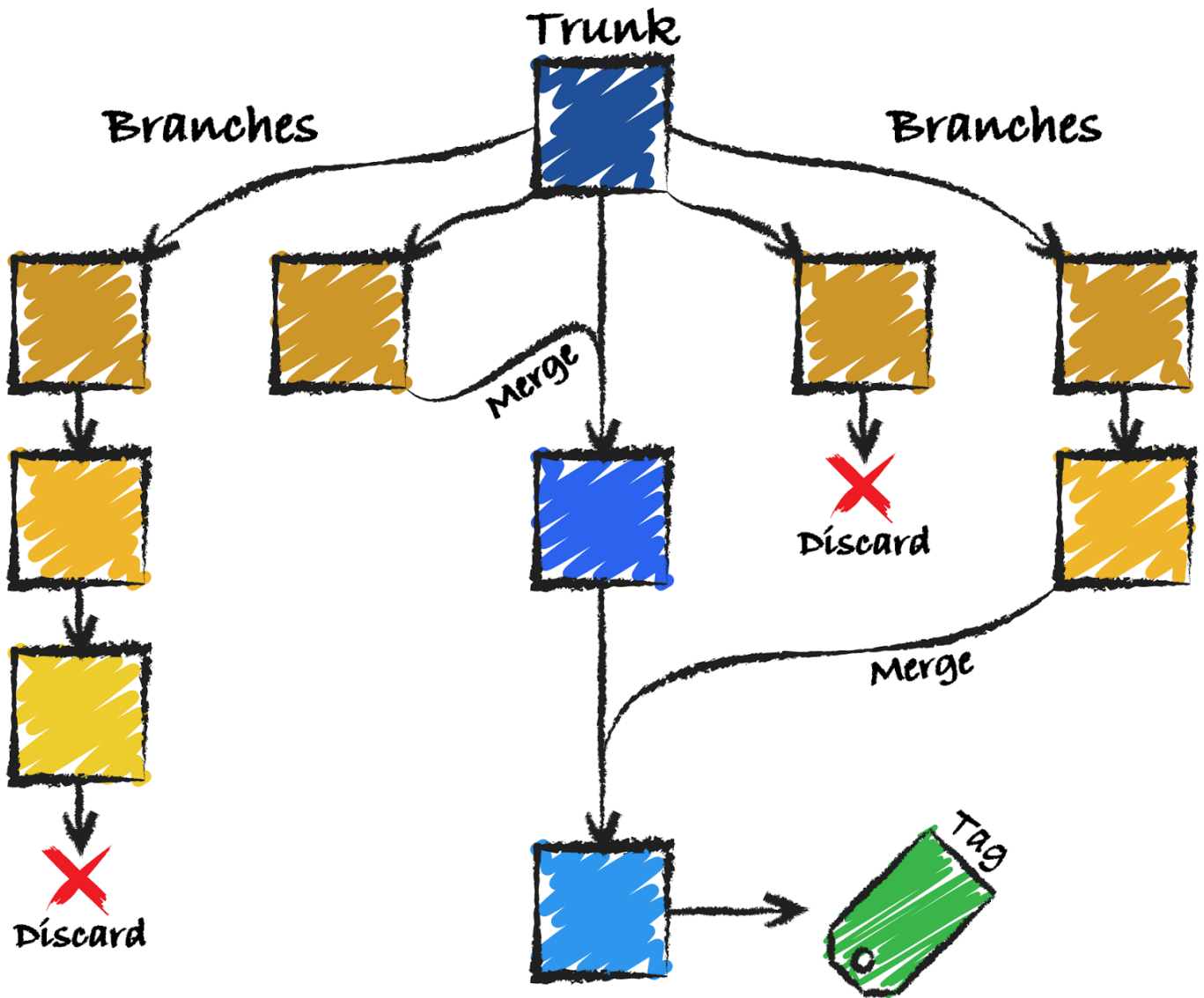


Figure 1: In a version control system, development is organized into a main line called the trunk, with parallel lines stemming from that called branches. Branches can either be merged back into the trunk, or discarded, and certain milestone revisions can be tagged.

Branches can either be merged or discarded, depending on their relevance or usefulness to the final design. If it's determined (through the use of design collaboration comparison tools) that a particular branch doesn't have any useful information to add to the head revision, it will simply be discarded. If, on the other hand, progress can be made towards the final design using the developments of a certain branch, those changes should be merged with the trunk to create a new head revision.

