NOVEL APPROACH TO IMPROVE THE PROCESS OF REGISTER VERIFICATION IN UVM

Sougata Bhattacharjee,

Senior Staff Engineer, Samsung Semiconductor

Email: sougata.b@samsung.com

1. Problem Statement and Motivation

➢The paper introduced two novel algorithms (March-Bash and Modified Register access) to improve the process of register verification and efficiency of the testbench.

➢The first method (March-Bash) is an improvement of bit bash algorithm and provides additional advantages in addition to what bit bash is already providing. It also nullify the effect of bit bash when the number of registers are more which will lead to increase in memory size. In short, this method makes the register and memory verification process more thorough and efficient and also useful in unearthing hidden corner case bugs. More importantly it bridges the gap between Pre – Si and Post – Si Verification and saves a lot of time.

➢The next method is Modified register access and as the name suggest is the modified version of reg access sequence in uvm. The utility of this method is to use it only for frontdoor instead of both frontdoor and backdoor, and makes it simpler to use and easy to plug in the verification environment. This method also reduces the simulation time in accessing the registers.

2. Proposed Methodology

Custom (March-Bash) Algorithm:

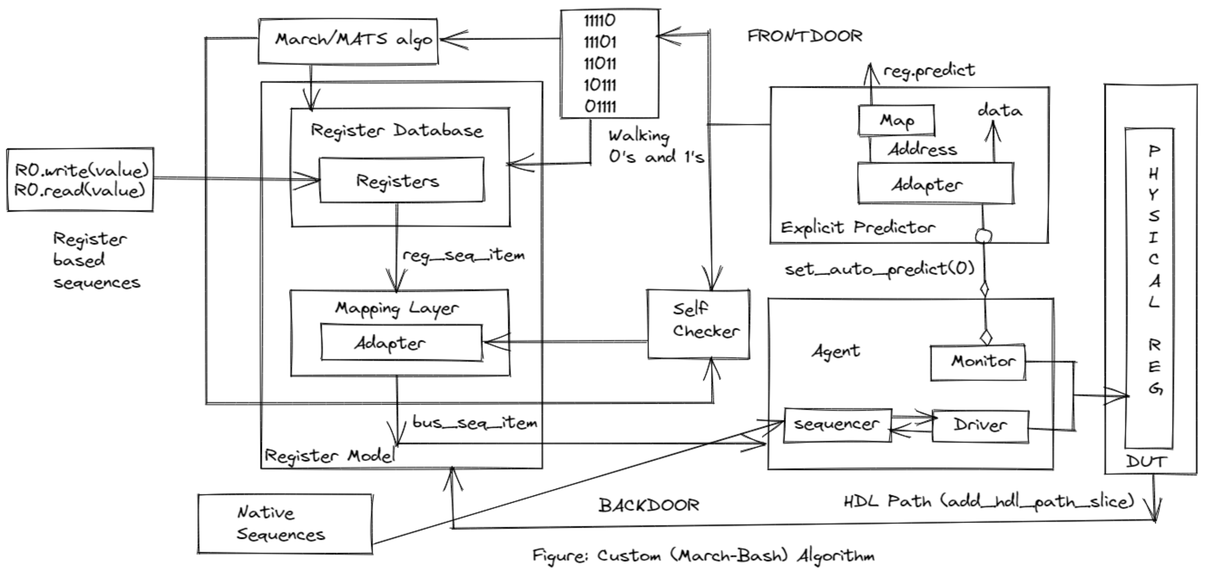


Figure: Custom (March-Bash) Algorithm

3. Execution Steps of March-Bash Algorithm

1. Writing data in the register triggered in the register database in the register model.
2. The register sequence item is converted to a bus sequence item with the help of the Adapter.
3. Map each content of the register offset with the corresponding address map.
4. Write data in the form of a reglist pattern to make sure each register within the set is covered.
5. The data is passed through the DUT interface and accesses the physical register
6. Compare the written data with the read data with the help of a predictor
7. The checked data is taken as input and passed on to the bus for accuracy.
8. Write each individual bit of the register in the form of walking 0 / 1
9. Write 0 with up addressing order within the register/memory set
10. Read 0 and write 1 with increasing addressing order and Read 1 and write 0 with decreasing addressing order.
11. Continue the same process with an increased number of register and memory set
12. The corresponding data is checked with predicted data & compared and if it is correct update the mirror value

