An Inventive Procedure for Composite Organization on Chip Confirmation

Chang chun, Daquan da
ME Students, Department of Electronics and Communication Engineering
Shandong Jianzhu University, China

Abstract

Verification is the greatest considerable task in achieving advanced SOC strategies in market. The important competition to be described in the Semiconductor manufacturing is the developing difficulty of SOCs. Manufacturing specialists anticipate that the verification power is almost 70% to 75% of the overall purpose asset. Verification semantic cannot unassisted growth confirmation effectiveness but it must be supplemented by a system to permit repossess to the dangerous level under dissimilar approach IP structures. The expansion in the hardware field made it conceivable the grouping of a widespread yet composite system on a one chip. A task facing the SoC inventors is to agree which system level language we have to practice and how the confirmation mission will be realized. This Innovative recyclable test bench improvement will reduce the time to market for a chip. It will comfort in code reclaim so that the same code used in sub-block level can be used in block level and top level as well that helps in saving cost for a tape-out of a chip.

I. INTRODUCTION

While considering previous years, the EDA industry went increasingly from entry level to register-transfer level intellection. This is one of the elementary causes why this method increased a great rise in the efficiency. Currently, an essential strength is being spent in command to improve a system level language and to describe new strategy and confirmation approaches and substantiation at this intellection level. The complication of the chip has improved in present years and incorporation of more numbers of constituents in a single Soc makes verification of any Soc design very dangerous.

Thus we want proper confirmation methodology for any Soc or IP. Since the object oriented programming concepts in authentication make it tranquil. In this paper, the difficulties concerning code reusability, quicker time to market, and suppleness are determined by emerging the test bench environment by an advanced Verification reusable methodology. Less energy intake, reusability, better performance, lesser replication time were the targets achieved by using this advanced methodology.

By observing at the disclaimers of systems, mostly for communications, movable and multimedia apparatus, we can understand an essential and prompt growth in complication constructing it requirement System-on-a-Chip (SOC) solutions that usually take part diverse hardware and software. Time-to-market and cost are required to be condensed more than ever before and assisted up by an effective marketing-driven approach that can encounter today's highly economical and challenging conditions. To attain all this, the product growth process must guarantee the product requirement stage is combined easily with the product design part, permitting the customer's demands, promoting aims and designer knowledge, to be assessed and examined at suggestively less cost in time and possessions, and to be promptly combined into the final product.

Motivation is the most principal method used in useful confirmation today and delivers ability to confirm the application before a method is contrived which protects improvement time and struggle to a huge level. To pretend the DUT under a range of test circumstances comprising correct and damaged test contributions. Productivity, elasticity and reclaim are the aims in emerging the test bench. Accomplishing these objectives often makes test forms more problematic to use and more intricate to generate. Each test bench designer should create a trade-off between the time and determination to make and use the test bench versus the possible gain from constructing the test bench recyclable, effective and stretchy.

The confirmation of SoC is a more thoughtful tailback in the design cycle. In detail, describing SoC design language and approach is a material of time, though; the verification is a very undefended and indefinite question. Practical verification is overriding an excessive amount of the design cycle time. Assessments vary, but most predictors and engineers agree that of the design cycle is expended by efficient verification. In accumulation, the excellence of these confirmation determinations has become more significant than ever because the newest silicon processes are now complemented by higher re-spin costs.