Periodically, a revision will be tagged, giving a snapshot of a project during development. A tag is simply used to give users somewhat of a milestone during a product cycle. For the case of collaboration however, it can also be thought of as a good benchmark after collaboration branches have been merged into the trunk.

So What Does This Have to Do with Collaboration?

Of course, understanding version control doesn't mean that every designer will suddenly be able to collaborate flawlessly. Figure 1 is



# COLLABORATIVE DESIGN

---

simply intended to help visually define parallel development. While collaboration tools, even those that visualize changes in real-time, would seem to be following some complex, cryptic algorithm, they rely on nothing more than a gussied up version control system at their core. The difference is that the type of structure discussed in the previous section is masked by some sort of GUI that is specific to collaboration.

Designers independently make changes to their own version of a project (i.e. their own branch), and at the end of the day, those changes are merged or discarded. Collaboration software simply provides the tools to help designers avoid stepping on each others' toes, while being able to efficiently make decisions about conflict resolution.

## Building a Better System

Version control is a good foundation for collaboration, but there are ways to implement an even better structure. Expanding on the idea of a design snapshot, what if we could take that snapshot and enter it into a larger data management system, with lifecycle state changes and approvals? This is the ultimate goal, because while version control does keep an accurate revision history to help maintain data integrity, it doesn't necessarily provide integration into a product's overall lifecycle. As shown in Figure 2, a product spans several domains, from supply chain, to electrical and mechanical design, all the way to manufacturing. Version control keeps the revision history of items in these domains separate, whereas a true data management system can span all domains required to bring a product to market.

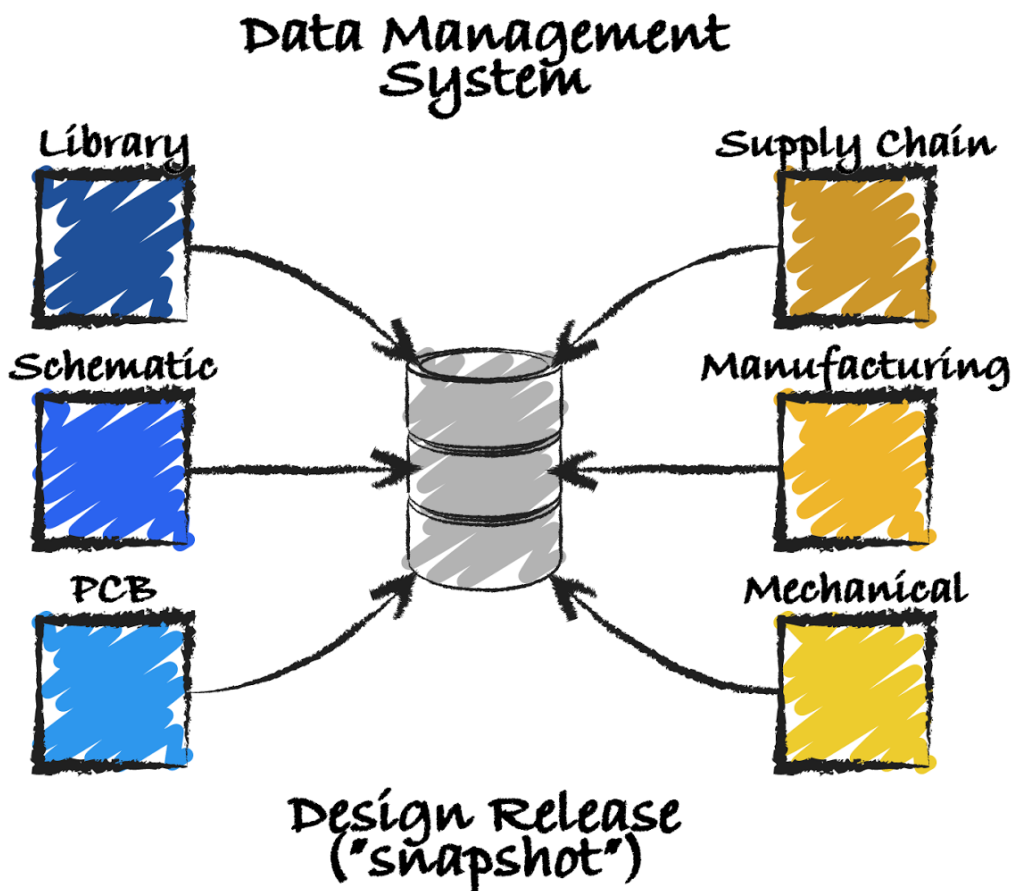


Figure 2: A design release encompasses all information from the design side, as well as from other relevant domains for bringing a product to market, at a particular point in time.

