Strategies for Managing Data Across Multi-Site Design Projects

Contributor: Integrated Device Technology, Inc. (IDT)

Printer friendly
E-Mail Item URL

August 5, 2009 -- In today's global economy, many large engineering companies have design teams spread out all over the world, and these teams often collaborate on design and development projects. Managing multi-site projects is difficult enough without having to deal with issues related to incomplete, out-of-sync or simply incorrect data. With sub-90-nm design gate counts and complexity ever increasing, the sheer file count and size of design databases is exploding. Manual methods of transferring files back and forth and then attempting to integrate them back into design databases are highly error prone, and can lead to errors, which can, in turn, result in huge costs and loss of productivity.

Defining a strategy to handle seamless data transfer and synchronization on a project is often overlooked, but a good solution can greatly facilitate efficient multi-site design collaboration.

Multi-site design engagement model

Multi-site design is successfully accomplished in many ways and, depending on how the design engagement model will be defined, different data-management strategies can be employed. However, if there is a possibility that the engagement model could change, then this should be considered up front.

Design complexity today routinely reaches into the millions of gates, involving many thousands of data files and an extensive data-dependency tree. If even one file in the design hierarchy is not the correct version, the ramifications can be huge. This is particularly true when blocks are in flux and have not been completely locked down, since many files are constantly changing. For example, consider the situation where constraints or a floor plan file are out-of-sync. Any resulting synthesis or place-and-route runs will be incorrect. If the out-of-date issue is not

Search for:
Site | Current Category
Go | Search Options

found quickly, a lot of wasted time and effort could occur.

If each site is going to be working fairly independently of the other (Figure 1), then most of the data can be transferred once, and the remote site can simply send back a finished product. This is the traditional "throw it over a wall" method, and is the easiest method of collaboration. However, if multiple sites plan to work on a block simultaneously — for example, logic design is done at site A and functional verification at site B (Figure 2), a much tighter data sharing model is needed. As mentioned earlier, even one file being out of sync in the RTL tree here can result in errors and a lot of wasted time.

