

Fast, accurate layout merging for SoC flows

By James Paris | [No Comments](#) | Posted: February 8, 2019

Topics/Categories: [EDA - DFM](#) (<https://www.techdesignforums.com/practice/topics/eda-topics/dfm-dfm/>), [Verification](#) (<https://www.techdesignforums.com/practice/topics/eda-topics/design-verification/>) | Tags: [data translation](#) (<https://www.techdesignforums.com/practice/tag/data-translation/>), [database](#) (<https://www.techdesignforums.com/practice/tag/database/>), [DRC](#) (<https://www.techdesignforums.com/practice/tag/drc/>), [export](#) (<https://www.techdesignforums.com/practice/tag/export/>), [GDS](#) (<https://www.techdesignforums.com/practice/tag/gds/>), [Oasis](#) (<https://www.techdesignforums.com/practice/tag/oasis/>), [OpenAccess](#) (<https://www.techdesignforums.com/practice/tag/openaccess/>) | Organizations: [Siemens EDA](#) (<https://www.techdesignforums.com/practice/organization/mentor/>)

How to achieve efficient merging of data from formats such as OASIS, GDS, and OpenAccess to ensure timely verification through DRC runs.

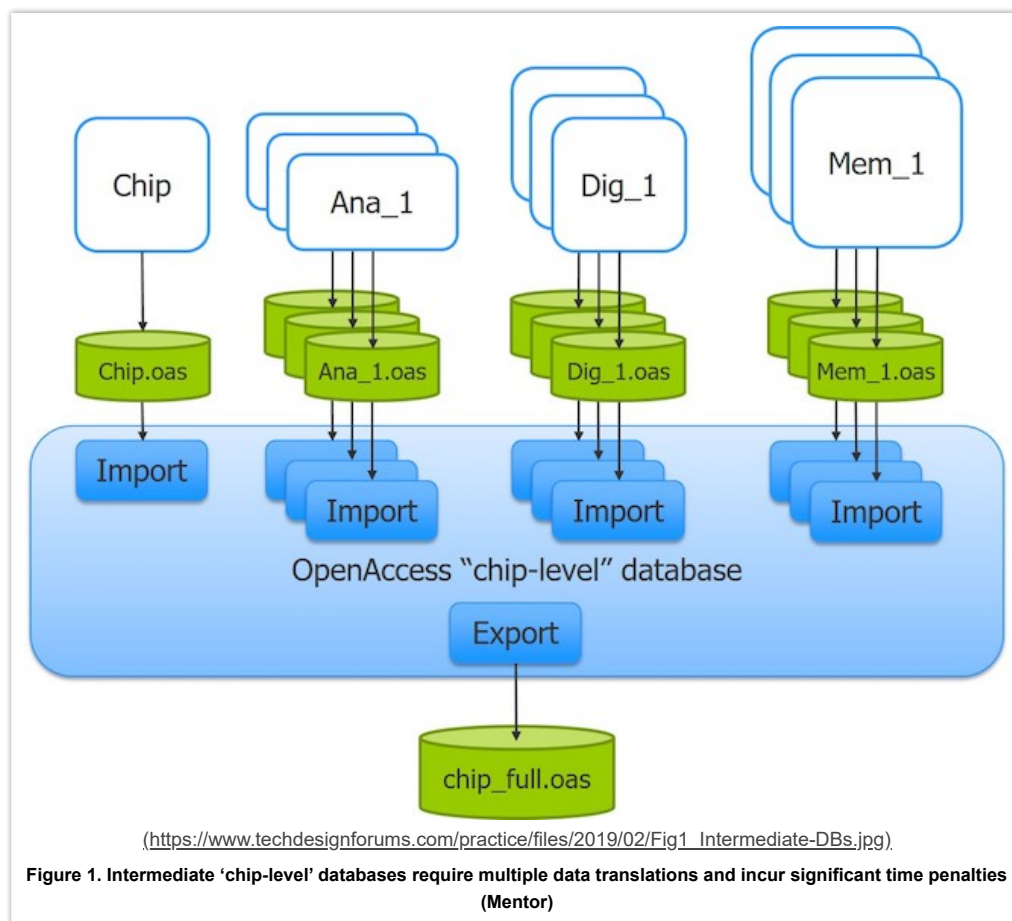
System-on-chip (SoC) design teams typically run full-chip design rule checking (DRC) verification multiple times throughout the design implementation. This best practice lets design teams find any chip-level violations between block instances early in the design flow, when they are easier to correct. However, assembling the latest physical layout data for each block and merging it into a single OASIS or GDSII database for full-chip verification can be difficult and time-consuming. To perform these DRC runs with the needed frequency, chip-level designers need a fast, reliable data merging flow that satisfies the needs of the continuous-build parallel design implementation flow while minimizing or eliminating time-intensive data management activities.

Database format conversion

One of the first issues that must be resolved is database conversion. A full-chip database always uses either the GDS or the OASIS format. External or intellectual property (IP) blocks already in the OASIS/GDS format can simply be referenced for merger during layout export from the place and route (P&R) tool as chip-level OASIS or GDS files. However, if some external block data is not available in the OASIS/GDS format, chip-level export and full-chip verification cannot happen until these data sources have been converted.

