

Electronic Design Data Management

The impact of team-based design and re-use on data management

Dizain-Sync

Design Data Management is more than just version control. If properly setup, it encompasses work-area management, release management, build-management, issue tracking and project control and is then a key to team productivity and product quality

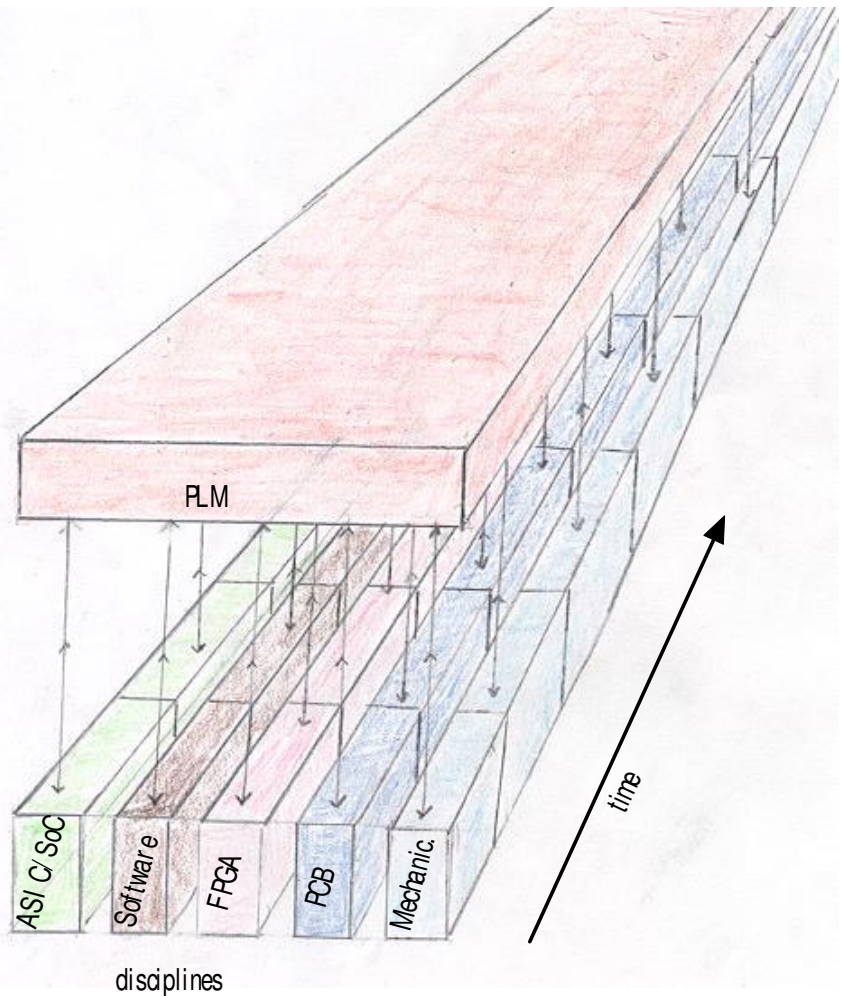
The different disciplines that together create your product, all need a different approach to design data management.

ASIC and SoC design data management need a strong focus on IP re-use.

Software data management has a long tradition with methodologies such as SEI's Capability Maturity Model and Rational's Unified Change Management.

For PCB data management part selection and library management are of most importance.

