Data Management in Large Digital System Designs

John Tramel

CECS 460 CSULB

Large Systems Demand Good Methods

- Large designs may include the work of many engineers and perhaps even engineers from many companies
- To create a design that may be verified, tested, and reliably manufactured there is the need to impose standard practices on the design teams in order to assure the success of each stage of the development
- This discussion will focus on management of the design components - mainly files, and the incremental deliverables - builds

Revision Control - Files

- Revision control is also known as version control, source control, source code management, etc.
- Revision control is the management of changes to documents, programs, and other information stored as computer files
- It is typically found in software development but with the advent of HDLs has been incorporated in the the hardware development process as well

http://en.wikipedia.org/wiki/Revision_control

Revision Control - Controls Files

- The target environment is where a team of engineers are working together in developing an ASIC or SoC
- The design product will be the many files that are produced along the way: source code, scripts, environments, simulations, documentation, etc. - our raw material is not iron or steel, it is computer files
- It is the control of these files that is the subject of revision control

Revision Control Tool and Use

- Typically revision control is an application from a third party that is utilized by a team, it may be like SCCS, which was included with the UNIX OS
- Each file is required to have a header/footer that is updated by the revision control system to indicate what is its current release
- These may be indicated by a revision number 0.12. Where the release is still zero (0), but this file's revision is 12.

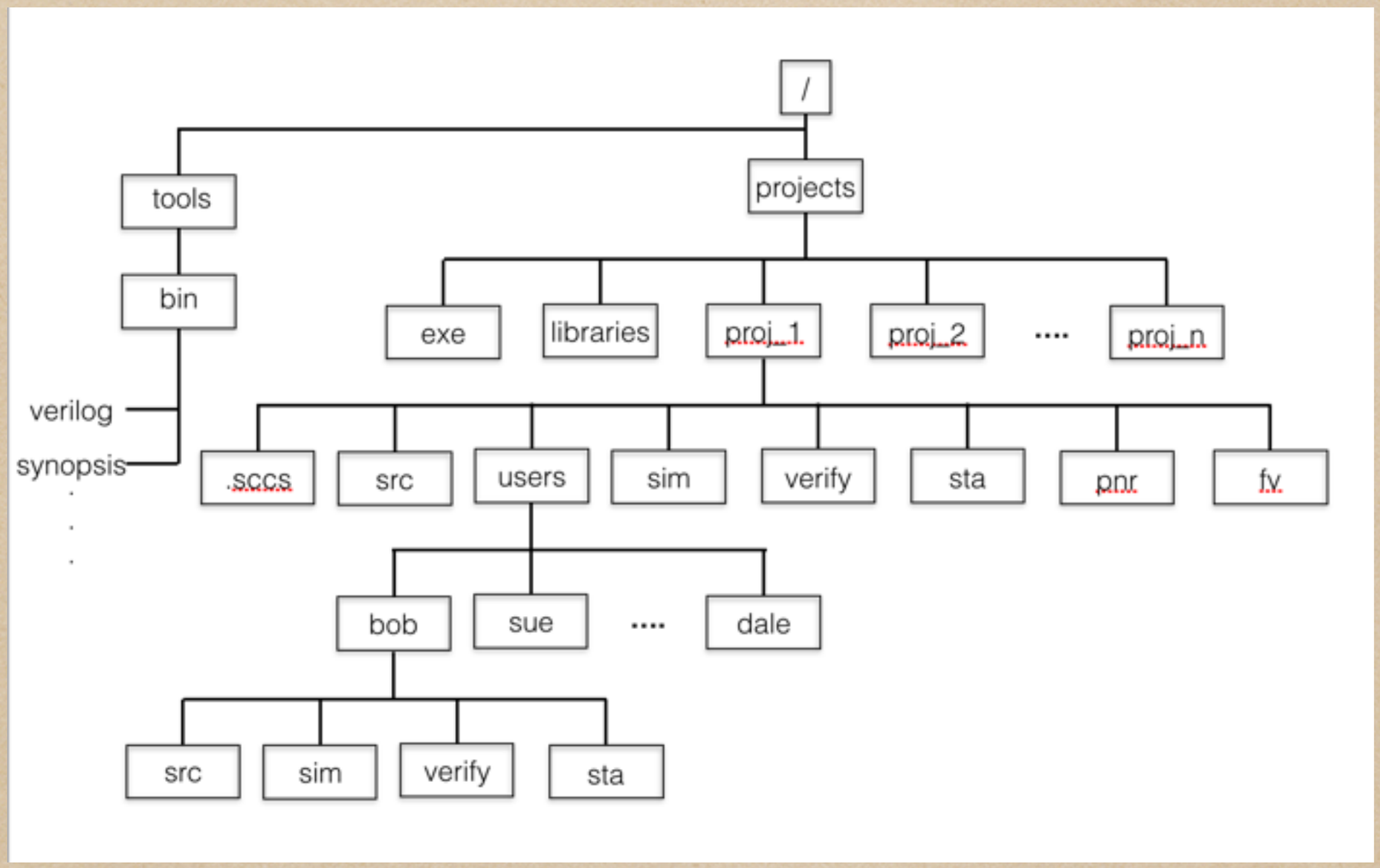
Revision Control Tasks

- Typical features of a revision control are 1) the ability to view a read-only version of the file (no accountability), 2) check out an editable version of a file, 3) check in an edited version of a file (thus incrementing the revision 4), locking a file (so no one can edit it), 5) allowing the check out of a previous version of a file, and many more sophisticated tasks
- The current state of a design is comprised of the current revision of each file in the system

