VMM BASED CONSTRAINED RANDOM VERIFICATION OF AN SOC BLOCK

Swathi M. Mohan and J. C. Narayana Swamy
Department of Electronics and Communication Engineering, BIT, VTU University, Bangalore, India

ABSTRACT

This paper describes a Verification Methodology Manual (VMM) based constrained random verification of a SoC block.The SoC block taken into consideration is Camera interface module (CIM). The environment of verification, which is created by means of verification methodology manual (VMM) for System Verilog, is scalable, predictable and reusableand can reduce verification time. Using this environment of verification, we can not only know whether there are bugs in the design under test (DUT) or not, butalso can easily locate design errors and corner cases with the help of constraints.

KEYWORDS:*VMM, System Verilog, Constrained random verification, CIM.*

I. INTRODUCTION

Verification is the single biggest challenge in the design of system-on-chip (SoC) devices and reusable IP blocks. Despite the introduction of successive new verification technologies, the challenges associated with verifying complex SoCs and IP persist [2]. The gap between design capability and verification confidence continues to widen. Repeated studies have shown that half to two-thirds of all SoCs fail at first silicon, with functional bugs a major reason. Anticipating all possible corner cases and discovering deeply buried design bugs is one of the key verification challenges. Given the realities of project resources and time-to-market demands, it is also critical to find these bugs as early in the process as possible and with as little effort as possible.

Traditional verification relies on directed tests, in which the TestBench contains code to explicitly create scenarios, provide stimulus to the design, and check (manually or with self-checks) results at the end of simulation. Directed testbenches may also use a limited amount of randomization, often by creating random data values rather than simply filling in each data element with a predetermined value. The directed test approach works fairly well for small designs, but a typical SoC design requires thousands of test cases and requires lot of time.

VMM takes SystemVerilog hardware verification language as itsfoundation. SystemVerilog [3] is the industry's first unified Hardware Description and Verification Language (HDVL) which can beviewed as an extension of the Verilog language with theadded benefit of supporting Object Orientated (OO) constructs.Verification engineers can take the advantage of the OO constructs to develop high level test scenarios with bothscalability and reusability.

The VMM for SystemVerilog [4] shows how to use the capabilities of the language to architect an automated test bench. By using the right strategies in setting up the verification environment to take full advantage of automation, the time it takes to create new tests can be dramatically reduced. Using constrained-random stimulus generation, scenarios can be generated in an automated fashion under the control of a set of rules, or constraints, specified by the user. The key is to craft the TestBench

such that additional tests can be derived from a relatively small set of base tests by simply modifying test parameters or adding and refining constraints.
In this paper the camera interface module is designed using Verilog and verification environment is carried out using Verification Methodology Manual based constrained random verification in system Verilog.
The rest of the paper is organized as follows: section 2 describes about constrained random verification, section 3 gives the overview of camera interface module (CIM),section 4 describes the hierarchy of the test platform used, section 5 explains the use of verification IP, section 6 gives the implementation of the results, section 7 is the conclusion and section 8 is future work.

II. CONSTRAINED RANDOM VERIFICATION

It supports the use of developing a set of test cases that use multiple seeds to implement several verification runs. The constrained random approach differs from the traditional verification flow whereby the engineer maximises the use of functional coverage to close off the task rather than build specific test cases to gradually close off on the verification task. Traditionally the verification engineer will use the verification plan to write directed test cases that exercise the various features of the DUT, thereby making steady incremental progress.
There is a downside to implementing the traditional approach. Specifically, it is necessary to write 100% of all stimuli being transmitted to the DUT. The technique is extremely time-consuming and each test is targeted at a very specific part of the design. Given that up to 70% of an ASIC lifecycle is spent in the verification task, any improvement in reducing that figure is warmly received within the industry. This quantifies the effort undertaken by the EDA tool development community in creating and supporting such methodologies.
Through the use of constrained random testing it is possible to reduce the time required to fulfil the verification task. A random test will often cover a wider range of stimuli than a directed test [6]. It can be manipulated by changing the seed on which the stimuli is created to yield a greater result from a single test scenario.