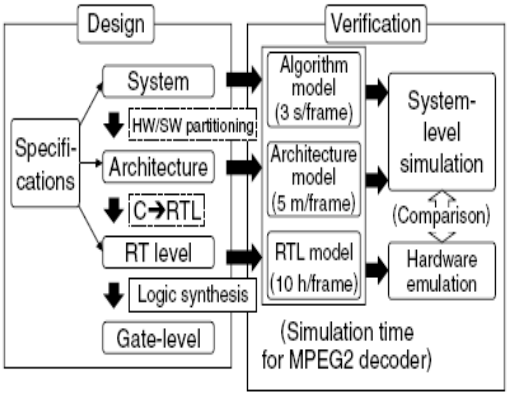

Fig 1 SoC design flow

II. VERIFICATION METHOD

The verification approach contemplates a proper structure to accomplish coverage driven verification. This system associates self-checking test benches, instinctive test generation and exposure metrics to noticeably reducing the time over to verify a design. The determination of this method is to guarantee that thorough verification is done by means of up -front goal location and eradicate the struggle and time spent for generating hundreds of tests. It also maintenances in getting initial notices of mistakes and systematizes error consideration to abbreviate reestablishing and runtime assessment. The customary attentive testing flow is dissimilar than the present flow. The purposes are recognized in CDV by using a prepared scheduling procedure. Then a test bench that produces and centrals legal inducements to the DUT is formed.

Revelation monitors are encompassed to the situation for evaluating progress and recognize non-exercised functionality. For documentation of undesired behavior, assessors are added. Simulations are agreed after both the test bench and analysis model are executed. Using this phenomenon, detailed confirmation of the design is attained by altering the randomization seed or test bench limitations. Test limitations are comprised on top of the substructure so that the verification objectives can be attained hurriedly. These significant concepts help in code reusability and decrease the time to market for a chip. Figure 2 shows the test bench architecture for this radical verification methodology.

A. Transaction centered test bench

Traditional interactive test benches can develop awkward and ungainly as the complication of strategies grows. Through system level verification, transaction centered test benches deliver a greater level of concept by producing
Thorough communication envelopes or entire read/write cycles complete bus architectures. Commonly these test benches can be measured via steering files written in pseudo code or some form of developed level language thus allowing the designer to target composite functional areas.

B. Verification Hastening

Verification hastening systems, such as the Palladium from Cadence, may be used for direction debug and analysis, simulation hastening or for competition. Palladium delivers access to the DUT's I/O to the real world. Speed Bridge hardware, which defenses data transfers among the DUT and the real word, types it conceivable to test software or firmware in a real system before going to silicon or even prior to use on an FPGA.

C. Procedure exploration

There are a number of dissimilar procedure analyzers obtainable for an extensive variety of values, and with changing features. Analyzers may be used as inactive monitors being plugged in-circuit between an initiator and a target, or alternatively they may be used to initiate traffic as an exerciser or to accept traffic as an endpoint. We have originate the passive and exerciser modes to be the greatest useful since our strategy efforts have continuously been targeted concerning endpoints. In passive mode the analyzer will achieve in the similar way as confirmation IP associated as a display in a test bench. The analyzer retains track of the traffic over the interface being tested and reports on any procedure desecrations.

D. Transaction-based Verification

Transaction-based verification permits simulation and correct of the enterprise at the transaction level, in calculation to the signal/pin level. All believable transaction categories among blocks in a system are formed and methodically tested. Transaction based confirmation does not necessitate thorough test benches with great vectors. The bus function model is used in transaction-based verification. BFMs deliver a resource of consecutively the transactions on the hardware enterprise interfaces. They determination signals on the interconnects conferring to the necessities of the interface protocols.

E. Hardware Hastening

Hardware hastening maps around or all of the devices in software regeneration into a hardware stage accurately planned to speed up guaranteed

simulation procedures. Most commonly, the test bench residues running in software, though the certain design being established is run in the hardware hastening. Around of the selections distribute hastening capability level for test bench.

