A Functional Verification Method for an All-Digital Automatic Gain Control Block

M. H. G. Oliveira[1,2], V. Martins[1,2] and W. J. Chau[2]
[1]IC Training Center of University of São Paulo (CT-SP), São Paulo, Brazil
[2]PPGEE, School of Engineering, USP, São Paulo, Brazil
e-mail: {marcio.hermany, viniciusomartins }@gmail.com
jcwang@lme.usp.br

Abstract - There are multiple verification efforts within the design flow and among these, the most time consuming is the functional verification, which is responsible for ensure that the circuit to be implemented is an accurate representation of its specification, i.e., its intended behavior. In this paper, we present a case study of a functional verification method applied to an All-Digital Automatic Gain Control Block, belonging to the digital part of the phase modulation employed in the Brazilian System Data Collection DSP transponder.

Keywords - functional verification; coverage analysis; verification strategies.

1. Introduction

The design verification is one of the most crucial activities of any system design effort. Actually, it consumes between 50% and 70% of the design effort and has become one of the key bottlenecks and challenges in the chip and system design [1].

Ensuring functional correctness is considered by several researchers, the most difficult part of designing a hardware system [2]. The impact of a non-detected functional failure may vary from a simple annoyance to a major catastrophe, in which the cost for individuals, companies, and society as a whole, can be very high.

In this paper, we present a functional verification methodology to be applied to a specific All-Digital Automatic Gain Control (AGC) Block, employed in the Brazilian System Data Collection (***S****istema* ***B****rasileiro de* ***C****oleta de* ***D****ados* - SBCD) DSP transponder. This block differs of a standard digital AGC block found in literature because it also accumulates the function of an Upsampler and Interpolation Filter. This fact increases the complexity of the functional verification and justifies the adoption of a hierarchical approach in order to reduce time and work in this task.

2. Design Specifications

The complete architecture for the SBCD transponder SoC is presented in details in [3]. The AGC block, along with the Phase Modulator block, are the responsible for implementing the digital part of the phase modulation employed in the SBCD transponder.

The objective of the AGC block in the phase modulation operation is to assure that the output power of the transmitter will not exceed the limits imposed by international regulations.

Another specification aspect to take into account is the several harmonics resulting from the non-linear operation involved in the modulation process; for this reason, the phase modulation operation should have the output frequency bandwidth greater than the input signal bandwidth. Therefore, it is needed an upsampling operation in the input signals of the AGC block.

The upsampling operation introduces undesired images in the spectrum that must be filtered out by an Interpolation Filter. Hence, the AGC block is composed by the following chain of signal processing sub-blocks: Upsampler, Interpolation Filter and Automatic Gain Control.

3. Verification Strategies and Verification Environments

The first strategy planned for the AGC block's functional verification method takes into account that the behaviors of Upsampler and Interpolation Filter are directly related. So, for verification purposes, they were considered as one single block, named UP_IF1 and UP_IF2 (for the first and second stage of upsampling and filtering, respectively). This action has been aimed to simplify the verification project by reducing the number of verification environments to be created. These two blocks are represented in Fig. 1 by dashed lines.

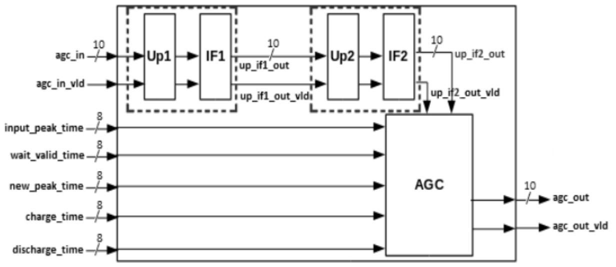


Fig.1. *Block diagram of the Register Transfer Level model (RTL) implemented for the AGC block.*

The verification follows a bottom-up approach, when firstly, a verification environment will be constructed for the sub-blocks UP_IF1, UP_IF2 and AGC, and each one are to be verified individually. The advantage of this approach is that the sub-blocks could be verified in parallel, saving time spent on these tasks. Afterwards, the verification must ascend to the top level and an environment for the top block is to be created.