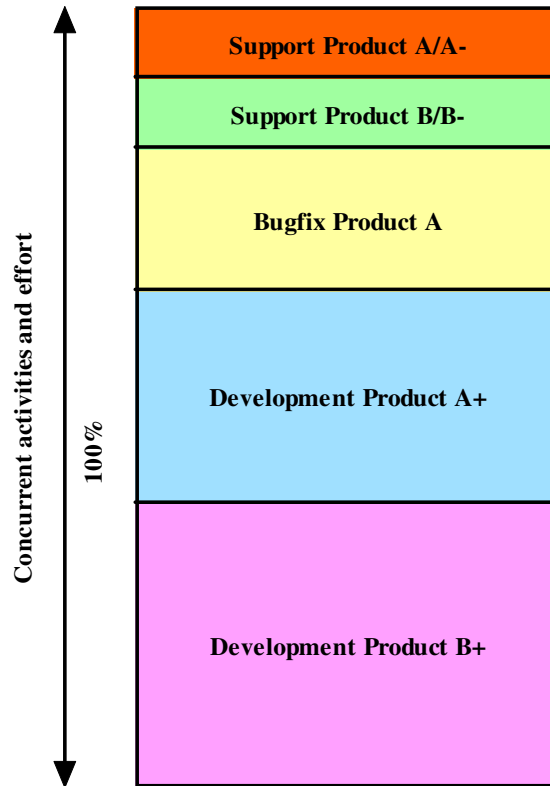
FPGA design data management is of increasing importance for many companies, because the sizes of the FPGA's nowadays FPGA design requires more-and-more collaboration.



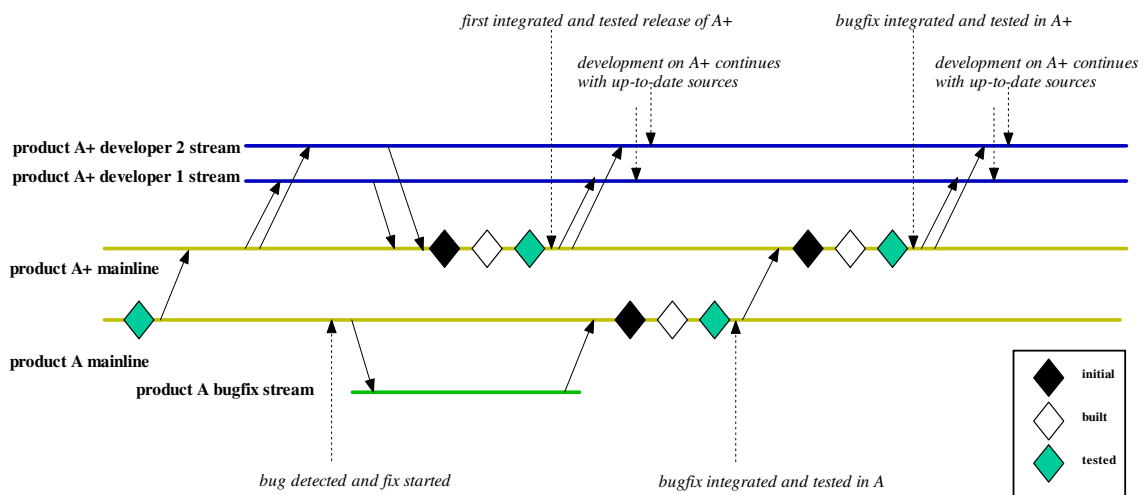
As sketched in the figure above, the multiple disciplines perform their design tasks over time and on critical points in time store and retrieve product data to and from the Product Lifecycle Management (PLM) system. The PLM system should contain all the product's data from its design to its disposal.

Let us consider the situation of a small FPGA design group that has two main productlines called A and B, the endproduct is a measurement system and product A performs the data acquisition and product B is a dedicated network interface. Product A and B are the most recent versions that are delivered to the group's customers. There are still some older version in use by customers, shown here as A- and B-. Furthermore, new product development has started on products A+ and B+.

At some point in time, the activities and efforts in the FPGA design group could be such as shown in the figure below. There is significant development effort on the new products, A+ and B+, but a bug has been discovered by a customer in the delivered product A, so a bugfix is on its way (and requires also significant effort). Furthermore, all products in the field (A, A-, B and B-) require some support.



Then the data management for product A can look like the figure below.