III. PROPOSED SYSTEM

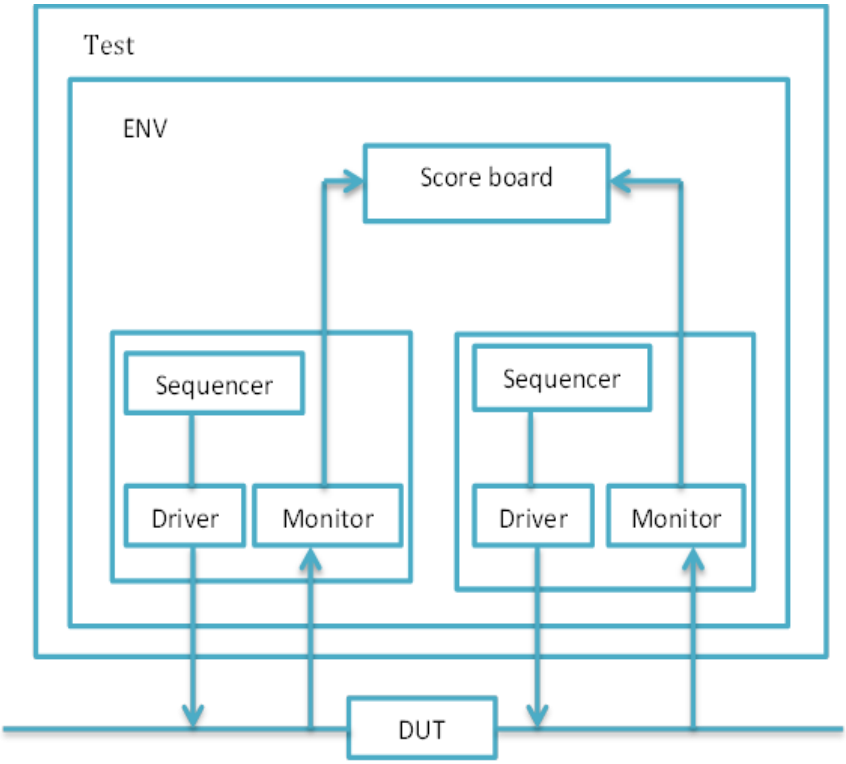

Fig 2 Verification methodology Test Bench setup

The blocks of the exceeding setup is defined below:

A. Data

The answer to the Device under Test is Data materials encompass directives and bus infrastructures. The data article's requirement originates features and arenas of a data article. Typically, numerous data items are created and concentrating to the Design below test by energetically randomizing data article fields using System Verilog limitations which properties in more amount of tests and assistances in exploiting reporting.

B. Driver

A driver is a dynamic thing which rivals logic that initiates DUT. The data items are recurrently expected by the driver and models them drives it to the DUT.

C. Sequencer

An innovative motivation generator is sequencer that switches the items which are delivered to driver for implementation. In that situation, a sequencer performs parallel to a simple stimulus generator and also yields a random data item on appeal from driver. The evasion behavior of driver permits comprising restraints in data item class for monitoring the distribution of randomized values. Generators are used to randomize groups of transactions while sequencer is used to capture significant randomization necessities.

D. Monitor

Monitor is quiet thing which a prototype signals without inspiring them. It accumulates exposure information along with comprehensive checking. A monitor is used to collect dealings and selections signal information from bus. The next explains the information to an operation which can be made offered for other constituents and to the test writer as well.

E. Agent

Sequencers, drivers and monitors can be used separately. For decreasing the extent of work and information as per the necessity of test writer, this approach mentions the design of a more abstract model known as agent. Agents can confirm DUT devices. Some agents also pledge connections to the DUT for example master or communicate agents,

while other agents react to transaction requirements which are known as slave or obtain agents. Agents should be configurable to be as either active or passive. Connections are determined affording to test directives by Active agents. DUT activity is monitored by passive agents.

F. Environment

The top-level constituent of the Confirmation Constituent is the background. It can comprise one or more agents, along with a bus monitor. The atmosphere comprises arrangement. An asset which permits in modifying the topology and performance make it recyclable. For illustration, active agents can be improved to passive agents when verification setting is recycled in system verification units.

IV. PERFORMANCE ANALYSIS

By using verification simulation software, the Authentication of communication based SOC have been agreed and the log files for the test circumstances are produced with handling report. So the entire design is carried out using HDL and the verification is carried out by using unconventional verification Methodology. The communication based SOC has been set as DUT for the efficient authentication and 95% code exposure has been acquired by using verification simulation software.

