Unification of Pre- and Post-Silicon Verification Flows in Mixed-Signal Designs

Stefan Tauner, Dominik Widhalm, Martin Horauer

Josef Ressel Center for
Verification of Embedded Computing Systems

Department of Embedded Systems
University of Applied Sciences Technikum Wien
Höchstädtplatz 6, 1200 Vienna, Austria

Abstract

In recent years rising complexity, reduced design cycles, and shrinking silicon feature sizes pose new challenges on the verification of modern system-on-chip solutions. To tackle these issues various design and verification languages, as well as methodologies and tools were introduced. Likewise, new and better physical process models allow for improved simulation of both analog and digital designs. Despite all these efforts, however, many problems exist in industrial state-of-the-art processes and tools.

In this work, some insights are given on lessons learned from the design and verification of a recent automotive microcontroller, a complex system-on-chip solution. Based on these findings, a new verification flow is proposed that closes an identified gap between pre-silicon and post-silicon verification. To that end, a common test description language is introduced that can be used in various development environments such as simulation and post-silicon testing.

Motivation

- Gaps between . . .
  - pre-silicon simulations and post-silicon tests,
  - post-silicon verification and customer.
- Respective development stages treated distinctly by separate teams/tools
  → colaboration and sharing tests difficult

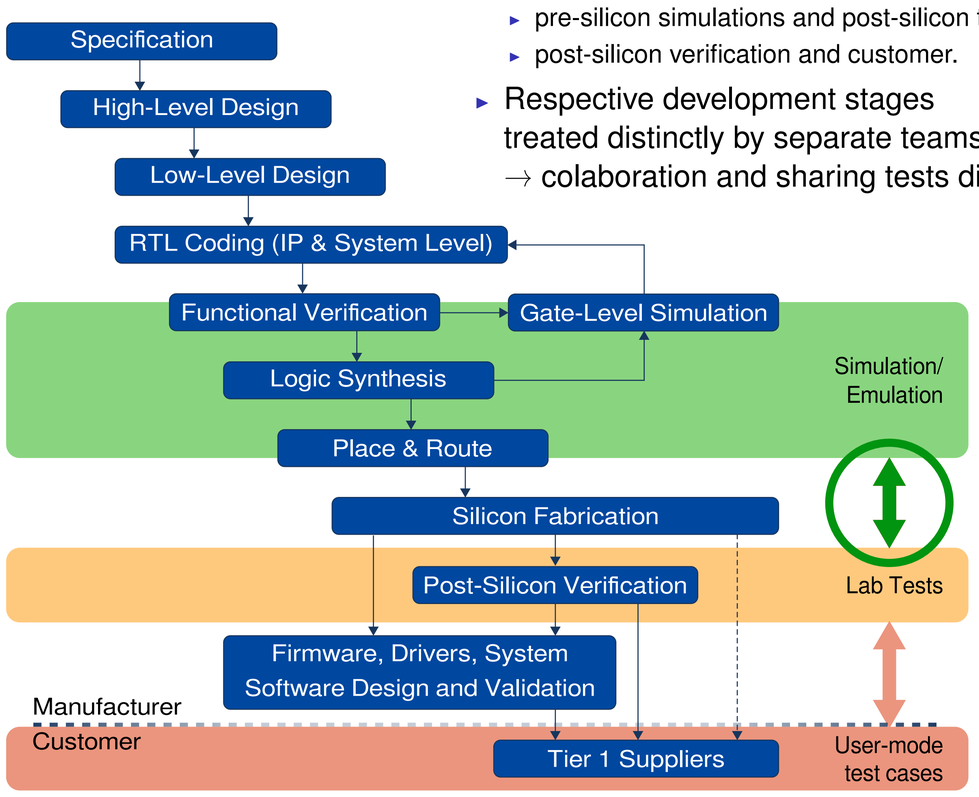

Figure 1: Typical design flow of mixed-signal designs.