In this figure, a new mainline is started for the new product A+, because it is a significant new development and uses a newer FPGA type and a tested release of the source code for product A is

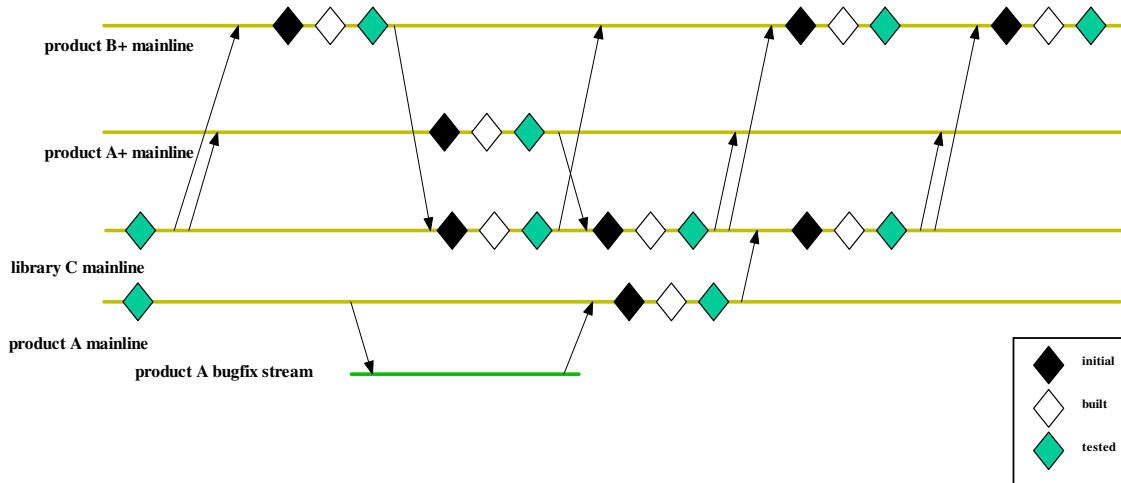
used as starting point for the development of product A+. Two developers are allocated for the new development and both are given a development stream, so that they can perform their changes isolated from each other. At some point(s) in time the developers integrate (merge) their code in the mainline for product A+ and the result is built and tested. After successful integration and testing, development is resumed by both developers with the tested sources (rebase). This process may be repeated several times and good practice is to plan ahead the points in time at which the integration will be done.

A bug in product A was discovered and appropriate measures to fix it were taken by starting a bugfix using the sources for product A. This bugfix is then integrated back into the mainline for product A and tested. On successful completion of these tests, a new release of product A in which the bug is solved may be released to the customer(s). Furthermore, the bugfix is also integrated in the mainline of product A+, to assure that product A+ will not contain the same bug that has been solved in product A (a situation that is not uncommon and very annoying for customers).

At some point in time, development on A+ will then continue with the up-to-date sources (this can be as shown in the figure above, but when the changes for the bugfix are more significant, it will usually be more efficient to wait to the next integration of the development streams has to be done).

The situation described above is not very complex and could be dealt with without any form of data management software; however this is not very advisable and would undoubtedly lead to errors. Furthermore, it is impossible to oversee the status by anybody besides the developers themselves.

A more complex situation arises, when re-use is taken into account. Re-use has been proven to be the most effective way to cope with the increasing size of FPGA's and ASIC's (other ways may be high-level synthesis based on Matlab or C++ code). In the case of our FPGA design group, it is likely that the data acquisition and dedicated network products have some functionality in common. And after over time this functionality is contained in an Intellectual Property library of common components, called C. For library C, the data management picture will look as shown in the figure below.

