

TORQUE COMMUNICATIONS PVT. LTD.

◆ Delhi ◆ Mumbai ◆ Bangalore ◆ Ahmedabad

Website:	www.techonlineindia.com
Date:	15 February 2010
Link:	http://www.techonlineindia.com/showArticle.jhtml;jsessionid=1ETN1FXUK5JANQE1GHPSKH4ATMY32JVN?articleID=222900430

Guidelines for complex SoC verification

A plan-manage-execute approach to verification

By Jignesh Oza, eInfochips

[TechOnline India](#)

(02/15/10, 11:28:00 PM EST)

Almost 60-70% of time in the ASIC cycle is occupied by Functional Verification and so, the main aim of this paper is to provide overall guidelines in verification. More specifically, on the adoption of various planning strategies, managing the dynamics in projects and a metric-driven execution approach with the maximum possible automation and reusability that helps deliver a quality product on time and achieve silicon success.

Inevitable changes

Consider the example of a typical SoC, consisting of a processor, several IPs, Direct Memory Access for data control, a common bus matrix for data transfers and inter-block communication and a system memory for data storage.

The requirements may change many times during the execution of a project. For instance, system memory size may be changed from X to Y to meet software needs. Third party IP may need to be changed, or a new IP or feature may need to be added. So, significant SoC functional specification change happens and we have to deal with adding, changing and removing features targeted for verification, updates in register definition and the like.

With above dynamics, a verification team has to tackle the following hurdles:

- How to or what to plan in verification?
- What type of execution flow model needs to be setup?
- How to manage SoC verification on time?

TORQUE COMMUNICATIONS PVT. LTD.

◆ Delhi ◆ Mumbai ◆ Bangalore ◆ Ahmedabad

Website: www.techonlineindia.com

Plan and manage

The real skill in planning is to adopt and to manage the verification plan which can cope with inevitable functional specification or requirement changes. There is a need to have a proper plan on automation and also to implement and manage reusable test benches.

Automation

Considering that there are significant changes in register specifications, planning to manually change the register tests will be a huge, time-consuming exercise. You need to plan for automation in which register contents are extracted from specification to your verification environment register model. You need to have an increasing amount of automation in place to cope faster with functional and requirement changes.

Below, in Figure 1, DMA_CFG register is shown as an example from the complete register set of specifications. PERL script is used to extract information from register worksheet file and convert into a register model file which is then imported imported by register model in the verification environment. Register model test library performs Read, Write and other application specific functions from the Register file set.

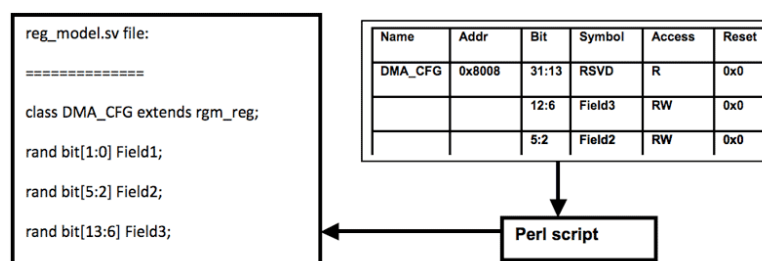


Figure 1

Click on image to enlarge.

Once the above flow is established, when there any changes in register fields or any additions/deletions of register contents, only the script has to be executed to get a completely updated register file. The same register model test library is used with updated register file to perform read, write and other application specific functions (may be minor changes) from the register file set.

Flexibility

Also plan to develop a verification environment which can have flexible and reusable test benches. Develop IP component test-benches in OVM (System Verilog), or eRM

TORQUE COMMUNICATIONS PVT. LTD.

◆ Delhi ◆ Mumbai ◆ Bangalore ◆ Ahmedabad

Website:	www.techonlineindia.com
----------	--

(Specman) methodology so that it can be reused in multiple verification environments and projects. Also, in the verification environment, adopt a transaction level modeling approach where, instead of dealing with signal level activities, you can try to capture at a higher level abstraction called Transaction, which DUT recognizes like packet transfer or set of instructions to perform specific tasks. This will ease component (Driver/Monitor/Generator) addition/removal/swapping in a flexible way as long as the interface is common.