Intermediate database formats

Internal data merging flows that use intermediate databases like OpenAccess can create unintended bottlenecks. These flows require that all the blocks and the chip-level design are converted to and stored in the OpenAccess format. This final OpenAccess chip-level database design is then exported to GDS or OASIS (Figure 1). As the blocks and chip-level design are exported to the GDS/OASIS chip-level database and merged, chip-level block abstracts are replaced with the physical design blocks, and this creates a full hierarchical database. Unfortunately, runtime requirements for this multiple translation flow make its use prohibitive. Some users report data merging under it taking up to 20 hours to finish for a full chip.

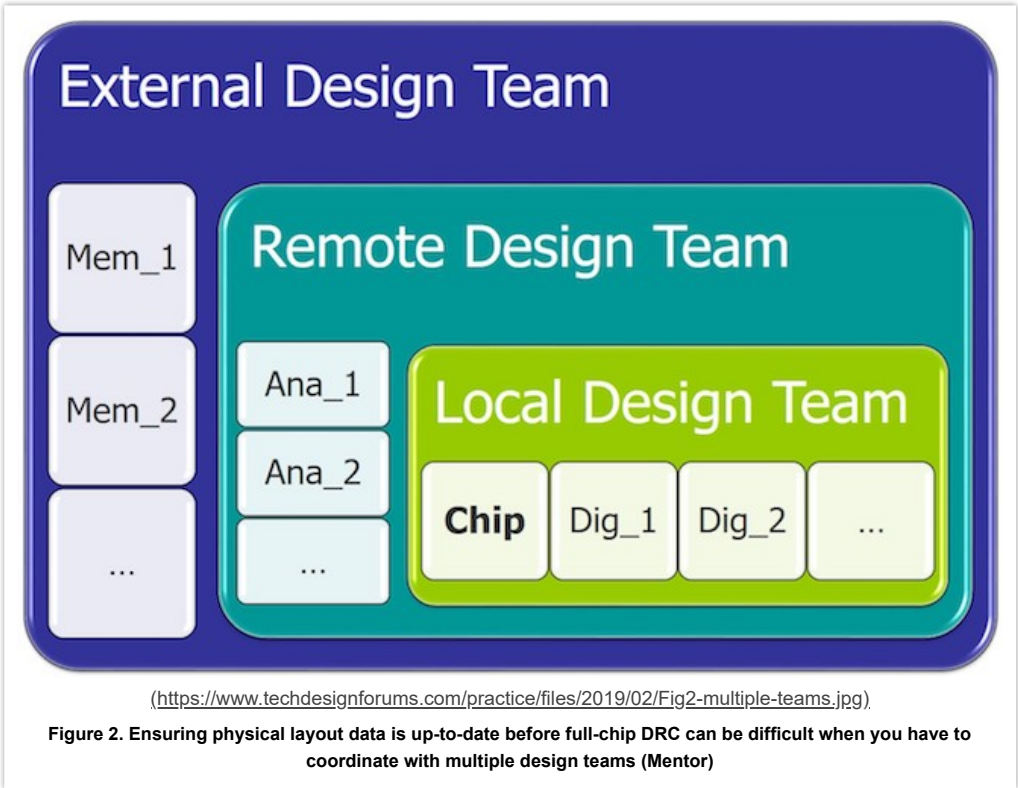


Unless the use of intermediate database formats is unavoidable, it should be discouraged during physical verification iterations. If you minimizing or eliminating the use of intermediate conversions, design teams gain more time to analyze results and fix layouts within delivery schedules.

Database updates

All the data being merged should be the latest version of that data. If any chip-level data is exported with an out-of-date reference, the output violations for the DRC run will be invalid. We use cookies to help us understand how the website is used and to make on-site navigation easier. If you continue to use this site we will assume that you are happy with it.

However, ensuring that all data is up-to-date is not always straightforward. Block development is often done in parallel with chip-level implementation, meaning the work is distributed between internal and external teams. For any SoC, there may be dozens of designs in various stages of development at a single time. In addition, block development teams working in parallel may be located at different internal sites, and/or in different time zones. Even co-located teams may have different work schedules. Some blocks may be assigned to an external team or contractor who may not have internal network access, and that could delay data transfer (Figure 2). Being aware of and accounting for any or all of these concerns when planning the data merge flow helps ensure all data is current for each interim chip-level DRC run.



Defining a schedule for full-chip data merge and physical verification runs is one technique that can help mitigate these issues. If all teams and suppliers are aware of the schedule for full-chip verification runs, they can ensure their block data is up-to-date and available in the required format when needed.

Database merging

Once you know you have up-to-date data, the next step is to manage the data merger. Many database filemerge utilities must read each database into memory before merging them in a single full-chip database. This approach not only requires a significant amount of system memory. It also results in extensive runtimes when used with full-chip verification flows. Using a disk-based in-memory merging utility (layout merge) lets you merge multiple input OASIS or GDS databases into a complete hierarchical database for full-chip DRC without having to read each database into memory first (Figure 3). This technique dramatically reduces memory requirements and enables faster runtimes than in-memory merging solutions.