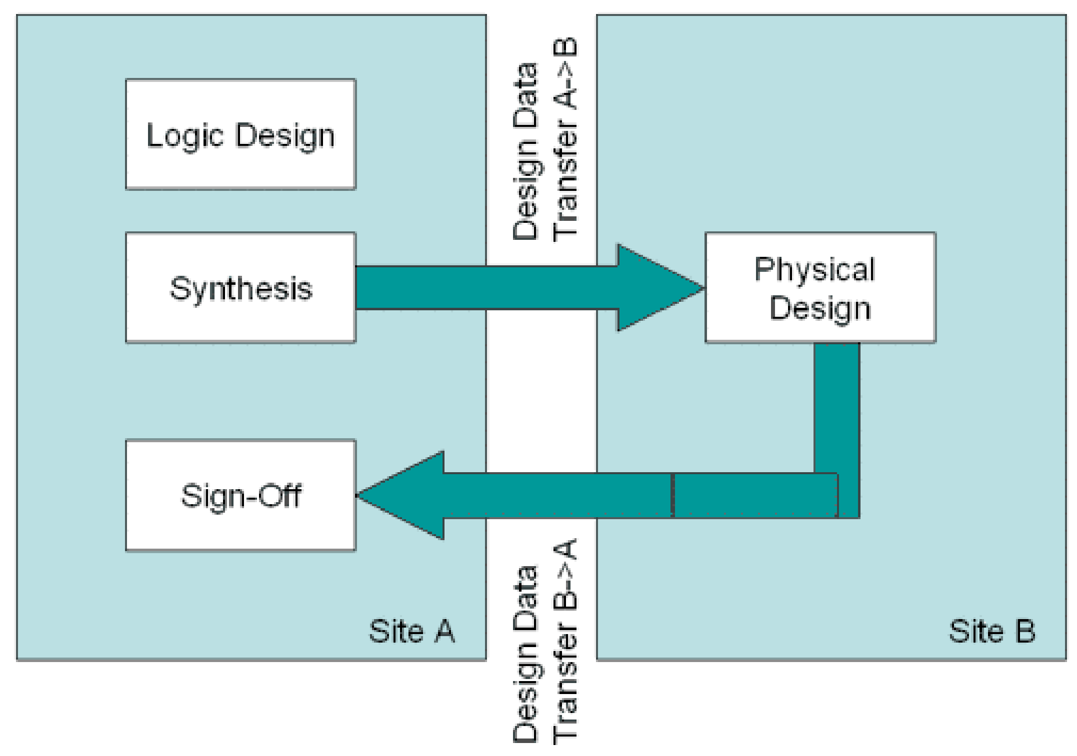

Figure 1. Simple "over-the-wall" data-sharing model.

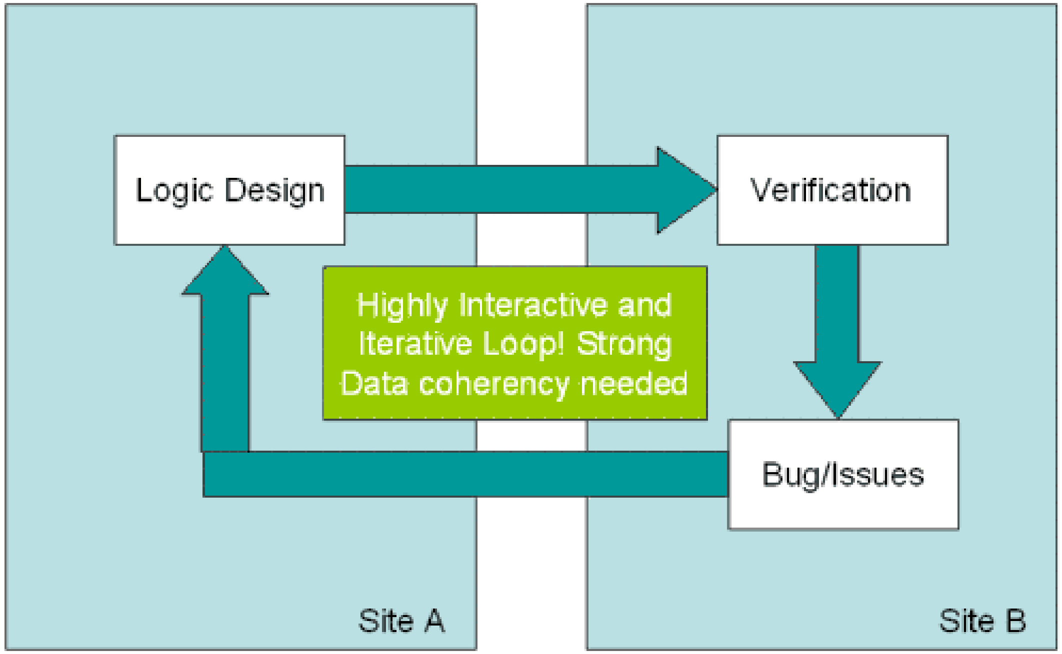

Figure 2. Interactive data sharing model.

Export control considerations

One of the primary issues that must be considered when working with international teams is compliance with export control. Project managers need to work with their legal department to determine what can and cannot be legally transferred to the sites in different countries, even if they are part of the same company. In some cases, you may be able to obtain a clearance for the entire project, thus making the overall process much simpler. In other cases, only parts of the design may be transferable, and mechanisms must be put in place to prevent the unauthorized parts of the design from being transferred. This should be hammered out before engaging on a multicountry design project to prevent any legal issues.

Communications

A very important factor for successful multi-site design is a good means of effective communication. This may seem very obvious, but experience has shown that it is absolutely critical, particularly when dealing with people of different cultures and whose native language is not the same as yours. Verbal communication with no written follow-up is almost guaranteed to result in some kind of mistake or misinterpretation. Telling the remote site over the phone to verify block A with version B, with the exception of using sub-block C version B-1 is almost guaranteed to cause problems. It is critical to put in place systems that facilitate written communication. Simply using e-mail is not enough since it becomes difficult to track current status and issues, and sometimes important team members are left off the e-mail threads.