# COLLABORATIVE DESIGN

---

With so many different facets, and potential hazards, for electronic designers to worry about, life can quickly start to look like something out of a horror movie. Product Lifecycle Management (PLM) is all about managing and tracking a product from beginning to end, but this is not limited to the design release as a whole. The best way to understand a design release is to start from the top:

- Design Release
- PCB documents
- PCB footprints
- Land patterns
- Pad stacks
- Bounding courtyards
- Assembly drawings
- 3D mechanical models
- Re-use blocks
- Templates
- Rules and constraints
- Schematic documents
- Schematic symbols
- Alternate views
- Supply chain information
- Datasheets
- Re-use blocks
- Templates
- Pin swap data
- Library documents
- PCB footprints
- Schematic symbols
- Parametric data
- Lifecycle status
- Supply chain information
- Mechanical assembly documents
- 3D mechanical models
- 3D output prints
- 2D CAD drawings
- Output documents
- Fabrication outputs
- Assembly drawings
- Bill of materials
- Supply chain information

## COLLABORATIVE DESIGN

---

- 3D output prints

There is a definite top-down structure, but there is also a significant amount of cross-linking and dependency between different items. Proper data management ensures that all parent and child items are managed independently, and have a revision history kept for them. Similarly, lifecycle status is tracked for all items, down to the finest granularity. While this may seem complicated, what it boils down to is that each designer needs to worry only about themselves, and the natural structure in place will ensure that changes and revisions are visible and approved by an administrator when needed. There are no guessing games when it comes to item tracking, and a product can only move forward in its lifecycle when all parties involved are on the same page.

Having a strong medium for integration between several domains promotes teamwork and collaboration far better than a straight version control system would. So, we've figured out that a design release process is ultimately the best engine to drive collaboration, especially across separate domains, but another interesting topic is just how end users can handle moving between these domains. In the next blog, we'll look into this.

## COLLABORATIVE DESIGN PART 3: CROSSING THE ECAD/MCAD BRIDGE

Interfacing between electrical and mechanical design software is one of the most important forms of design collaboration in modern electronic design. As PCBs become smaller and more dense, and as mechanical housing restrictions become tougher, there is a great need for seamless design data transfer. For the third installment in this blog series, we'll take a look at some of the options available for crossing the boundary between electrical and mechanical domains.

So now that we've established the inner workings of a collaboration system (see [Part 1](#) and [Part 2](#) of this blog series), let's check out one of the applications of collaborative design. As discussed in Part 1 of this series, the term design collaboration can mean different things to different people, and can span a single domain or several. Some of the different domains include electronic design (PCB/Schematic/Library), mechanical design, supply chain, manufacturing, and other spaces accessed during a product's lifecycle.

ECAD/MCAD collaboration, between the electrical and mechanical design domains, is particularly important as PCBs become smaller and more densely populated, and as mechanical constraints become more strict in turn. Having a good bidirectional interface between the two domains can make life infinitely easier for all parties involved, and can help ensure a cleaner design cycle, with greater product integrity in the long run. While there are several options for interfacing across CAD software, there are currently two standards commonly used.

