*i*Journals: International Journal of Software & Hardware Research in Engineering

ISSN-2347-9698

Volume 8 Issue 4 April 2020

A REVIEW OF CHECKER AND CHECKER TOOLS USED IN REGISTER TRANSFER LOGIC

Author: Rakesh K R[1]; Krishna C V[2]

Affiliation: [1]Assistant Professor, Department of Telecommunication Engineering, R.V College of Engineering, Bengaluru, India [1]; [2]Department of Telecommunication Engineering, R.V College of Engineering, Bengaluru, India [2]

E-mail: rakesh2989@gmail.com[1]; krishnacv23@gmail.com[2]

ABSTRACT

With increase in the complexity of the designs developed in the recent technology containing lakhs of gates, and many blocks and many intermediate releases, the checkers become very necessary and these checkers should be able to provide accurate results and in a very fast manner. These checkers are so developed that they check all types of errors starting from the vendor to the overall test coverage of the design. Hence these must be developed with utmost care and, the time consumed for generating the reports from the initializing all the inputs must very less as this will be done in every milestone release of the design. In this paper the various checkers which are used from the SoC side and IP side are discussed and the tool for generating the reports and their benefits are also added in this paper.

Keywords: RTL, checkers, SoC, IP.

1. INTRODUCTION

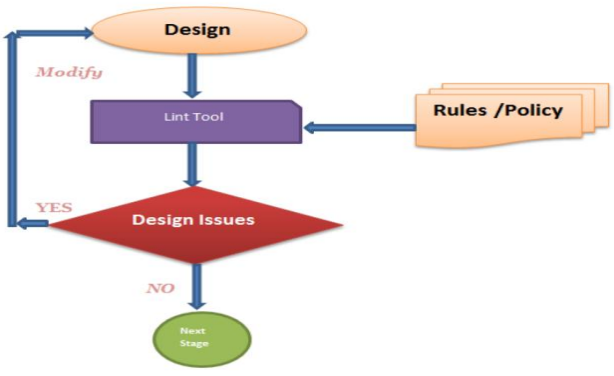
The complete access of the database design can be accessed register transfer logic (RTL) design checker. This is done by going deep into the explaining of the design and with the help graphical user interface(GUI) debugging. With the use of any high level programing language that can be run dynamically the checkers will provide a lot of advantages namely:

1.Will help in examining the design in depth

2.To trace the simulation from each point to another

3.With the development of dashboard or GUI the readability and understandability increases

The next step as shown in Fig.1 would be to check the lint using the linting tools as shown in the figure, there will be set of rules and procedures to be followed by the lint tools to check the working of the circuit. RTL linting is a process which is performed before the simulation is run so as to compensate any functional issues.



. **Fig 1: Lint check methodology**

The synthesis log check is performed only on the files which are needed to be synthesized, these files which are updated newly can be formed on to the make-file and running the make-file reduces the compilation time and complies only the files needed to be synthesized.

The dashboard gives the most critical information which are performed after the checks are done. There can be use of a graphical user interface to make it in more informative manner.

2. LITREATURE REVIEW

The RTL top level can be a combination of many functional blocks. To check each of these functional blocks there needs to be the presence of checkers developed which gives the specification about the area, speed and power constraints. The functional verification can be run parallel with the RTL design phase. Those logics which consists of a lakhs of gates the design verification can be time consuming and the epochs may be very large. [1] In order to have multiple core SoC these needs to be verification of many design parameters in RTL. Embedded SoC verification by the method of hybrid verification proposes to address the challenges of the multi-core system on chip verification was developed [4]. The cost for verification would be a lot higher if there were no software which are used for the verification of the design. The verification is present in every step of a design flow [5]. The issues which are critical are caused by the intentional latches generated and there is a proposition of automatic linting tools to have a check on the errors and help in correcting them after the evaluation. With the development of automatic checkers there will no need to go through the codes and handle them manually. Python programing language can be used to make the checkers run fast in a parallel processor [8].

3. VARIOUS CHECKERS USED IN RTL DESIGN IMPLEMENTATION

Some of the various checkers used for RTL design are namely Lint, IP-XACT, clock domain crossing test(CDC), design for testability(DFT), X-propagation(XPROP) and synthesis check. There can be more checks or less based on the design milestone or on which version the design is to be checked.

The inefficiencies that are present during the initial design can be more affecting ones in the latter stage because of which silicon respells may be created. Lint tools are therefore used to debug the detected errors and warnings of the RTL design in every stage. Lint tools are always aimed to provide the place where more consistency and correctness of the design, which are prior importance for the chip integration teams. Integration of the IP's and how effectively the reuse of the existing designs and using the existing designs should not affect the aimed design features is also what a correct and consistent design parameters are which are mostly checked by the lint tools.