A. Sub-blocks Verification Environment

A representation of the verification environment suggested for the UP_IF1 sub-block is shown in Fig. 2. It is composed by one input Bus Functional Model (BFM), one sequence generator, one reset BFM, two monitors and two coverage collectors (one for input and the other for output), one agent and one checker. The UP_IF2 and the AGC sub-blocks have similar verification environments.

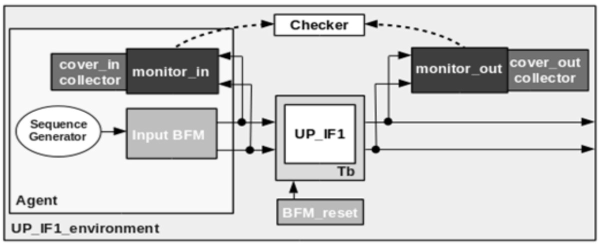


Fig.2. *UP_IF1 verification environment.*

B. Top-Level Verification Environment

The verification architecture for the AGC block is just an instantiation of the sub-blocks verification environments. In Fig. 3, it is shown a representation of the AGC block top-level verification environment with the sub-blocks verification environments instantiated.

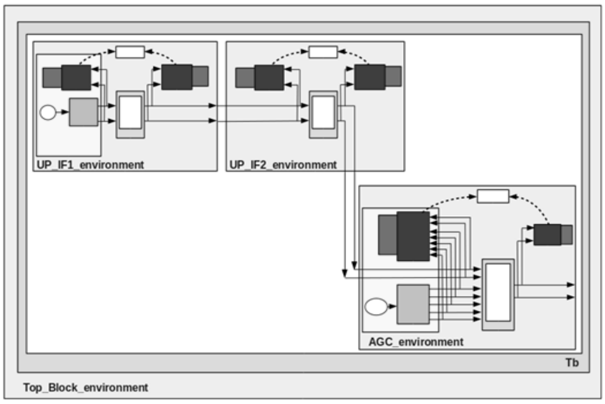
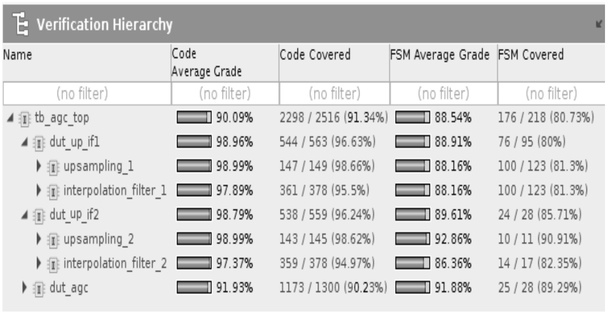


Fig.3. *Block diagram of the Register Transfer Level model implemented for the AGC block.*

4. Results Analyses

All the results presented in this paper were collected using the software Cadence Incisive Metrics Center (IMC) version 15.10-s007. In Fig. 4, the results of code coverage and for FSM (Finite State Machine) coverage for the environments are presented.