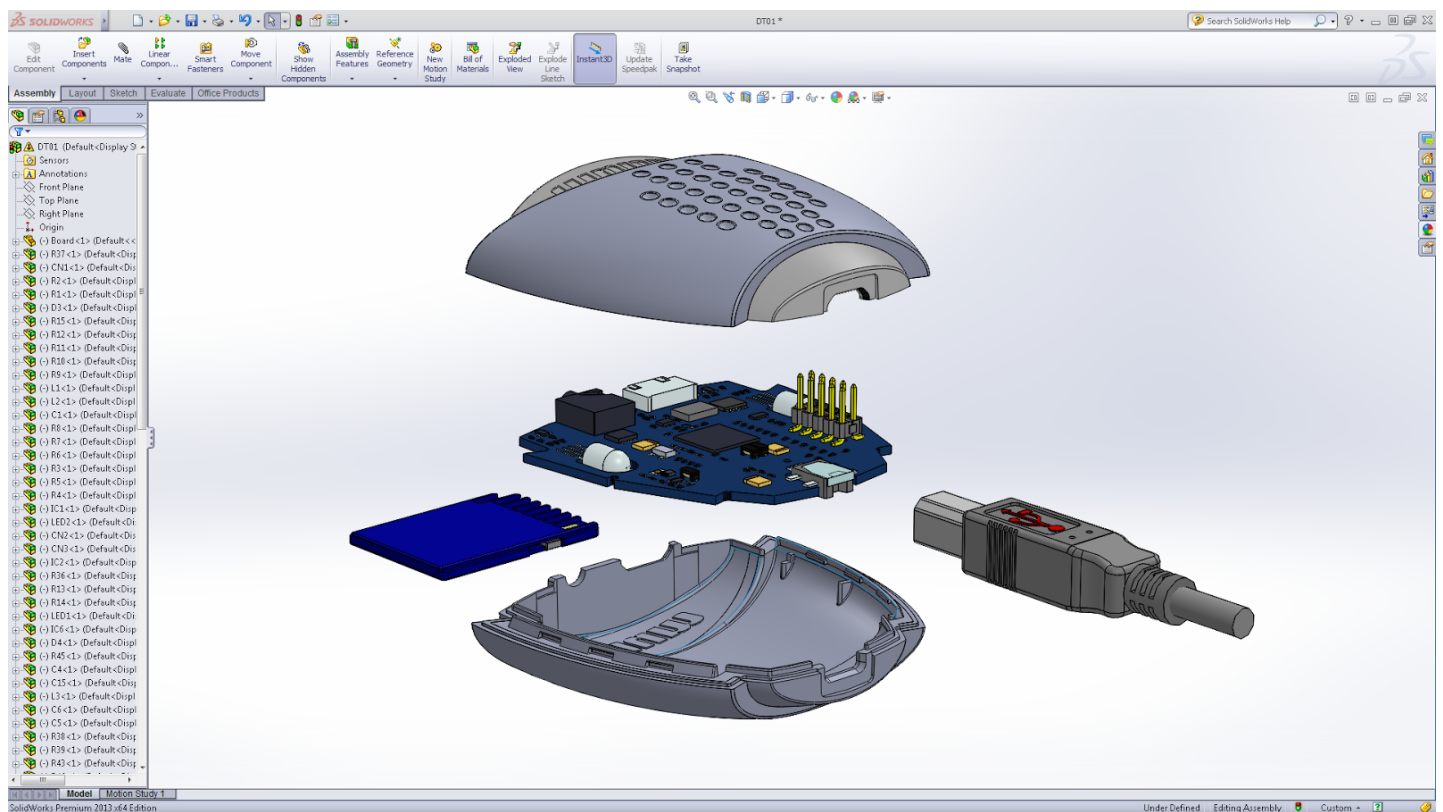


Figure 1: Having the ability to interface between electrical and mechanical CAD software is essential for modern electronic design.

### STEP (STANDARD FOR THE EXCHANGE OF PRODUCT MODEL DATA)

A true 3D representation of design data, STEP models can be used for PCBs, components, mechanical assemblies/housings, and any other design files which may be collaborated on by multiple designers using different programs. It's important, however, to understand both the benefits and limitations associated with using STEP for bidirectional transfer between programs.

Benefits	Limitations
Adheres to <a href="#">ISO 10303 standard</a> , meaning all design models are consistent and follow the same protocol.	Since it is intended as a common format for data exchange between programs, some features exclusive to each piece of software will inherently be lost.
Having such a common and generic format means several component manufacturers make 3D models of their parts available for download.	Does not lend itself well to complex design features, and the addition of some items (such as copper traces, silkscreen, etc.) can increase file sizes significantly.
Being a true 3D format (instead of a pseudo 3D / "2.5D" format) allows for complex model features and curvature, such as folded board shapes for Rigid-Flex PCBs.	Has no function for true collaboration, and simply requires manual checking by each designer to ensure design data is correct and up to date.

The two Application Protocols (APs) under ISO 10303 used for 3D CAD data are AP203 (older format) and AP214 (newer format). More information about these APs can be found [here](#).

### IDF (INTERMEDIATE DATA FORMAT)

For several years, IDF was considered the de-facto standard file format for transferring design data between electrical and mechanical CAD software. In reality, IDF is not truly 3D, and is instead represented as so-called "2.5D", that is, viewable as 2D in ECAD, and extrapolated using model height information to 3D in MCAD. Of course, this scheme does not lend itself to complex 3D shapes, which may be required for certain data exported to the mechanical domain, such as folded Rigid-Flex board shapes.

