Electronic Design Automation (EDA) challenges in verification

Tuna B. Tarim
WW Design Automation, Analog EDA, Texas Instruments (TI), Inc.
21 April 2014, Istanbul, Turkey
Agenda

• Analog EDA organization at TI
• What is verification?
• Challenges in verification
• How is EDA helping?
• Closing comments
Analog EDA organization at TI
WW Analog organization

- High Performance Analog
  - Audio / Imaging Products
  - High-Speed Products
  - Interface & Clock Products
  - Medical / High Reliability
  - Precision Analog

- High Volume Analog & Logic
  - Integrated Products
  - High Volume Linear
  - Mixed Signal Automotive
  - Standard Linear & Logic
  - Storage Products
  - Motor Drive

- Power Management
  - Advanced Low Power Solutions
  - Battery Management Solutions
  - DC Solutions / FET
  - Power Supply Solutions
  - Linear Power

- Silicon Valley Analog
  - Power Products
  - Mobile Devices Power
  - Signal & Datapath Solutions
  - Precision Signal Path
Analog Design Services (ADS)

• ADS has five organizations to serve the needs of WW Analog teams:
  – Embedded Platforms
  – Analog Packaging
  – Design Support Services
  – WEBENCH®
  – Analog EDA
    • WW Design Automation
    • EDA Execution & Support
    • System Design Automation
    • EDA Supplier Strategy
### WW Design Automation

<table>
<thead>
<tr>
<th>Category</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Analog and Mixed Signal Flow Development</td>
<td>• Development and release of TIs common analog and mixed signal design platform</td>
</tr>
<tr>
<td>Simulation Development</td>
<td>• Definition and development of tools related to analog and mixed signal design and verification</td>
</tr>
<tr>
<td>Custom Development</td>
<td>• Definition and development of internal tools that are common across Analog design teams</td>
</tr>
<tr>
<td>Design Methodology</td>
<td>• Definition and development of design methodologies that are common across Analog design teams</td>
</tr>
</tbody>
</table>
What is verification?
What is verification?

• Design verification is “assuming that the design is broken until proven otherwise”

• Having the discipline to run design verification results in more on-time tapeouts, fewer silicon respins, and a reduced chance of customers finding silicon problems

• Most design groups start design verification towards the end of a project. Advanced verification pays off by identifying issues and/or potential issues in the design earlier in the project and therefore allowing the design team plenty of time to address these issues
What happens when we don’t do a good job in verification?

• Silicon respins $\rightarrow$ $$$
• Yield loss $\rightarrow$ $$$
• Test time increase $\rightarrow$ $$$
• Embarrassment with customer $\rightarrow$ $$$
• …
Challenges in verification
Challenges in chip level verification

• The objective of chip level verification is to ensure that the design operates correctly for all possible functional scenarios
  – List these possible scenarios → Verification plan
  – Select test cases and check if the results are correct → Regression testing
  – Ensure that sufficiently good testing is done → Coverage
Challenges in system level verification

• The following testing is needed during system level verification (beyond individual blocks):
  – Testing of IP integration
  – Testing of IP-to-IP interaction
  – Testing of power
  – Functionality/performance of the entire system
  – Functionality/performance of the shared resources such as memory blocks
How do we address these challenges
How do we address these challenges?

Independent design verification (DV) engineers

• Must have independent DV engineers who are dedicated to verification from the start of the project

• Become more than familiar with running a variety of EDA tools:
  – Simulators (e.g.; Spectre, AMS Designer, Ultrasim)
  – EMIR analysis (e.g.; VPS)
  – Behavioral modeling (e.g.; Verilog AMS, Verilog-A, VHDL-AMS, …)
  – Change management (e.g.; DesignSync)
  – Variety of reliability checks
  – …

• Become familiar with the system level and the specification to eliminate assumptions and surprises
How do we address these challenges?

Putting together a verification plan (1)

• One of the critical challenges today is that because of lack of resources and time pressure to complete the project, the test plan is prepared towards the end of the design cycle → Too late to run a complete set of design verification simulations

• The test plan for the chip level needs to be put together early in the design process

• This test plan needs to be run starting early in the project
How do we address these challenges?

Putting together a verification plan (2)

• Verification plan should include the following:
  – All usage scenarios
  – Power-up and reset
  – Test mode
  – Breaking the part
  – System level scenarios
  – Various checks (floating nodes, MOS overvoltage, forward-biased diodes, topology checks)
    • Other considerations (communication with adjacent chips, provide behavioral models to customer, board and system parasitics, etc.)
How do we address these challenges?

