Editorial

ICSES Transactions on Computer Hardware and Electrical Engineering (ITCHEE)

Journal Homepage: www.i-cses.com/itchee

Latest trends in Hardware Verification

Joydeep Bhattacharyya

Intel Corporation, San Jose, USA

Corresponding Author: joydeep.bhattacharyya@intel.com

THE semiconductor industry has seen a rapid growth in the last four decades. As people became successful in integrating more and more computing resources onto the same tiny piece of silicon, the market demand for more complex and more powerful chips grew in parallel, almost exponentially. Indeed, it's one of the wonders of the world how scientists and engineers continued on their relentless pursuit of Moore's law, doubling the transistor density on chips roughly every two years, as dictated by the famous scientist Gordon Moore, the founder of Intel Corporation [1].

Statistically speaking, the lion's share of both the time and effort in a traditional chip design project is spent on verification cycles, checking for both functional and timing correctness of the chip. Until about mid-90s of the last millennium, most of the testing used to employ a directed test-bench approach, with simulation cycles being spent to verify that the design does what it is "supposed to do". However, people soon recognized that a lot of the silicon failed just because not enough state-space explorations of the design internals were performed during the verification phase. In other words, there is almost an infinite set of possible states where the design can find itself in, based on the external stimuli provided. Hence, it was equally important to verify that (almost) none of these states take the design to a potential "hang" condition, a deadlock which can only be fixed by resetting the chip, i.e. by bringing it to a totally known state. The "almost" was used to signify that covering all these states in functional simulation is impossible, because it can take hundreds of years of simulation effort. The job of the verification engineer is thus, to prioritize among the different verification tasks, in order to maximize the chance of success for a first time silicon fabrication.

The biggest change that has happened in the semiconductor design industry in the last two decades is the introduction of self-checking constraint random test-benches [2] and [3]. In this approach, an equivalent model (or a collection of multiple models) of the design is constructed first. The design and the reference model are then exposed to a set of random stimuli which is somewhat constrained and directed in order to guide the design to explore a specific feature. The goal is more or less decided, the travelling path is not. The random tests are then run thousands of times, each time with a different random seed, and hence changing the journey of the design a little bit. The behavior of the design is compared with that of the reference model and any error is flagged and debugged. This is a very powerful technique which can yield literally hundreds or even thousands of design flaws with very minimal recurring effort.

Hardware verification has smartly adapted a lot of the ideas from software testing domain, and thus the concept of coverage collection has almost become a must for any present day chip design project. While thousands of random test-cases are running on the design to unveil any possible bug, it is necessary to know what useful purpose the stimuli is serving. Are they exercising the design well enough, or are they just confined within a small state-space, both in terms of the design features and design implementation? Latest simulation tools all support collection of code and functional coverage data while simulations are running at different times. These huge datasets are then merged and processed to identify how strong the random seeds are, and whether the verification strategies need a complete refresh.

Perhaps, introduction of no other verification methodology has created an impact more than that of assertion and formal verification, possibly because they offer a fundamentally different approach than conventional simulation model based verification [4]. Assertions are powerful tools which describe correctness in design behavior and are extremely friendly to multi-threading. Hence hundreds of thousands of assertions can run in parallel, thereby checking its behavior at all times from different aspects. Formal verification tools are based on mathematical completeness and logical equivalency checks, where certain properties of the design can be checked for all possible scenarios. These are also some of the areas where there is ample scope of introducing artificial intelligence and machine learning algorithms, thereby reducing the human effort.

As a recent example, in January 2018 the Consumer Electronics Show was held in Las Vegas, Nevada and a major attraction of the same was "Xavier", a postage stamp sized tiny chip from Nvidia Corporation [5]. This chip was designed by 2000 engineers working for four years, with a budget of 2 billion USD, has 8k video processors, an 8-core CPU and a 512-core GPU. This would serve as the brain of all "level five" autonomous cars in the upcoming future. Successful implementation of such an enormously complex chip depends more on the verification methodologies than anything else, both for quality as well as time to market. The potential arena of impact for design verification thus now is bigger than ever, and will continue being so.

With Regards,
Joydeep Bhattacharyya
February 26, 2018

References

[1] Intel official website, http://intel.com, 2018.

[2] Janick Bergeron, *Writing Testbenches: Functional Verification of HDL Models*. 2nd ed., pp. 319-325, USA:Springer, ISBN: 978-1-4615-0302-6

[3] Chris Spear and Greg Tumbush, *SystemVerilog for Verification: A Guide to Learning the Testbench Language Features*. 2nd ed., pp. 169-227, USA:Springer, ISBN: 978-1-4614-0715-7

[4] Ashok B. Mehta, *SystemVerilog Assertions and Functional Coverage: Guide to Language, Methodology and Applications*. 2nd ed., pp. 1-8, USA:Springer, ISBN: 978-3-319-30539-4

[5] Nvidia official website, http://nvidia.com, 2018.