IDF is good for preliminary mechanical clearance checking, since component and board heights are recorded and transferred with the file, but there are limitations. If clearance checking or modelling is more involved, component bodies must be replaced post-transfer, meaning additional work for the mechanical designer, as well as potential issues matching body alignment and rotation.

### OTHER SOLUTIONS FOR ELECTRICAL/MECHANICAL INTERFACING

#### Native 3D PCB Editing

Including tools within ECAD software for mechanical design, or at least alignment, placement, and export of 3D mechanical models, allows much of the work to be done in a single software package. Altium Designer includes capabilities for aligning 3D component models to footprints, modelling and clearance checking for housings/enclosures, and if necessary, standard exports of complex PCB features for MCAD interfacing.

#### ECAD/MCAD Collaboration Tools and Formats

## COLLABORATIVE DESIGN

---

Certain third-party tools enable better collaboration than could be offered using either of the aforementioned file formats. Using a more robust design data transfer system, these tools allow much more comprehensive bidirectional design, visibility and conflict resolution, as well as more complete integration with each native software interface. Similarly, IDX (Incremental Data eXchange) offers this same sort of bi-directional capability in a vendor-neutral file format.

## WHERE DOES ECAD/MCAD COLLABORATION GO FROM HERE?

With the inclusion of 3D editing features in most ECAD software packages, the need for good software file transfer solutions is dwindling, beyond perhaps the initial creation and import of mechanical data.

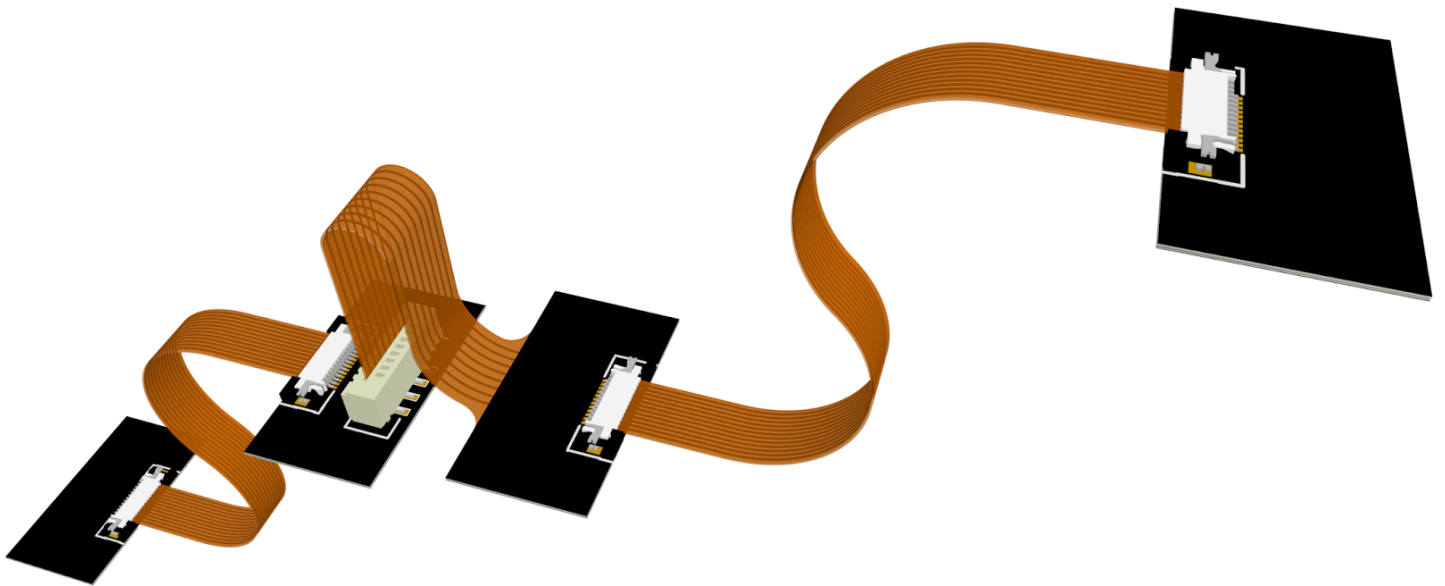


Figure 2: Complex electro-mechanical assemblies, such as multi-board designs connected through flex cabling, will be a driving force for ECAD/MCAD collaboration moving forward.