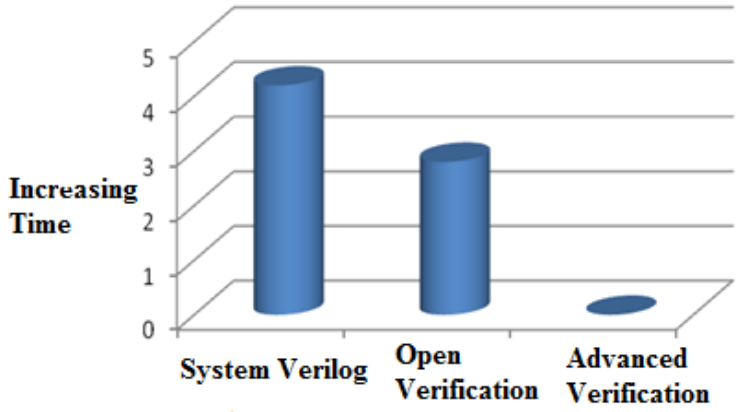

Fig 3 Comparison graph for Different Simulation time

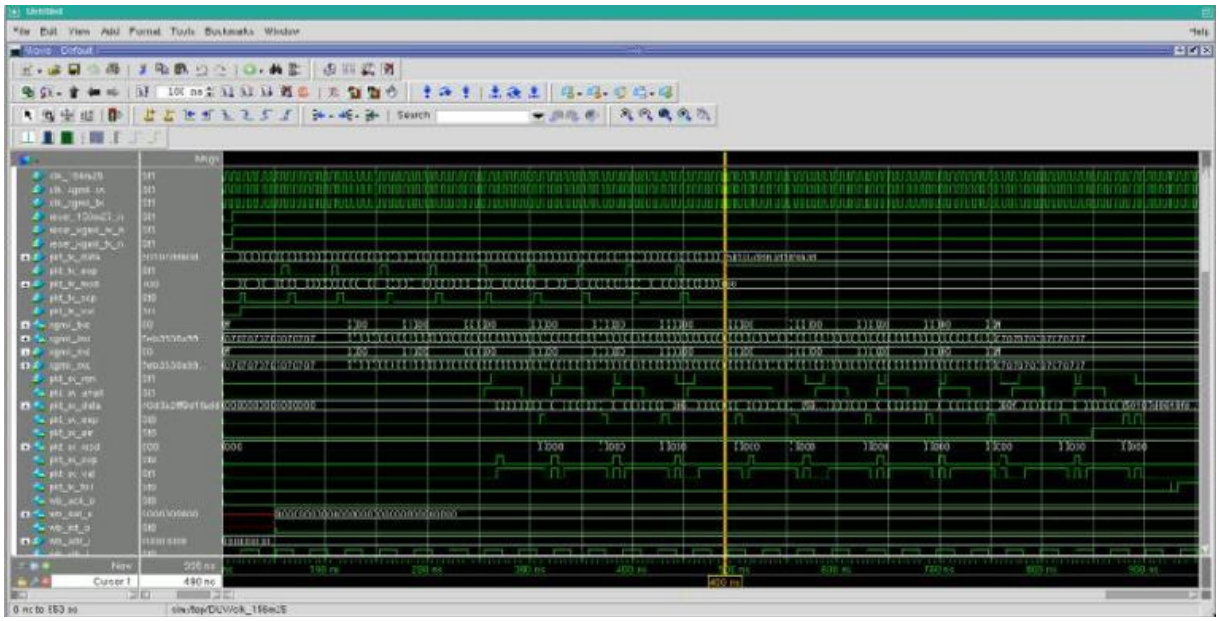

Fig 4 Simulation Result

Fig.3. shows the evaluation between different verification approaches i.e. System Verilog, Open authentication approach and innovative verification system. It is perfect from the figure that

progressive verification methodology proceeds the minimum time for assessment to system Verilog and OVM. Progressive verification approach is more time capable for accomplishment coverage objective compared to other methods.