III. OVERVIEW OF CIM

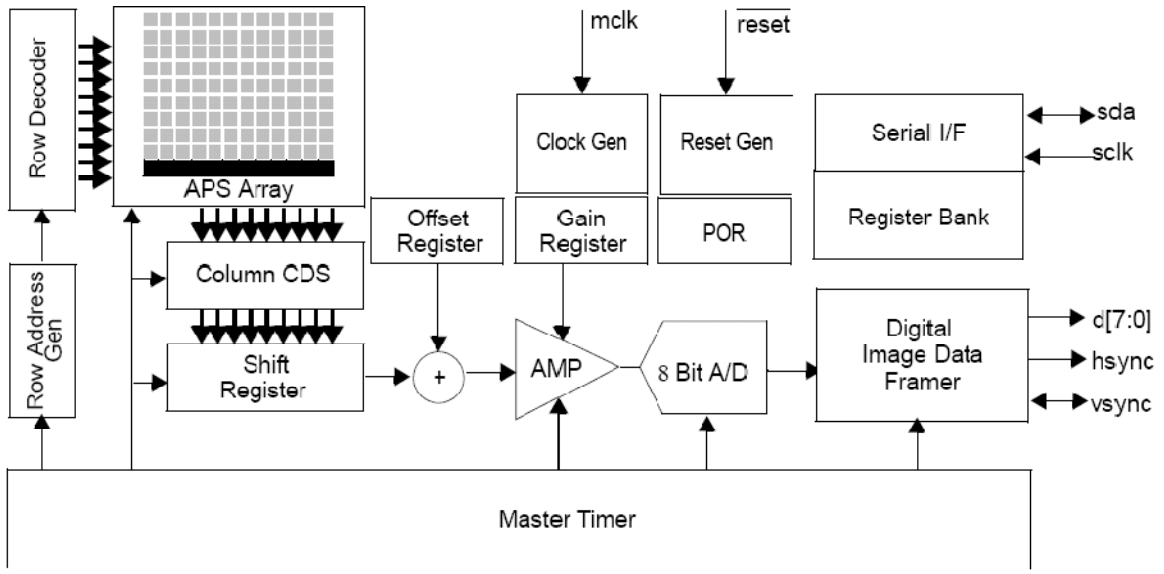

Figure 1: KAC 9630 architecture

Figure 1 shows the block diagram of DUT (design under test). The KAC-9630 is a high performance, low power, CMOS ActivePixel Image Sensor capable of capturing grey-scale images at 580 frames per second. In addition to the active pixel array, an on-chip 8 bit A/D convertor, fixed pattern noise elimination circuits and a video gainamplifier are provided.

The integrated programmable timing and control circuit allowsthe user maximum flexibility in adjusting integration time andframe rate. Furthermore, a fast read out circuit is provided allowing a full frame to read out on a single 8-bit digital data bus inless than 2ms.
The sensor utilizes anintegrated electronic shutter. This together with its ultra-high sensitivity makes the KAC-9630 an ideal choice for low light imagingapplications or applications where images of fast moving objectsneed to be captured with minimum motion blur.