This doesn't mean there is no need to support and improve such an interface, however. It simply means that true collaboration between software domains will become that much more important to the design process. More sophisticated projects, such as multi-board designs and complex electro-mechanical assemblies, are still generally beyond the scope of a single domain. But that won't be for long. Our flagship PCB design software is changing [ECAD/MCAD collaboration](#) for the better.

## COLLABORATIVE DESIGN PART 4: TRUE ECAD/MCAD COLLABORATION WITH IDX

True ECAD/MCAD collaboration is more than just preserving details when moving between design environments. Visibility into design changes, comparing and merging capabilities, revision tracking and commenting are all key elements of collaborative design as well. In the final part of this blog series, I look at how IDX might be the best solution yet.

In the last part of this blog series, I dove into ECAD/MCAD collaboration, and why it's so important to have a good system for passing design data back and forth between the two domains. But even through my breakdown of the popular methods used today, the question remained: isn't there a better way to do this? Why can't we have a true system for collaboration between electrical and mechanical designers?

And I'm not just talking about preserving details when moving back and forth between environments, although that's important as well. True collaboration means designing simultaneously or in parallel using a system with features for visibility, comparing, merging, tracking, and commenting on design changes. Looking at the available options, I came across a few solutions, each with their own strengths and weaknesses.

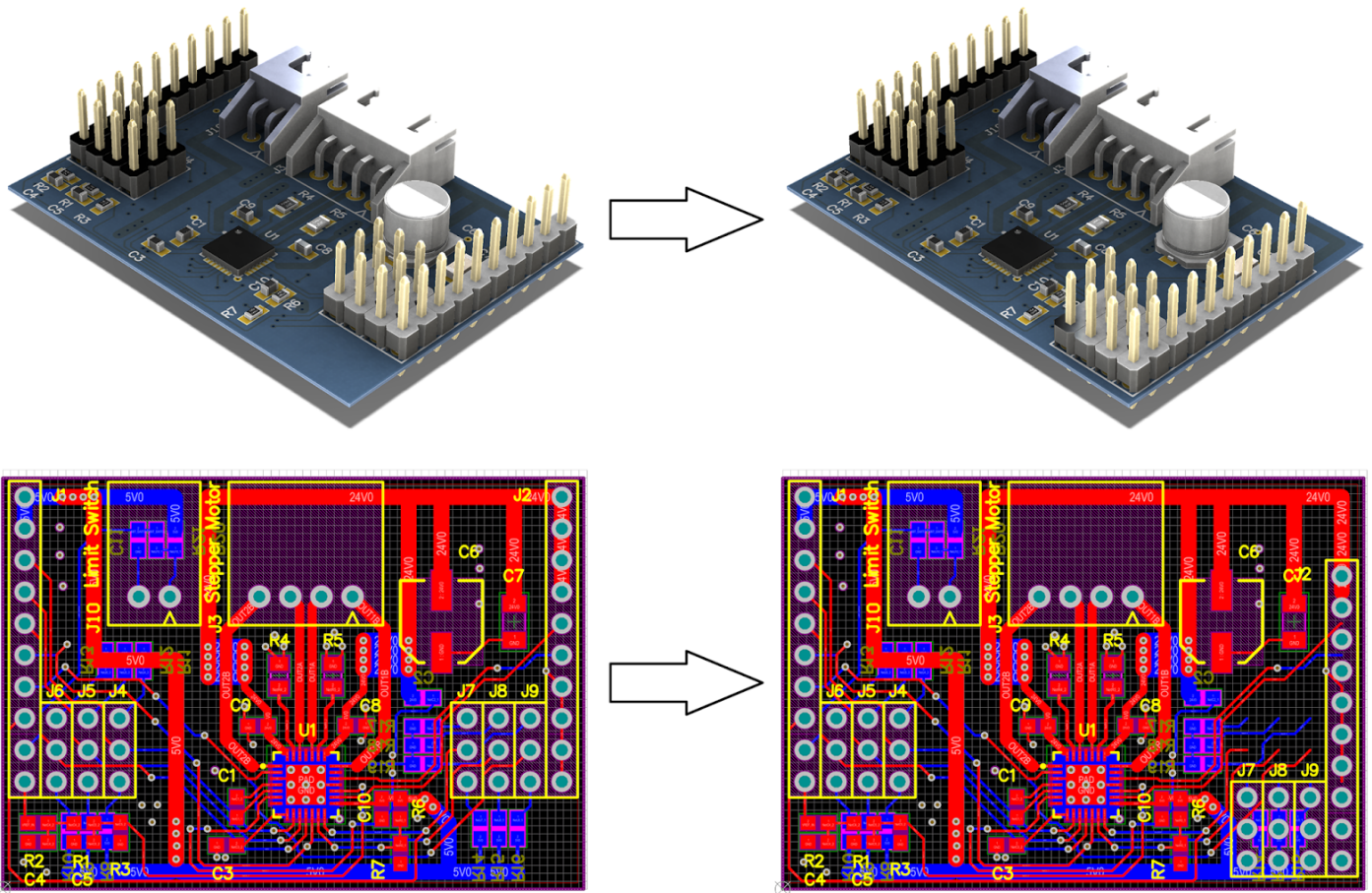


Figure 1: True ECAD/MCAD collaboration gives designers visibility into incremental changes, allowing them to design simultaneously.

## COLLABORATIVE DESIGN

---

If a connector needs to be moved, both designers can see those changes and make any necessary alterations in response.

## NATIVE ECAD/MCAD COLLABORATION

There's nothing better than being able to use a single software tool for all electrical and mechanical design. We're getting closer to this every day with features for creating extruded models and enclosures available right from within ECAD tools. However, the single application approach may not have all necessary functionality at the moment, and designers may simply have their own preferences for mechanical design.

## THIRD-PARTY APPLICATIONS

There are a number of external tools that can directly plug in to electrical and mechanical CAD packages, providing a better gateway between the two programs. While this allows more seamless transition and collaboration, it's yet another tool with its own little details and nuances. Plus, it's another link in the chain you'll have to troubleshoot when things aren't going quite right.