V. CONCLUSION

SoC design and verification is an unprotected project introduced an argument among hardware and software systems. The definite issues for embedded chip approach are progressively embedded software concerns. Three central improvements are planned to design of SoC. The software based approaches see that founding everything from the HDL method is beginning from the erroneous point. The provisions of Communication based SOC are confirmed efficaciously using Advance verification methodology on verification simulator and code coverage has been taken out. For development of code coverage alteration in the code has been done as stated by the requirement. Protocol design of communication based SOC and that can also be executed in real time systems. The authentication flow in this exploration has not only reduced possessions and determinations of SOC team to collect knowledge, progress test bench, test cases and restoring but also diminished the IP team's exertions as well. There are also limitations with nowadays system in the face of the emerging use of SoCs. Specific designs that were once a complete structure are blocks in a SoC. The related verification approach with many of these blocks was not intended to scale up to a higher combination level.

REFERENCES

[1] P. Alexander and D. Barton, "A Tutorial Introduction to Rosetta," Hardware Description Languages Conference (HDLCon'01), San Jose, CA., March 2001.

[2] A Low-Cost and High-Performance Embedded System Architecture and an Evaluation Methodology'', 2014 IEEE Computer Society Annual Symposium on VLSI.

[3] P. Alexander, R. Kamath and D. Barton, "System Specification in Rosetta", presented at the IEEE Engineering of Computer Based Systems Symposium, Nashville, Edinburgh, UK, April 2000.

[4] Early development of UVM Based verification environment of image signal processing design using TLM reference model of RTL" an international journal of advanced computer science and Applications vol. 5,No2,2014.

[5] R. Armoni, L. Fix, A. Flaisher, R. Gerth, B. Ginsburg, T. Kanza, A. Landver, S. Mador-Haim, E. Singer man, A. Tiemeyer, M.Y. Vardi, and Y. Zbar, "The For Spec temporal logic: A new temporal property-specification language", Tools and Algorithms for the Construction and Analysis of Systems (TACAS'02), LNCS, Springer-Verlag, 2002.

[6] P. Bellows and B. Hutchings, "JHDL - An HDL for Reconfigurable Systems", IEEE Symposium on FPGAs for Custom Computing Machines, IEEE Computer Society Pres",Los Alamitos, CA, pp. 175-184, 1998.

[7] R. Domer, D. D. Gajski, A. Gerstlauer, S. Zhao and J. Zhu, "SpecC: Specification Language and Methodology", Kluwer Academic Publishers, 2000.

[8] P. L. Flake and Simon J. Davidmann, "Super Log, a unified design language for system-on-chip", Asia and South Pacific Design Automation Conference (ASP-DAC), pp. 583 -586, 2000.

[9] A. Ghosh, J. Kunkel and S. Liao, "Hardware Synthesis from C/C++", Design, Automation and Test in Europe, pp.382-383, March 99.

[10] A.A. Jerraya, M. Romdhani, Ph. Le Marrec, F. Hessel, P. Coste, C. Valderrama, G. F. Marchioro, J.M. Daveau,and N.-E. Zergainoh, "Multi-language Specification for System Design and Code sign", System Level Synthesis, Kluwer Academic Publishers, 1999.

[11] M. Mohtashemi and Azita Mofidian, "Verification Intellectual Property (IP) Modeling Architecture", Guide to Structured Development Using OpenVera, 2002.

[12] S. Swan, "An Introduction to System Level Modeling in SystemC 2.0. Open System C Initiative (OSCI)", Cadence Design Systems, Inc. May 2001.

[13] J. Yuan, K. Schultz, C. Pixley, H. Miller, and A. Aziz, "Automatic Vector Generation Using Constraints and Biasing", Journal of Electronic Testing: Theory and Applications, pp. 107-120, 2000.

[14] R. Otten, P. Stravers, "Challenges in Physical Chip Design", International Conference on Computer Aided Design (ICCAD), pp. 84-91, 2000.

[15] D. Moundanos, J. A. Abraham, and Y. V. Hoskote "Abstraction techniques for validation coverage analysis and test generation", IEEE Trans. Computers, vol. 47, no. 1, pp. 2-14, January 1998.