IV. HIERARCHY OF TEST PLATFORM

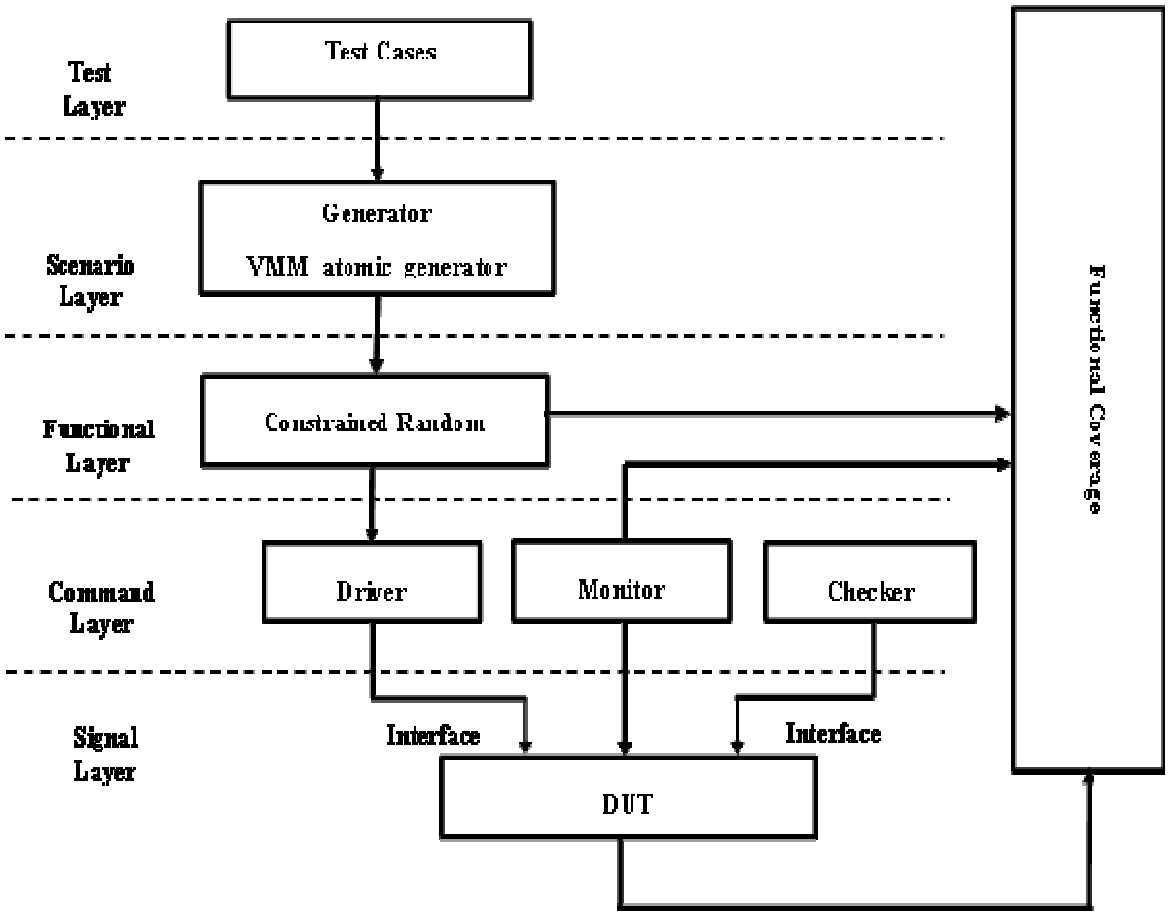

Figure 2: hierarchy of test platform

In order to reflect reusability, we place the test cases on top of verification environment and implement the abstraction and automation functions in every layer that help minimize the number and details of test cases that need to be written as VMM recommends. Another benefit of this test bench organization is easy for verification engineers to assemble the previous developed test vectors and test environments into the new verification platform. From top to bottom, the whole platform has five layers: test layer, scenario layer, functional layer, command layer, signal layer. We put some layers which have close relationship together to form several small structures. This implementation is easy and reusable.
Test layer is implemented on top of the verification environment, determining the configuration of the verification environment.We start to define the test suites according to the verification plan. For the highest abstraction of test platform, they can be easily implemented from verification plan without tough translating and dividing. The scenario layer provides synchronous and controllable stimulus sequences with the DUT (Design underTest) configuration information. The latter will be translated into necessary register reads and writes executions under the test case requirements. The stimulus is packed into a transaction object and transferred to the lower layer through channels. VMM class

"vmm_atomic_gen" can be easily invoked to generate random stimulus [5]. The functional layer processes the application-level transactions. It is the middle layer of our platform. This layer supplies the basic service to the upper layer and sends constrained-random generated parameters to the command layer. Command layer is one of the key layers of the verification platform. It implements the atomic functions of each module though signals assignment and combination to the upper layer, which have been defined and bound on the lower layer. It contains physical-level drivers, monitors, and checkers associated with various interfaces and physical-level protocols present in the DUT. The driver serves as a master to supply stimulus data to the DUT. The monitor reports the observed high level transaction timing and the data information. The checker is implemented by system Verilog assertion.The function of signal layer is connecting the signals of DUT and platform.

V. USE OF VERIFICATION IP

Verification IP models are inherently reusableblocks, simplifying the integration of multiple componentsand accelerating the schedule of verification platform building. As shown in Figure 3, the components in the big block are thereconfigured Verification IP components. The master serves as a driver and the DUT plays the slave role. Monitor IP isused to observe the behavior of the Verification IP master and the response of the DUT. Master IP and monitor IP have bothtransaction and physical level interface, which can adapt thetransaction transportation from upper layer and the connectionto the DUT ofRTL level.

