Scientific Journal of Impact Factor (SJIF): 4.72
e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Emerging Trends and Innovations in Electronics and Communication Engineering - ETIECE-2017

Volume 5, Special Issue 01, Jan.-2018 (UGC Approved)

A NOVEL APPROACH FOR VERIFICATION OF RISC-V PROCESSOR

Ramanpreet Virdi, Neeraj Kr. Shukla

Department of Electrical, Electronics & Communication Engineering,
The NorthCap University, Gurgaon, Haryana, India

Abstract — *The CPU is a standout amongst the most significant segments of our PCs, dependable of performing essential figuring, legitimate correlations and moving information around. These straightforward errands are the building squares of any more unpredictable operation, and make running our frameworks and projects conceivable. This paper shows the better approach to verify the RISC-V (Reduced Instruction Set Architecture) processor which can incite minimal effort as the open source Verification Language (Vlang).*

Keywords- *ISA, Risc-V, Verilator, Vlang.*

I. INTRODUCTION

The planners assert that the guideline set is the principle interface in a PC, since it lies between the hardware and software[5]. On the off chance that a decent direction set were open, accessible for use by all, it ought to drastically diminish the cost of programming by allowing much more reuse. It ought to likewise expand rivalry among equipment suppliers, who can utilize more assets for outline and less for programming support [2]. RISC-V (articulated "RISC five"), which is a free and open source ISA (Instruction Set Architecture) for SoCs (Systems-on-Chip), which fundamentally bundles the CPU and other vital segments to run a framework together[4]. Most extraordinary ISAs, for instance, the ones by Intel, IBM or ARM are restrictive and can't be used by others without licenses. On the other hand, RISC-V can be used, completed and conveyed by anyone, to no end, with the fundamental honest to goodness condition being to perceive the RISC-V makers [3].
This paper is organized as, the background of the work is presented in section II, a method is proposed to solve the cost effect and compile time problem. In section IV this paper is making conclusion about verification using Vlang.

II. Background

Here the conventional method means the way open source community always verifies the product. There are number of layers which are included in its verification[7]. Firstly, as it is open source anybody can download the database and start working on it as they want it to be. They can modify according to their own specifications and need [6]. There are some test cases which will run every time you modify design of RISC-V just to check the bugs. There is open source test project called Travis CI which run on the principle of continuous integration. Continuous integration depicts that whenever designer modifies some open source code they have to test it according to their own modifications plus the test cases which were running before the modifications. There are number of steps involved in this type of verification. ELF (Extended Log Format) Executable fetch the test cases from the program called torture which generate random C++ code. It will convert the code to hex format. DTM, front end emulator, fetch the code from ELF and send it to the DUT to check its response. It is primarily checking the response of DUT for certain C++ code and comparing it to the expected results. Below diagram is showing the conventional approach to verify RISC-V Processor.

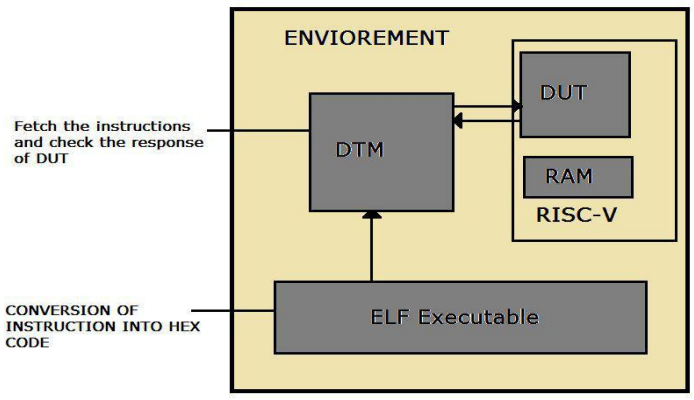

Fig 1. Block Diagram For Conventional Method

III. Proposed Method

The Proposed method includes the use of Vlang instead of vcs as it is open source. The main problem in doing modification is compile time. In conventional method the RTL, first have to change from Verilog to System C using Verilator. It takes around 2-3 minute after every edit and it will affect the productivity. In the proposed design the compilation time will be comparatively less as it is using directly Verilog RTL.

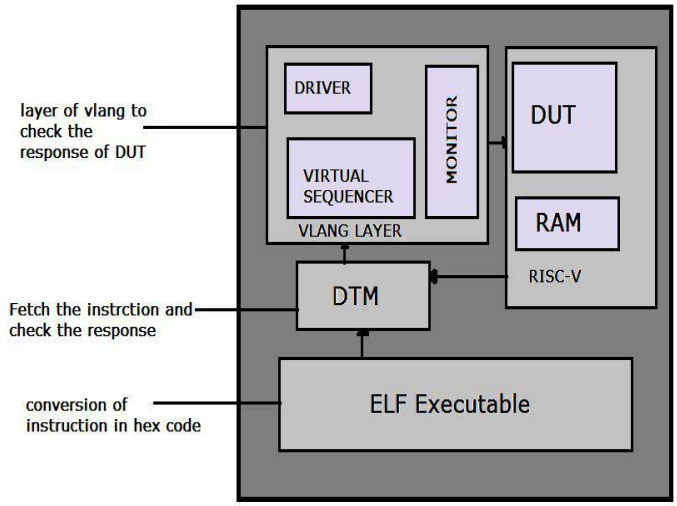

Fig 2. Block diagram of Proposed Design

IV. Conclusions

The open source processor needs an open source verification environment if it wants to be really cost effective. Multicore verification has imagined in vlang it can accelerate verification by utilizing multiple cores accessible in advanced processors the oddity lies in decrease of general plan confirmation time with the center of Multicore simulation [1]. With that the entire set up is open source including every one of the tools and software utilized as a part of setting the environment for verification. In this paper, we have proposed the method to include the DTM layer from which we use already generated test cases efficiently. The future aspect of this work is to remove that DTM layer and using our own generated test case using only Vlang.

References

[1] Puneet Goel ," Vlang A System Level Verification Perspective" DVCON India 2015. URL:https://dvconindia.org/sites/dvcon-india.org/files/archive/2015/proceedings/89_Vlang_System_Level_Verification_Perspective_paper.pdf
[2] Puneet Goel and Sumit Adhikari, "Introduction to Next Generation VerificationLanguage-Vlang",DVCON Europe 2014. URL: http://vlang.org/dvcon2014.pdf
[3] Online available at URL: https://www.xda developers.com/risc-v-cores-and-why -they-matter
[4] Online available at URL: https://www.nextplatform.com/20 16/07/11/startup-takes-risk-risc-v-cust om- silicon
[5] Online available at URL: http://www.adapteva.com/andreas-blog/why-i-will-be-using-the-risc-v-in -my-next-chip
[6] Online available at URL: https://riscv.org/specifications
[7] Rishiyur S. Nikhil, "Verification and Debugging for High-Assurance RISC-V", RISC-V International Conference at IIT Madras, April 2, 2017. URL:http://rise.cse.iitm.ac.in/ric2017/proceedings/keynote3.pdf