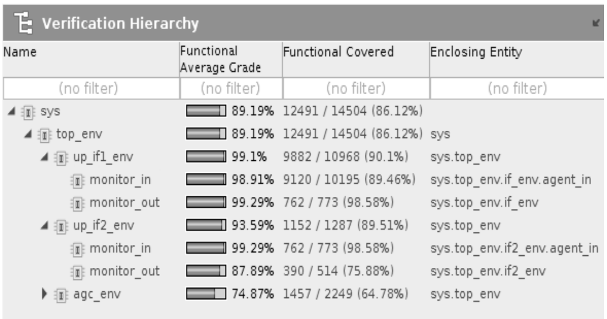


Name	Code Average Grade	Code Covered	FSM Average Grade	FSM Covered
(no filter)	(no filter)	(no filter)	(no filter)	(no filter)
tb_agc_top	90.09%	2298 / 2516 (91.34%)	88.54%	176 / 218 (80.73%)
dut_up_if1	98.96%	544 / 563 (96.63%)	88.91%	76 / 95 (80%)
upsampling_1	98.99%	147 / 149 (98.66%)	88.16%	100 / 123 (81.3%)
interpolation_filter_1	97.89%	361 / 378 (95.5%)	88.16%	100 / 123 (81.3%)
dut_up_if2	98.79%	538 / 559 (96.24%)	89.61%	24 / 28 (85.71%)
upsampling_2	98.99%	143 / 145 (98.62%)	92.86%	10 / 11 (90.91%)
interpolation_filter_2	97.37%	359 / 378 (94.97%)	86.36%	14 / 17 (82.35%)
dut_agc	91.93%	1173 / 1300 (90.23%)	91.88%	25 / 28 (89.29%)

Fig.4. *Code Coverage Results.*

It is possible to observe that high code and FSM coverage was achieved. This means that practically all lines of code, all states and all possible transaction of states in the finite state machine were exercised.

In Fig. 5, it is presented the results of functional coverage obtained. The functional coverage points are defined by considering that all possible values of all signals (see Fig. 2) must occur at least once.



Name	Functional Average Grade	Functional Covered	Enclosing Entity
(no filter)	(no filter)	(no filter)	(no filter)
sys	89.19%	12491 / 14504 (86.12%)	
top_env	89.19%	12491 / 14504 (86.12%)	sys
up_if1_env	99.1%	9882 / 10968 (90.1%)	sys.top_env
monitor_in	98.91%	9120 / 10195 (89.46%)	sys.top_env.if_env.agent_in
monitor_out	99.29%	762 / 773 (98.58%)	sys.top_env.if_env
up_if2_env	93.59%	1152 / 1287 (89.51%)	sys.top_env
monitor_in	99.29%	762 / 773 (98.58%)	sys.top_env.if2_env.agent_in
monitor_out	87.89%	390 / 514 (75.88%)	sys.top_env.if2_env
agc_env	74.87%	1457 / 2249 (64.78%)	sys.top_env

Fig.5. *Functional Coverage Results.*

The functional coverage presents expressive results, except for the input monitor of the *AGC_env*, which presents a coverage of 74.87%. That occurs because the signals are sent by the Control Block only once during the operation of the DSP, after the reset. Thus, for obtaining a high coverage of those signals, it would be necessary a very high number of resets during the simulation. That would increase considerably the duration of the verification task, extrapolating the time allocated for it and delaying all other following steps in the design flow. This low coverage of the control signals is a limitation for the functional verification method presented, therefore, alternative methodologies should be tried to get around it.

5. Conclusions

The present work shows a case study of a functional verification method applied to a specific All-Digital AGC Block, employed in the SBCD DSP transponder.

Due the strict time-to-market deadlines, strategies as merging of blocks for verification purposes and hierarchical approach have been taken in order to simplify the verification project and reduce time with simulations. One limitation of our functional verification method is the low coverage of the control parameters. This tradeoff is necessary in order not to extrapolate the time allocated for the functional verification task in the design flow.

References

[1] B. Wile, J. C. Goss, W. Roesner, “Comprehensive functional verification the complete industry cycle”, Morgan Kaufmann, 2005.

[2] S. Tasiran, K. Keutzer, “Coverage metrics for functional validation of hardware designs”, pp. 36 - 45, July/August 2001 [IEEE Design & Test of Computers (Volume: 18, Issue: 4)].

[3] M. Negreiros et al., “System-level analysis for a new sbcd transponder soc” WCAS 2015 - Workshop on Circuits and System Design August/September, 2015 - Salvador, Brazil.