IP-XACT checks are used to create the automated configuration by using the integration tools. This checker checks on the various architectures and all the IP libraries management are checked upon. IP-XACT checks for the different formats of IP and does it support for the different kind of IP descriptors or not. The port consistency and parameter consistency are also part of the checks done by IP-XACT. Some of the tools also support the information about the clock interfaces.

Clock domain crossing (CDC) is the check which is very difficult to perform and debug the errors. With the increase in complexity of the design, there can be any number of asynchronous clocks. Static timing analysis does not take into consideration of the asynchronous clocks. If there is any clock domain problem, then it involves a huge delay to debug and this cannot be pocket friendly also.CDC checks for the metastable signals if produced and major concern will be if that signal is driving any other circuitry. The conditions where asynchronous resets are used is looked into where if the flops are not reset before the arrival of next clock signal. The setup and hold time violations, race around conditions are the things which are looked by the CDC checks.

DFT performs testability of the RTL designs and analyse them to improve the test coverage goals as set by the consumer of the design.DFT is performed after the design is over.DFT checks and gives the information about getting errors in ports and voltage levels. Fault in coverage and test in coverage are the two parameter which tells as to how the design can be developed so that it can achieve higher testability.

X-propagation analyses the issues related to generation of unwanted X's.Most of the time the semantics are tool dependent. It should be ensured so that even

though the X propagates it should not affect the circuitry function and final results of the automated test pattern generation should be equivalent to what the design was designed for. The x-propagation tells which part of the circuitry is affected most and these errors must be reviewed and then moved to the next stage. Synthesis checks are majorly done on the simulation traces. These are the basic checks that must be done to ensure all proper working conditions. All the behavioural patterns should be checked before moving to next stage. Some checkers assign assertions as high quality or medium also.

4. CHECKER TOOLS AND ITS BENEFITS

Linting checks can be done using spyglass lint tool, which provides deep analysis of the RTL design in every phase. Analysis, debugging and fixing the electrical and different types of structural issues are the features of spyglass tool for every RTL design description. Spyglass lint tool supports variety of language design namely Verilog, VHDL, System Verily and V2K.This tool provides very fast performance using the SoC flow with very less noise. This tool will also help in identifying the issues related to design in RTL both in static and dynamic conditions. This tools will also check for the design reuse compilation to make sure that there is consistent style that can be used for the design.

CDC checks can be done using spyglass CDC or meridian CDC tool. Spyglass CDC tool is a protocol independent tool and has the ability to auto detect the minimum count of false vacillations which are the results of quasi static signals. Spyglass tool is comparable for both CDC and reset domain crossing (RDC) checks along with centric debugging abilities.

This tool also supports methodology based on IP which is achieved through a hierarchical SoC flow, using these capabilities the checks can be done in a very fast pace and for large size SoCs, the check time will be drastically reduced. The ease of use and easy learning of the tool is the added advantage. Meridian CDC has very good capacity and can provide results with great accuracy. Meridian CDC ensures the asynchronous clocks are received with good reliability by performing both structural and functional analysis. The use of static algorithm and in depth analysis of clock crossings design structure will enable meridian CDC tool to provide results with very low noise.

For the DFT checkers there is a tool from the synopsis namely Spyglass DFT, this tool is used to check the total fault in coverage and test in coverage. This tool reduces the implementation time and comparatively reduces the cost of the design. The quality is made good by checking for the issues in DFT in early stages of the design and then combine to increase the overall coverage of the final design. Cross probing debug environment is the main added advantage this tool provides the users. Based on the factors like observability and controllability the tool can provide the results of transition delay coverage also. The pattern less coverage will reduce the runtime.

Magillem IP-XACT packager can be used as tool for IP-XACT checker which provides fully automated benefit which can collect all the IP libraries from a large scale range and creates an description of IP - XACT .The key benefit of this tool is that it's fully automatic and reduces the time required for performing the checks to be very minimal .The integration of the IP with other clients is also facility this tool provides ,thereby having the same directory structure as it is present in the client side ,which makes it more user friendly. There is a guarantee of validity of IEEE 1685 standard generated files with correct semantics and grammar used. There are some additional benefits like identifying the strategy for importing those libraries and handle the specifications as required by the customer. In addition to all these magillem also supports incremental or full packaging.

The design compiler of RTL synthesis has launched its new product Design Compiler NXT. This new technology supports synthesis checks which is very fast and highly reliable tool, this tool has an additional feature of cloud enabling of the synthesis distribution. This tool has better quality in terms of dynamic power measurements and timing power measurements compared to other previous releases. This tool has a common physical library and runs with twice the runtime speed. With the use of cloud distribution there is good method of partitioning of workload. Synthesis of 5nm and below technology are compatible using this tool.