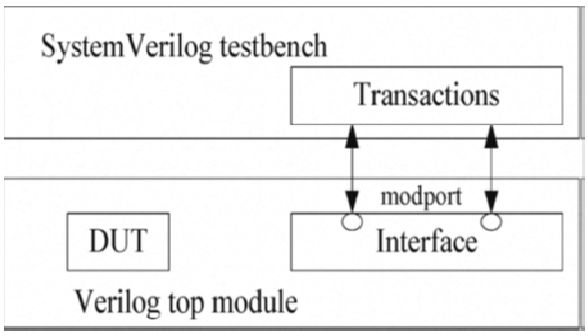

Figure 3: verification platform

VI. EXPERIMENT AND RESULTS

CIM is a typical Verilog-top multi-layer HDL design, so Synopsys VCS tool, is used to implement the verification. Design of the DUT is written in Verilog and verification is done using System Verilog using VMM methodology.

Pixel data is output on a 20-bit digital video bus and is synchronized to the positive edge of mclk. Data is transmitted when hsync and vsync are both high. A vsync pulse separates the two frames/fields and indicates the start of the next frame/field. A hsync pulse separates each line of video and indicates the beginning of a new scan line. Blanking intervals contain vertical sync (vsync) and horizontal sync (hsync) pulses.Blanking occurs when either hsync or vsync or both go low. This can be observed in figure 5.