For bug tracking, particularly if logic verification is being done at a remote site, a system such as Bugzilla should be put in place. It is likely that you already have a system like this in place, but it may be beneficial to see if it can be ported to multiple sites. If not, the remote site should be able to log into the system via VNC (Virtual Network Computing). For general communications, a Web site should also be set up with some kind of collaboration tool, such as a Wiki system or SharePoint site.

In-person meetings are invaluable, but in today's cost-conscious environment, this can be difficult to accomplish. Video-conferencing systems are a reasonable alternative, and these can be installed fairly inexpensively. Indeed, large corporations invest in high-end systems, such as Cisco's Telepresence, but lower-end solutions are also available.

Data transfer methods

Multiple ways exist to transfer design data to remote sites, and they each have their pros and cons. Some of the more popular methods are summarized below.

E-mail — This is one of the most prevalent methods used for data transfer. Engineers like to simply tar/gzip files, attach them to an e-mail and send them off. This method has the benefit of ease, but is not advisable for security and data-synchronization reasons. Security can be addressed by encoding the attachments using a method such as PGP, and this can actually be set up to work with e-mail systems transparently; but check with your IT department.

FTP — Similar to e-mail, FTP is fairly easy to use and is often employed when large amounts of data need to be transferred. This method also is not advisable for data-synchronization and security reasons. There are many secure FTP clients that offer strong encryption that should be investigated if you plan to transfer any kind of sensitive design data. For very large data files, FTP can sometimes create difficulties, so it is advisable to break up the files into smaller pieces. The utility "split" on Linux-based systems can be used to accomplish this for binary- and text-based files.

The example below splits a large file into a number of small files, with each one limited to 20MBytes. The subsequent "cat" command recreates the original file.

```
split –b20m Largefilename Smallfilename
cat Smallfilename* > Largefilename
```

Rsync/RCP — This utility is standard on most Unix- or Linux-based workstations. It allows for data synchronization between files and directories trees. The utility is very robust, and it works by computing differences in files and then sending only differences when it can (for text files). This system is often used for push synchronization, mirroring of tool installations, or design kits within an organization, which are modified by one group of people, but then read by many. It can also be used for project data synchronization, but it can be rather slow to run for large projects, unless run at more granular levels. Rsync also does not have all the benefits of a full-fledged version control system, in that it lacks basic functionality, such as change tracking and access control.

Version control/ design data management systems — Use of a design management (DM) system is the most comprehensive way of achieving data synchronization on a project spread out among different sites. There are multiple DM systems available currently, both commercial and open source, and they need to be evaluated based on your needs. When deployed on a multi-site project, these systems create local caches of design data to speed operations (Figure 3). As items are checked in or out, local cache servers are queried and updated with

Archived Webcasts

Newsletters

About SOCcentral.com

Sponsorship/Advertising Information

the latest data. This architecture is fairly high performance, except for the initial data population, and it can be configured to run in almost real-time. As a result, there is very little overhead from using a DM system on a day-to-day basis.