Approach

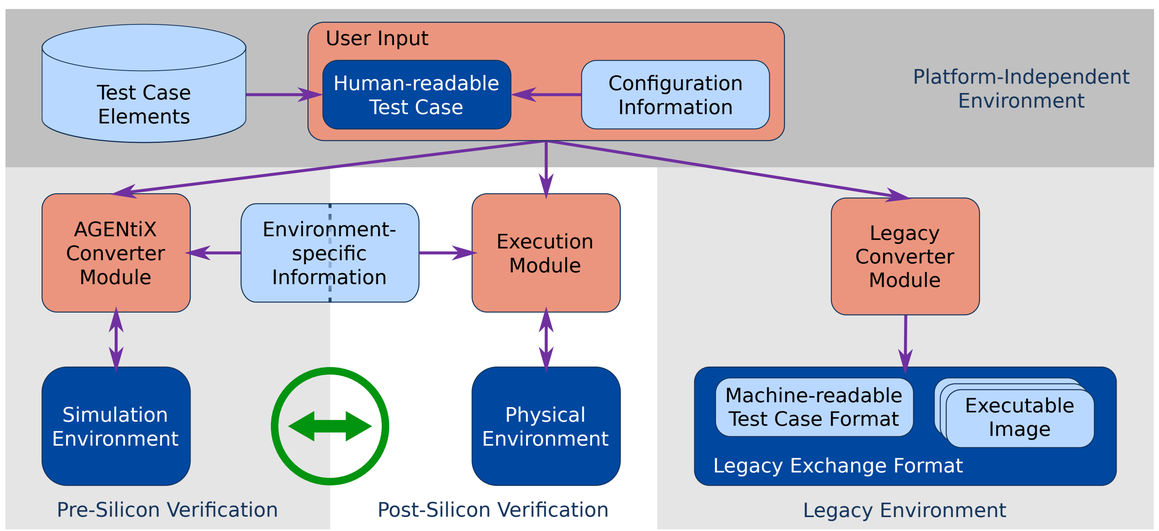

Figure 2: Overview of our approach.

- Common abstract test description language
  - Based on existing scripting language (e.g., Perl)
  - Allows engineer to write procedural test flows
  - Provides full expressiveness of underlaying language
- Single human-readable test case
- Test applied to respective environment automatically
- Environment-specific settings possible (e.g., pin mapping)

Implementation

Common:

- Test cases execute sophisticated sequences of commands
  → flow control needed
- Existing scripting language allowing to utilize complex data structures → Perl
- Test interpreter acts differently depending on environment
- Hardware abstraction segregates common parts from environments

Pre-silicon:

- Simulation environment completely mimics surroundings of DUT
  → requires cross-language interface

Post-silicon:

- Laboratory equipment controlled via GPIB stimulates DUT
- Communication with DUT via debug interface

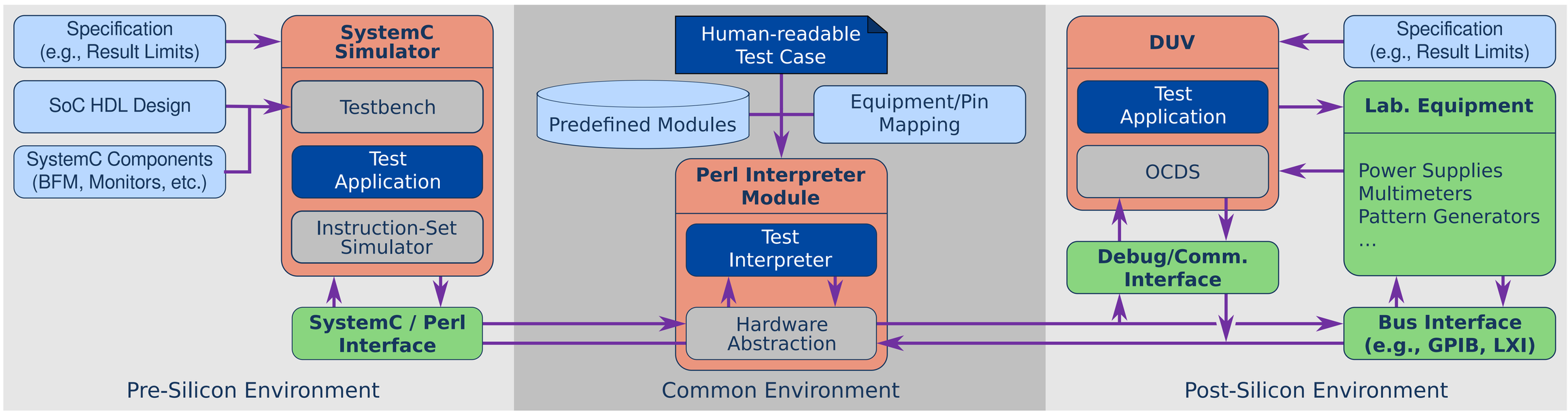

Figure 3: Schematic of the data/control flow in the two environments.

Summary

Although, standard flows are well established for the design and verification processes, there is a lack of suitable concepts to link them. The proposed concept shares test cases between different pre- and post-silicon verification activities. A common test description is introduced that can then be automatically converted to environment-specific test cases. This offers the advantage that test cases need only to be written once and can then be commonly used in different verification activities and environments. The feasibility of the proposed approach was verified using some predefined test cases.

Acknowledgment

This work has been supported by Infineon Technologies Austria AG and by the publicly funded R&D project Josef Ressel Center for Verification of Embedded Computing Systems (VECS) managed by the Christian Doppler Gesellschaft (CDG).

Contact

{tauner, widhalm, horauer}@technikum-wien.at