Project Data Organization

- Each project in a large organization will need to set up a directory structure in order to "house" and "control" all of the work that will be done on their project
- Typically this is initiated with a request to the IT organization to dedicated x gBytes of data to a project
- This also implies that the IT organization is periodically backing up the design file system in the event of a catastrophic failure
- The frequency of backing up is a function of the tolerance to lost data: if you can survive a weeks lost work, then back up one a week.
- Most organizations are willing to accept the loss of a day's work so their systems are backed up nightly

Sample Project Directory Structure



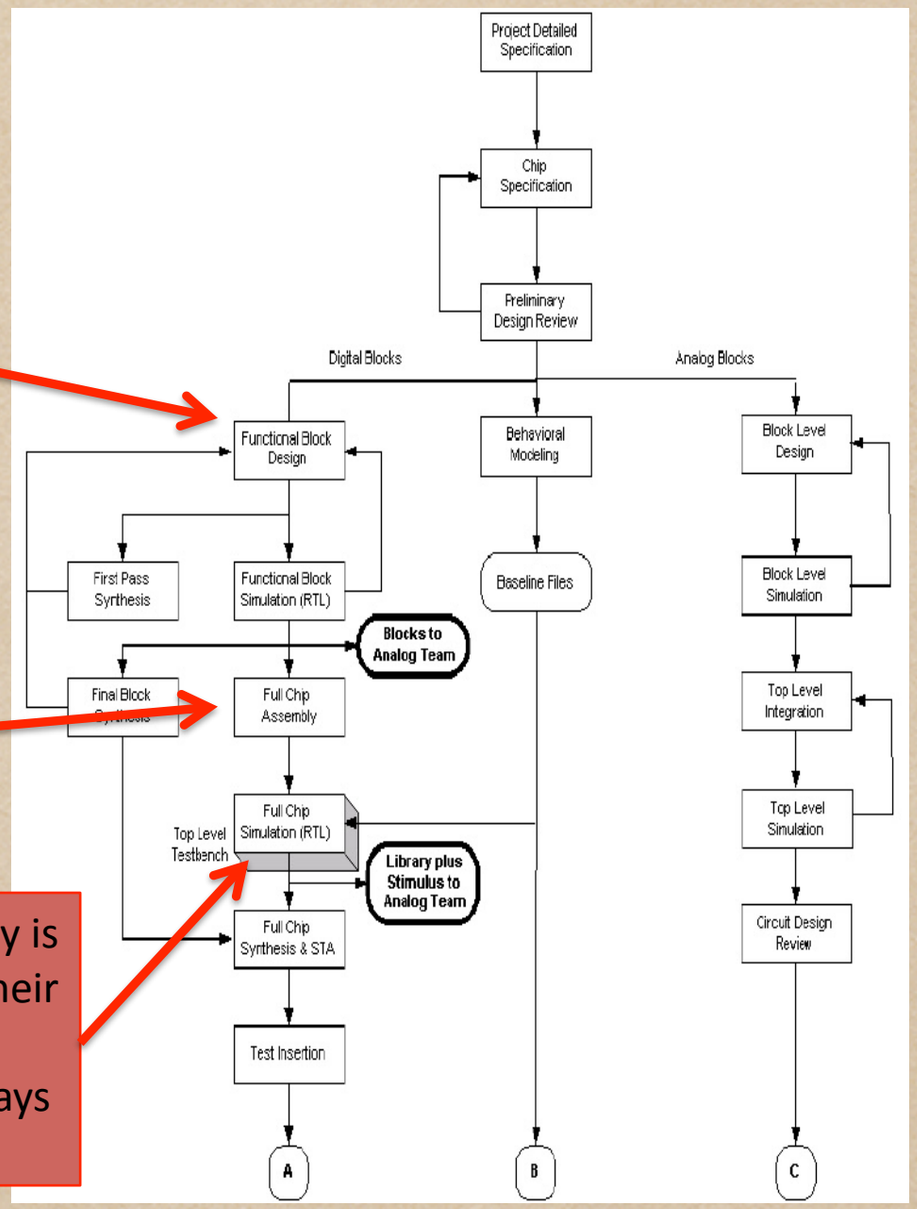
- This is meant to be "representative" not "comprehensive"
- For further information here refer to Reuse Methodology Manual

Source Code Management in the Project Environment

Work here is done in each user's own directory

Work here is done in project src (source) directory

Source code in the project source directory is managed by the project "librarian". It is their responsibility to make sure that the simulations at full chip always pass – "always working model"



Always Working Model

- Once the project has taken all of the individual designers' files, which each designer has verified themselves, then these files will be collected in the chip src directory to represent the top level design
- This top level design is verified by the independent verification team that has created a test environment along with a set of scenarios to check the design
- The first goal of integration is to get the top level design working correctly with the existing set of verification scenarios
- The project librarian has the responsibility to ensure that all of the team members have a working environment in which to develop/refine their designs by making sure that all the scenarios pass when checking the files in the source directory

Always Working Model - Continued

- Since each individual designer is able to continue developing their block, there needs to be a way to allow this without disturbing the other engineers
- This is accomplished by allowing the simulations (scenarios) to be run against the released source code - with their modified code included, in the designer's own directory
- Once a designer believes his code is "clean" he will then submit the code to the librarian to verify its compliance with the verification environment and if all is well to be included in source
- Each night the librarian will initiate a "regression" run where all scenarios are executed against the released code and if all passes the submitted code is "promoted" to the source directory

Release Control or Configuration Control

- In the software world there is the concept of a "build" that implies the release of the software to a certain level of acceptance
- The same analogy is true with hardware design, there is the need to periodically baseline the design by controlling a release of the design
- One technique is to utilize the revision control system to set all files to the same release - 1.0, 2.0, etc. depending on which release you are at
- By setting a baseline you establish a working standard that may be used - build 1.0, build 2.0, etc.
- The need to control the code that comprises a design and the control of the various builds can not be overstated