Putting together a verification plan (3)

• Spend enough time making a verification plan in order to
  – Estimate the type and quantity of testbenches needed
  – Estimate the number of skilled resources to execute the plan
  – Estimate the schedule dependencies between verification, design team, systems, and customer
  – Estimate any new tools, features, or methods needed to properly verify the device

• Verification plan can be in the format of a specific vendor tool, an internal tool, or simply an Excel or Word file

• This verification plan will then be used to create a regression suite
How do we address these challenges?

Regression testing (1)

• Regression testing is how bugs are found in analog verification
• It is through regression testing that the function of the chip is tested and verified, and bugs are found
• The critical aspect of design verification is putting together the regression tests
• Example: A DAC that has 10 inputs, 5 control bits, 1 bias input, 1 supply line, and 1 output will require over 2500 automated checks to be performed to verify that it fully functions
• Running each check manually is time consuming and inefficient
How do we address these challenges?

Regression testing (2)

- Regression testing will help
  - In/out pins and their conditions will change in the test suite
  - A pass/fail criteria will be set for each test
  - Errors are detected and error messages analyzed
  - Example: Without regression testing individual would have to visually inspect the output waveforms to validate that the design is working correctly
How do we address these challenges?

Coverage

• A large amount of testing needs to be done to ensure all scenarios are tested and failures analyzed
• The higher the coverage, the higher the confidence that the chip works
• Automation will help with running a large number of tests efficiently
• Automation will also help with analyzing the results without
How do we address these challenges?

Miscellaneous – Behavioral modeling (1)

• DV needs to be done fast and the models need to be accurate enough to represent the schematic

• Using behavioral models instead of the transistor level schematic will help speed up the simulations

• Behavioral modeling language needs to be selected carefully

• Modeling detail needs to be decided carefully (detailed enough so as not to slow down simulations)
How do we address these challenges?

Miscellaneous – Change management (CM) system

• CM within Cadence
  – DesignSync → More integrated into Cadence
  – ClearCase → Not as integrated into Cadence, can be difficult to use

• CM for digital Verilog, text files, scripts, …
  – ClearCase → Very well featured and easy to use
  – DesignSync → Not as well as Clearcase but if you are using DesignSync for analog designs, consider using it for digital to keep with one CM system
How do we address these challenges?

Miscellaneous – Bug tracking system

• Use a bug tracking system during DV
  – Bug tracking tools are available from multiple vendors
  – Many companies have bug tracking tools developed internally
  – Excel spreadsheets can also be useful for this

• Need to track bugs found and fixed during DV cycle. Helps show convergence to stability and risk to tape out

• Example bug status definitions:
  – **New:** Found and entered into the queue, not yet assigned to owner
  – **Assigned:** Waiting for the owner to fix, reject, or ask for more information
  – **Fixed:** Owner has applied design or spec fix, waiting to be re-verified
  – **Verified:** Originating engineer verified the fix is correct. Test and check remains in regression suite permanently
  – **Closed:** Appropriate action taken for the bug and item closed
  – **Rejected:** Was originally thought to be a bug but was later proven to be a false-alarm. No design or spec change
How do we address these challenges?

Last minute changes

- The goal is to verify all design changes even if they are made hours before tapeout
- Need the ability to quickly verify any change
- Need as much automation as possible
How is EDA helping?
How is EDA helping?

- Provide an easy-to-use tool for the DV engineer to enter the usage scenarios for the verification plan.
- Provide a regression flow for running the scenarios in the verification flow over corners while sweeping variables.
- Provide a method to quickly and easily post process the results, and analyze the pass/fail results.
- Still missing:
  - Three sets of data needs to be compared side to side and checked to see if they are matching: Specification data, simulation results, test results on actual silicon.
  - The capability to compare spec data and simulation data is available, though not fully automated.
  - The capability to include the test results on actual silicon to the comparison is a challenge because of the different nature of the tools in the IC world and test and characterization world.
How is EDA helping?

• Development of a bug tracking tool for design, design verification, characterization and test

• Development of internal tools to check for floating nodes, MOS overvoltage, forward-biased diodes, topology checks, etc.

• Development of a tool to compare behavioral models against their schematic

• A flow that will allow quick turn around for last minute changes
Closing comments
Closing comments

- Design verification continues to be a challenge for mixed signal designs.
- It is definitely worth considering having a dedicated team for design verification to ensure detailed simulations are run and an objective analysis of simulation results is done.
- Automation helps design verification become more efficient.
- EDA provides the methodologies, flows and tools to ensure quality and on-time design verification.
- Remember: First pass success $\rightarrow$ No respins $\rightarrow$ $$ savings.