## NEUTRAL FILE FORMATS

Neutral file formats are a bit tricky to categorize, and essentially exist outside of each design domain. They don't require a direct interface to either software environment, thus avoiding time-consuming connection troubleshooting issues. On the flip side, each tool can write to this standard interface while maintaining native methods and toolsets, ensuring each designer's workflow is consistent.

## WHAT ABOUT IDX?

This leads me to IDX. The IDX format is like a grab bag of the best features pulled from other collaborative ECAD/MCAD formats, with the added benefit of enabling true collaboration. It was based on the ProSTEP EDMD format, which itself was based on STEP AP 210 and AP 214 formats. The difference between IDX and standard file formats like STEP is that it tracks incremental changes, which electrical and mechanical designers can choose to accept or discard. Here's how it works:

- **IDX baseline file** - At the beginning of (or at any point during) a board design, an electrical designer starts the process by exporting a baseline file. This file is the actual 3D model data, similar to a STEP model.
- **Import into a mechanical environment** - The baseline file exported from the electrical side will be imported in a mechanical tool. It's likely that a mechanical designer will have some sort of enclosure or mechanical fixtures in place already, but they now have the potential to fully edit any part of the product to make everything fit together and work correctly
- **Incremental change XML file** - As changes are made to the design from either domain, they will be recorded in a file, stored in the same location as the baseline file. Design changes won't automatically be applied either, so there's no need to worry if something doesn't quite add up. Changes are tracked and comments are added to this file, and designers can choose to accept or discard any changes as they see fit.



## COLLABORATIVE DESIGN

---

- Design cycle in full effect - Using this methodology, designers can make changes and comment on those changes, but they won't take effect until the other side approves. By tracking a design like this, it's simple to revert to an older version if something didn't work out along the way.

IDX is by no means the ultimate solution, since it still employs a middleman between environments. But it's comprehensive enough for complex design features, is standardized to allow translation between several different programs, and enables design collaboration in the truest sense. It's currently one of [the best ways to create a seamless workflow for designers](#), potentially in different parts of the world, to do different tasks simultaneously.

## ADDITIONAL RESOURCES

Thank you for reading our guide on Collaborative Design. To read more Altium resources, visit the Altium resource center [here](#) or join the discussion at the bottom of each original blog post:

- [Collaborative Design Part 1: Designing in Perfect Harmony](#)
- [Collaborative Design Part 2: Building a Better System](#)
- [Collaborative Design Part 3: Crossing the ECAD/MCAD Bridge](#)
- [Collaborative Design Part 4: True ECAD/MCAD Collaboration with IDX](#)