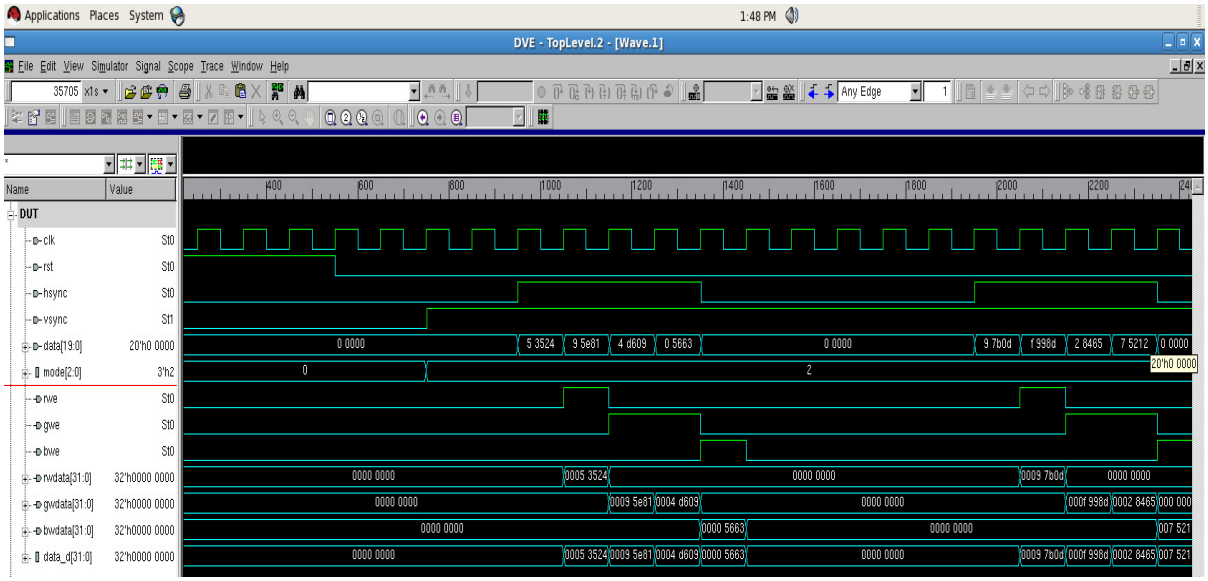

Figure 4: simulation waveform of design

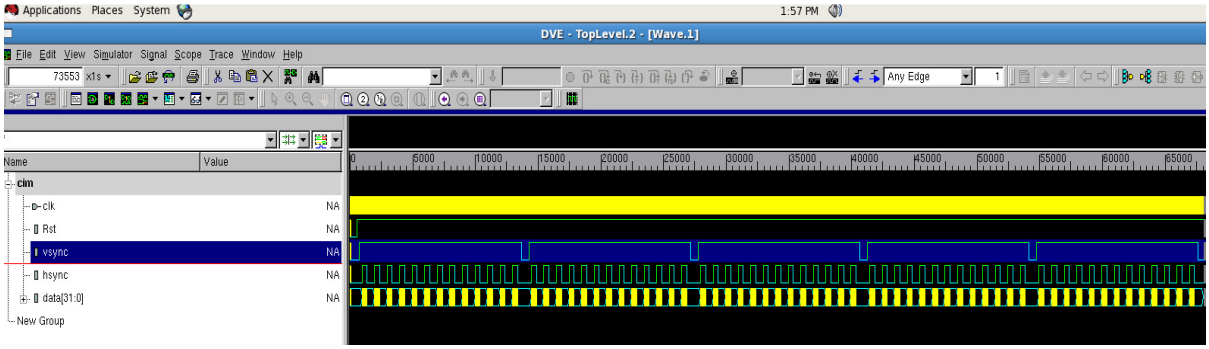

Figure 5: the simulation waveform of CIM (showing multiple hsync and vsync)

VII. CONCLUSION

Verification is most important in the design of system-on-chip (SoC) devices and reusable IP blocks.A VMM-based verification platform has been implemented and applied to camera interface module (CIM) SoC in this paper. The whole verification environment is implemented using system Verilog based on VMM methodology and the simulation tool used is Synopsys VCS. All processes of this verification areexecuted automatically by the verification environment. New constraints and tests are used to explore new areas according to the coverage results. If any bugs are found during the verification process, then the errors can be easilylocated according to the report. Moreover, if the environment created, is compiled once, it can berun more times and hence it is very easy to explore newareas just by changing seeds, without being compiled again. The verification platform used possesses fine configurability, flexibility, and high performance. The proposed methodology can also bereused in the similar verifications of different designs.

VIII. FUTURE WORK

Future work can include the combination of constrained random and directed test case methodology to further improve the speed of verification.

REFERENCES

[1] Martin Keaveney, Anthony McMahon, Niall O'Keeffe, Kevin Keanet, James O'Reilly,(2008) "The development of advanced verification environments using system verilog", ISSC 2008, Galway, June, 18-19.

[2] Yifan Wang, Stefan Joeres, RalfWunderlich, and Stefan Heinen, Fellow, IEEE, "Modeling Approaches for functional verification of RF-SoCs: Limits and Future Requirements",*IEEE Transactions on ComputerAidedDesign ofIntegrated Circuits and Systems,* vol. 28, no. 5, pp. 769773, May. 2009.
[3] IEEE Standard for System Verilog, IEEE Standard 1800tm- 2500.
[4]Janick Bergeron, Eduard Cerny, Alan Hunter, and Andrew Nigntingale,(2005) *Verification methodology manual for System Verilog,* Springer press.
[5] Chris Spears,*System Verilog for Verification*, (2008) Springer press, ISBN 978-0-387-76529-7
[6] Nathan Kitchen, Andreas Kuehlmann,(2007)"Stimulus generation for constrained random simulation", IEEE.
[7] Hu-HsiYeh, Chung-Yang (Ric) Huang,(2010) "Automatic constraint generation for guided random simulation", IEEE.
[8] Ben Cohen, SrinivasanVenkataramanan, and AjeethaKumari,(2006)*A Pragmatic Approach to VMM Adoption*, VhdlCohen press: Los Angeles.

Authors

Swathi M Mohan obtained her B.E degree in Electronics and Communication from BNM Institute of Technology, Bangalore of Visvesvaraya Technological University in the year 2010 and presently pursuing her M-Tech in VLSI Design and Embedded Systems from Bangalore Institute of Technology, Bangalore, India.

J C Narayana Swamy obtained his B.E degree in Electronics and Communication from AIT Bangalore of Bangalore University and M.E degree in Electronics from BMSCE Bangaloreof Bangalore University and presently pursuing Ph.D. in spread spectrum techniques for RFID from Visvesvaraya Technological University, Bangalore. He is currently working at Bangalore Institute of Technology as associate professor in Electronics and Communication department.