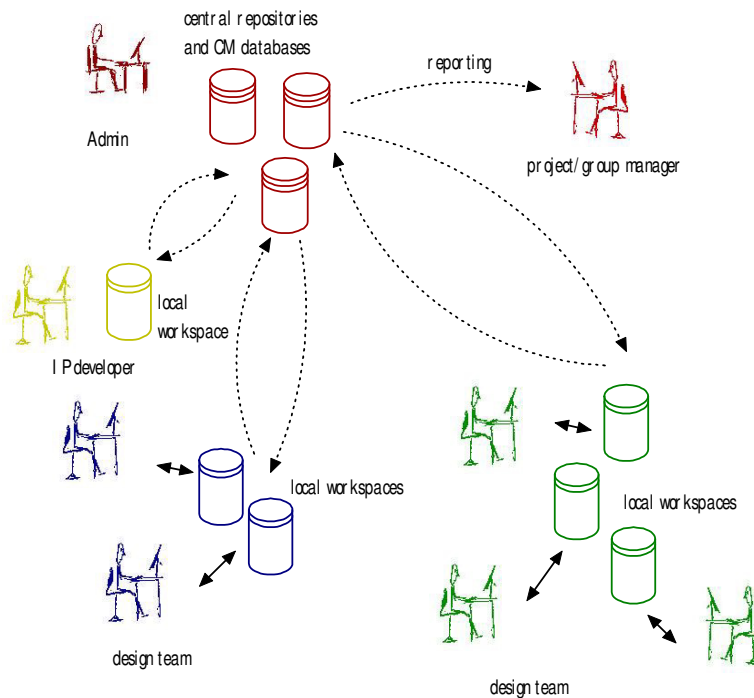


Obviously, this process is of a complexity that is not feasible without a good setup for revision control and change management.

Processes, reporting, issue tracking

Of equal importance is that proper work processes are in place. To support the work processes, change management is necessary, because there must be a link between the versions that are changed and the activity that was the reason “why”. Typically the links between activities and version changes is stored in a change management database. This will then create the powerful possibility of automatically generating reports, so that a manager may closely follow the status of all data under development.

Now, looking at the individuals that are involved in the process, this may look like the figure below.



The picture shows central collection of repositories and change management databases, which are controlled by an administrator. Furthermore it shows several design teams that are staffed with designers that each have their own local workspace that can be either isolated from each other and the central repository or can be automatically synchronized with the central repository and IP developers that develop the components that can be re-used by the design teams and are delivered to the central repository. The project or group manager can keep track of the developments by automatically generating reports from the change management databases. Developers that join one of the projects can have their workspace set up automatically and start working right away.

Distributed or multi-site development is getting more and more common. It can be that remote login or the use of a VPN is sufficient to support the distributed development and most revision control software will support limited control using HTTP, but for larger organisations, it is likely that the central repositories must be replicated over the different sites. There are a number of options that need to be considered to find the right implementation for a specific organization.

After all, for an engineer, the best thing a data management tool can do is stay out of the way and perform its activities as transparent as possible.

Dizain-Sync data management services

As shown above, re-use and team-based design are of increasing importance for ASIC, SoC and FPGA design groups.

Dizain-Sync offers the following capabilities to help existing design groups with setting up or changing their data management strategies and processes to keep-up with this increasing demand.

Dizain-Sync services can aid with:

- Development of business cases on DDM¹, PDM and PLM.
- Requirements gathering for tools and processes.
- Tool selection on PLM, PDM and DDM.
- Business process improvement.
- Getting Awareness and unity
- Training on PLM, PDM and DDM

Furthermore, when a higher level approach is required Dizain-Sync can perform *assessments* and deliver *solution services* tailored to each individual customer:

- Assessments
 - Business Awareness assessments
 - PLM Scan
- Solution Services
 - Requirements gathering in the areas of DDM, PDM and PLM
 - Business process review/optimization and documentation
 - Software vendor selection (in the areas of DDM, PDM and PLM)
 - Implementation services
 - Project Management
 - Bridge function between vendor and customer
 - Training

Dizain-Sync has experience with all major FPGA and ASIC development tools:

- Cadence: DFIL, Incisive
- Synopsys: Design Compiler, VCS, TetraMax, Galaxy and Milkyway databases
- Mentor Graphics: EPD, HDL-Designer, Modelsim, Seamless, Precision, Fastscan, Platform Express
- Synplicity: Synplify FPGA/ASIC
- CoWare: SPW, ConvergenSC
- Xilinx: ISE, EDK
- Altera: Quartus, SOPC Builder

Dizain-Sync has experience with the following data management tools and solutions:

- MatrixOne/Synchronicity DesignSync
- IBM/Rational ClearCase and ClearQuest
- Telelogic CM-Synergy
- Perforce SCM
- Oracle database servers
- RCS
- CVS
- Subversion
- Trac

¹ Design Data Management, the term used here to indicate data management specifically tailored to a design activity or tool (e.g. PCB or IC design).