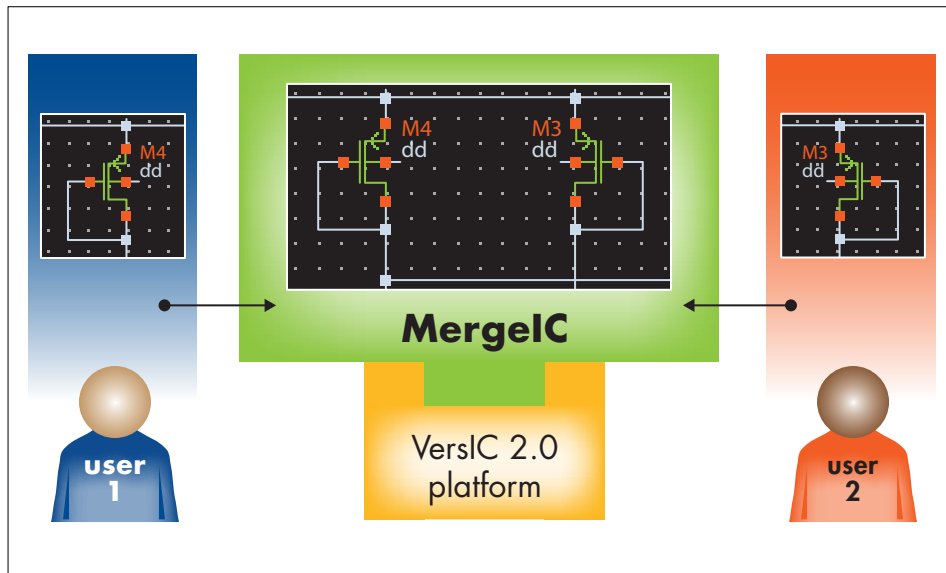


Hierarchical Visual Diff and Merge



Visually diff your schematics and layouts

Diff different versions of cellviews from the repository and view the results inside your Cadence session.

Intelligent diffs

Differentiate between functional (topology) changes and ECOs, display them intuitively as part of the MergelC GUI.

Merge your changes into the production branch

Merge your differences into the target schematic or production branch.

Manage your layout changes

Organize your layout changes on a layer-by-layer basis as part of the MergelC UI. Display the modified/added/deleted shapes between versions in the database, and track your changes in the MergelC UI.

Hierarchical compare

Binary diffs are most efficiently performed using Perforce or Subversion and can significantly outperform verification tool XOR operations. Pass the results of a binary diff to the MergelC GUI for cell-level comparisons.

Compare versions in the repository directly

Comparisons can be performed against older revisions in the DM database, without an explicit version retrieve.

Standalone operation

MergelC can be run standalone independent of VersiC or any DM infrastructure.

Finding (and intuitively displaying) the differences between two objects in a version-controlled environment is a base-level requirement for software configuration management (SCM), but, unfortunately, has been a glaring omission in IC data management (DM) offerings. MergelC is the first hardware-centric Diff/Merge tool to address these issues, and is available as a "DM-Driven" module in the VersiC 2.0 Platform.

With MergelC, schematic differences are categorized into non-functional changes (or ECOs), in which the circuit topology hasn't changed but instance behavior is different, and functional changes, in which the underlying netlist has changed. CDF properties, master-types, instance names and other ECOs are analyzed, and the difference pairings are identified and displayed in the form. When a new instance or net is detected, this is shown with no matching object in the MergelC UI, making it easy for the user to identify these.

Layout changes are identified on a layer-by-layer basis, and differences categorized by shapes added, deleted and modified in the MergelC UI.

Once differences have been identified, they may be merged into the target cellview from the MergelC UI. Property changes, instance names and master types are applied to the target instances for ECOs, and new instances/nets/shapes are created on the target cellview. These may be applied one-at-a-time or en-masse.