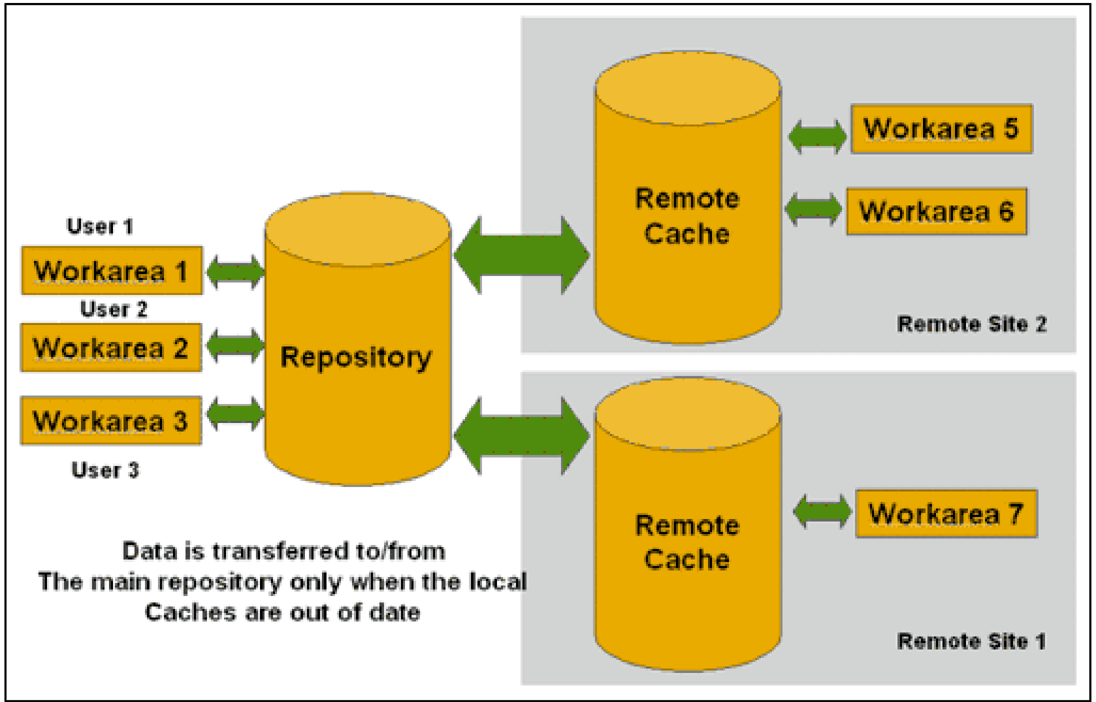

Figure 3. Design management system remote site data caching model.

By using access control lists, DM systems can also be set up to allow or disallow directories or blocks from being checked out by certain groups of users, thereby simplifying implementation of export control restrictions when required. Other useful features and benefits from using DM systems include tagging, making design snapshots, tracking changes and maintaining a history of the design and the ability to recreate the design state at any point in time.

This method is recommended for all projects where even a small amount of interactivity is required. It does take some amount of setup time, and may require IT involvement to open some data ports. But, once established, it can easily be scaled to multiple projects and additional individuals. It should be noted that for some teams, the use of design-management systems is a new idea, so some training would be involved to introduce them to the overall concept and data model.

At IDT, we used a design data-management system for managing custom and RTL trees for a 65-nm SOC between sites in San Jose, Canada and Atlanta. There was some initial setup work required for pipe-cleaning, but beyond that, the project data has been flowing between sites without issues. This system has essentially eliminated any issues with incorrect or incomplete data, and has made the overall project data management essentially seamless.

In summary

Today's global economy dictates that engineering work be increasingly distributed, and certain tasks will flow to lower-cost areas. Being able to capitalize on that fact and having the systems in place to facilitate the distribution and successful completion of work will greatly benefit an organization, both from a cost and schedule perspective. Two of the most important issues are effective communications and management of the flow of data. Evaluating and setting up both of these systems can take some time at the outset, but they can prevent many issues down the road.

By Ravi Poddar

Ravi Poddar is the Director of CAD at Integrated Device Technology, Inc. (IDT). His responsibilities include methodology development and support of custom digital and analog mixed-signal design flows. Ravi has been at IDT for almost two years. He holds a Ph.D. in Electrical Engineering from the Georgia Institute of Technology.

Go to the Integrated Device Technology, Inc. (IDT) website to learn more.

Keywords: ASICs, ASIC design, EDA, EDA tools, electronic design automation, design management, design collaboration, Integrated Device Technology (IDT),

488/29434 8/5/2009 3879 3879

Add a comment or evaluation (anonymous postings will be deleted)

Meridian Constraints is a comprehensive constraint-management solution that offers high-performance constraint validation, template generation, coverage analysis, equivalence checking and timing exception verification capabilities designed to provide users with ultimate confidence in the timing constraints employed across all phases of the implementation flow.

0.953125

Go back

Back to Top

553.488 1.015625