Common data object (sequence item) should be used for data/packet generation, driving, collection and coverage. If features are added or removed in some data object, the information is updated and all components - generator, driver, monitor/collector and coverage models will get updated information.

Metric-driven execution

How do we ensure all features are covered? Are we ready to Tape-out?

To avoid manual efforts of tracking and measurement and doing re-work on changes, we have to switch from traditional test plans (where only set of test description is mentioned) to a coverage-driven verification plan (vPlan) with the help of higher level languages such as like System Verilog, Specman and tools like Vmanager.

Ideally, a verification plan should include all sets of features and the coverage goal associated with each feature. A high degree of automation is again needed to extract all features from specification and map it to the verification plan and the verification environment.

Tools like Vmanager should be used to manage regressions, failure analysis and generate functional and code coverage matrix from our verification plan to track and measure progress.

Automated, reusable SoC Verification Environment

In Figure 2 below, the SoC verification environment is shown and it has scripts for automation, vPlan with coverage goals, reusable test benches and a sample functional coverage report.

- SoC DUT is Design under Test
- Verification Environment comprises of reusable test benches, OVC--eVC, register and coverage models, monitors, test suite, score-board.
- Verification Automated script/macro is used to extract register contents from Specification and dumps it to verification register definition files which is imported in Verification Register Model (VR_AD, REG_MEM or RAL).

TORQUE COMMUNICATIONS PVT. LTD.

◆ Delhi ◆ Mumbai ◆ Bangalore ◆ Ahmedabad

Website: www.techonlineindia.com

- Also partial automation of extracting features from feature specification into vPlan is setup.
- vPlan is shown with all features to be tested and associated coverage point link from Verification Environment for tracking.
- Coverage report sample for progress tracking using Vmanager is shown at the bottom.

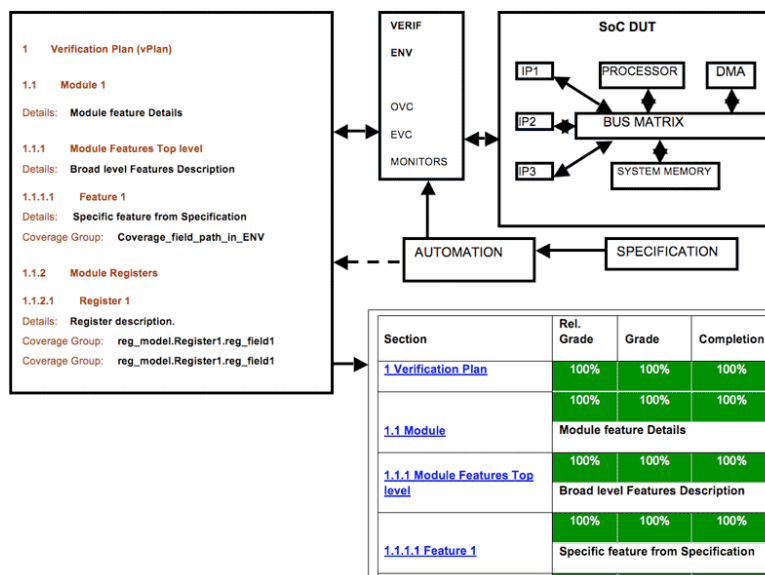


Figure 2

[Click on image to enlarge.](#)

Summary

How you plan, manage and execute SoC verification with proper automation, flexible test benches and tracking metrics plays a vital role in delivering a quality product on time in a constantly changing world. Also, a significant amount of time spent on verification can be saved when executing the next family of chipsets.

[About the Author:](#)

[Jignesh Oza is senior ASIC engineer, eInfochips Ltd. and can be reached at \[jignesh.oza@einfochips.com\]\(mailto:jignesh.oza@einfochips.com\)](#)