Jasper Gold X - propagation verification tool can be

used for the x-prop checks. This tool provides X-propagation faults which are propagated rapidly inside the design and its analysis. This app automatically first creates these checks and executes for them for all the conditions that might lead to some unwanted Xs.The major benefit is that this tool points out the reason for generation of X,and parallel speeds up the process of debbuging.This tool allows to change the input conditions and check the output on the fly, so that we can modify the RTL.Finally this tool also has the ability to provide higher throughputs and requires no prior experience or education about the system Verilog.

5. FUTURE SCOPE

All these checkers are needed to be present in every stage. So the efficiency of these checkers and reports generated by these needs to be highly efficient and they need to be automated by using some scripts so that they might be running all the checks simultaneously and this will reduce the time required. By automating these the second level what the front end developer would like to see can be the dashboard developed so for every release in different stages the user can be able to see what checkers are passed or failed. In future there is a lot of demand for low power and high frequency operating SoC, so is there any capabilities of the checkers so designed to check these types of checkers is also the future question to developed upon. There should checkers developed for mixed signal blocks as well. There should be a development of both hardware and software co-verification so that by the end, when the chip is tapped out, there should not be any errors and the coverage must by high as per the user requirements. In order to achieve all these things perfect high level language scripts might be used to automate and perform all the checks with high degree of precision. After the tapping of the chip if that is not proper then all the investment will be a waste, so the corner condition selection must be such that it can work in all domains and all extreme physical conditions as well

6. CONCLUSION

In this paper we have discussed the use of the new technology that are used and can be used to perform the checks on the RTL design. This new technology has its ups and downs but depending on the application where its being used. In the designs which has a lot of gates then time required for the checkers to check some amount of gates will be the parameter which the engineer would look for. These checker tools also have to be bought for the license of using those, cost can also be the criteria but this is very less compared to cost for tapping out the design. As the demand for low power VLSI is increasing there needs to be methods to check these type of designs in the future. Another parameter that's changing is the supply voltage, so there has been a decrease of supply voltage to a greater extent so these checkers must be able to differentiate the voltage levels and should be able to provide accurate results for the low voltage as well.

7. REFERENCES

[1]. Taraate, Vaibbhav. (2019). RTL Design and Verification: RTL Design Using Verilog. 10.1007/978-981-10-8776-9_4.

[2]. Feng, X., Jiang, Y., Yang, X., Du, M., & Li, X. (2019). *Computer vision algorithms and hardware implementations: A survey. Integration.* doi:10.1016/j.vlsi.2019.07.005

[3]. Mathur and V. Krishnaswamy, "Design for Verification in System-level Models and RTL," *2017 44th ACM/IEEE Design Automation Conference*, San Diego, CA, 2007, pp. 193-198.

[4]. P. Ghosh, "Analysis and verification of SoC design RTL parameters," *2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, 2016, pp. 1-6. doi: 10.1109/ICPEICES.2016.7853486

[5]. M. Abrahams and J. Barkley, "RTL verification strategies," *Wescon/98. Conference Proceedings (Cat. No.98CH3626)*, Anaheim, CA, USA, 1998, pp. 130-134. doi: 10.1109/WESCON.1998.716434.

[6]. R. Hariharan, T. Ghasempouri, B. Niazmand and J. Raik, "From RTL Liveness Assertions to Cost-Effective Hardware Checkers," *2018 Conference on Design of Circuits and Integrated Systems (DCIS)*, Lyon, France, 2018, pp. 1-6.

doi: 10.1109/DCIS.2018.8681487

[7]. T. Grimm, D. Lettnin and M. Hübner, "Automatic generation of RTL connectivity checkers from SystemC TLM and IP-XACT descriptions," *2016 IEEE Nordic Circuits and Systems Conference (NORCAS)*, Copenhagen, 2016, pp. 1-6. doi: 10.1109/NORCHIP.2016.7792922.

[8]. N. S. Soliman, K. Salah and A. H. Madian, "Automatic RTL coding correction Linting tool for critical issues," *2018 30th International Conference on Microelectronics(ICM)*,Sousse,Tunisia,2018,pp.252-255. doi:10.1109/ICM.2018.8704117.

[9]. D. Hansson, "Continuous Linting with Automatic Debug," *2014 15th International Microprocessor Test and Verification Workshop*, Austin, TX, 2014, pp. 70-72. doi: 10.1109/MTV.2014.25.

[10]. Soliman N.S,Salah K, & Madian A.H(2018).Automatic RTL coding correction Linting tool for critical issues. 2018 30th International Conference on Microelectronics (ICM).doi:10.1109/ICM.2018