4. Execution Steps of Modified Register Access Algorithm

1. Write data through a sequence in the register which will trigger the register database in the register model.
2. The register sequence is converted to a bus sequence item with the help of an adapter.
3. Map each content of the register offset with the register map and make sure the written data is through the agent interface.
4. The data passed through the agent interface and access the physical register of the DUT and during that point customization has been applied.
5. Read the data with frontdoor instead of backdoor which results in reduced simulation time due to reduction of two-way approach.
6. Route the data via implicit predictor and check if the data has been read properly and updated the mirrored value

5. Advantages of March-Bash and Modified Register Access algorithm

March-Bash:

➢Each bit or field of the register is verified and checked.

➢Each signal of RTL is being checked for Stuck at Fault 0 and 1

➢Cost and Time Saving in terms of the issues being addressed at Pre Silicon Verification itself.

➢Resolves the problem of Transition bugs.

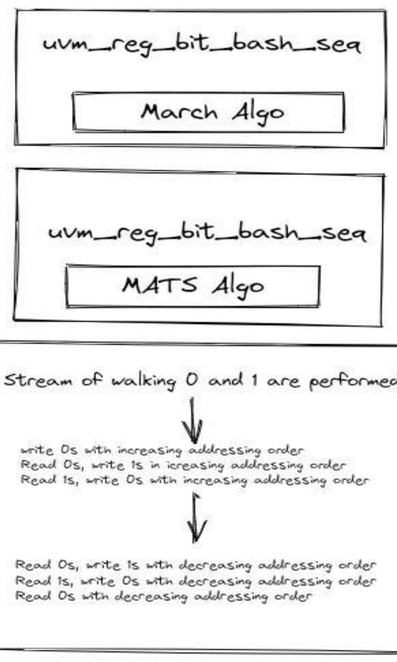
➢Resolves the Address Decoding Issues

➢Thorough checks increases the probability of identification of any corner case issues.

➢Reduces debug effort and improves the efficiency of the testbench.

➢The bug can be traced down from the log in the form of register or memory.

➢Memories can be claimed as regions in the form of clusters.



Modified Register Access:

➢Each register is accessed in the form of a register list instead of bit-by-bit approach.

➢The registers are accessed through frontdoor instead of both frontdoor and backdoor and hence saves simulation time.

➢Extended from uvm_reg_single_access_seq but the features are of modified register access sequence.

➢Minimal code and easily integrable in the verification environment and able to sweep any number of registers.

6. Summary and Results

➢The simulation time is reduced by 30% (calculated on the basis of time a test scenario is being executed) in case of modified register access sequence as compared to the existing register access sequence in uvm. The only exception in using modified register sequence is that it works only for frontdoor.

➢The source code length for developing modified register access is reduced by 50% (calculated on the basis of number of lines of code) as compared to the register sequence used in the existing testbench.

➢The efficiency of the register verification in frontdoor is increased by roughly 40% (calculated on the basis of feature an algorithm supports) by using March-Bash algorithm because it provides additional benefits of hitting more corner case bugs, finding the transition, stuck at fault and address decoding bugs.

➢The effort has been reduced by 10% (calculated on the basis of time) in the backend process of Design for Testability while checking the issues in the memory.

➢The reusability within the testbench is increased by 20% (calculated on the basis of plug-and-play feature and on the basis of change in address space) by adopting both the above-mentioned algorithms.

7. REFERENCES

1. IEEE Standard for Universal Verification Methodology 1800.2-2017 language reference manual
2. Universal Verification Methodology (UVM) 1.2 User Guide
3. Ram Testing Paper by National Central University, Taiwan