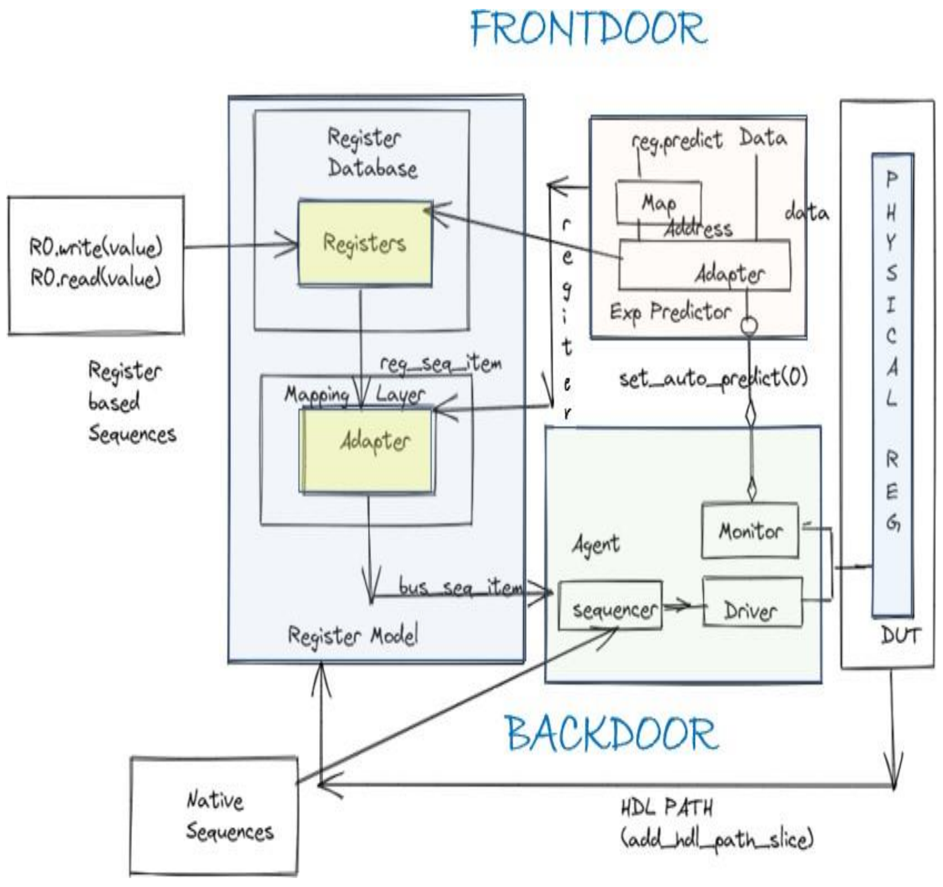
accellera
SYSTEMS INITIATIVE™

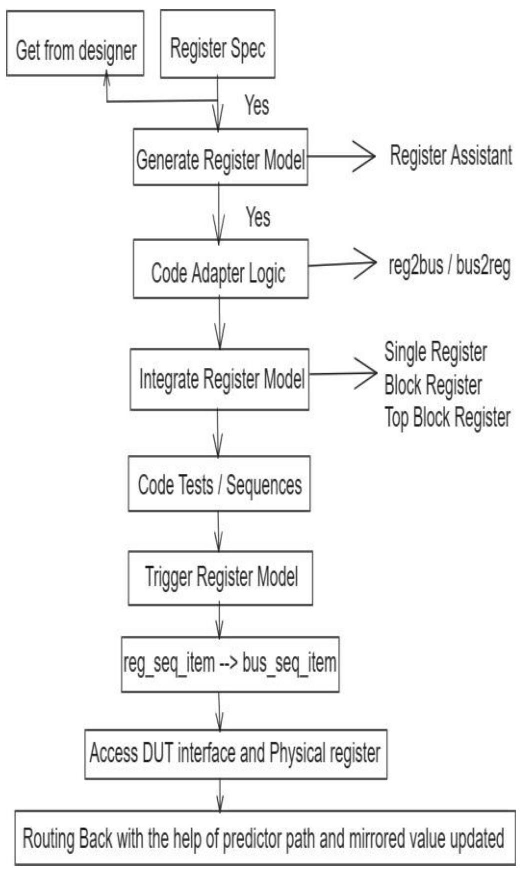
Novel Customized Algorithm and Verification Checklist to Improve the Process of Register Verification in UVM

Sougata Bhattacharjee,
Senior Staff Engineer, Samsung Semiconductor (SSIR), Bangalore, India
Email: Sougata.b@samsung.com
Akshaya Kumar Jain,
Associate Director, Samsung Semiconductor (SSIR), Bangalore, India
Email: Akshaya.jain@samsung.com

2025
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
INDIA

Introduction – UVM RAL Flow





Get from designer
Register Spec
Yes
Generate Register Model → Register Assistant
Yes
Code Adapter Logic → reg2bus / bus2reg
Integrate Register Model → Single Register, Block Register, Top Block Register
Code Tests / Sequences
Trigger Register Model
reg_seq_item --> bus_seq_item
Access DUT interface and Physical register
Routing Back with the help of predictor path and mirrored value updated

Custom March-Bash Algorithm

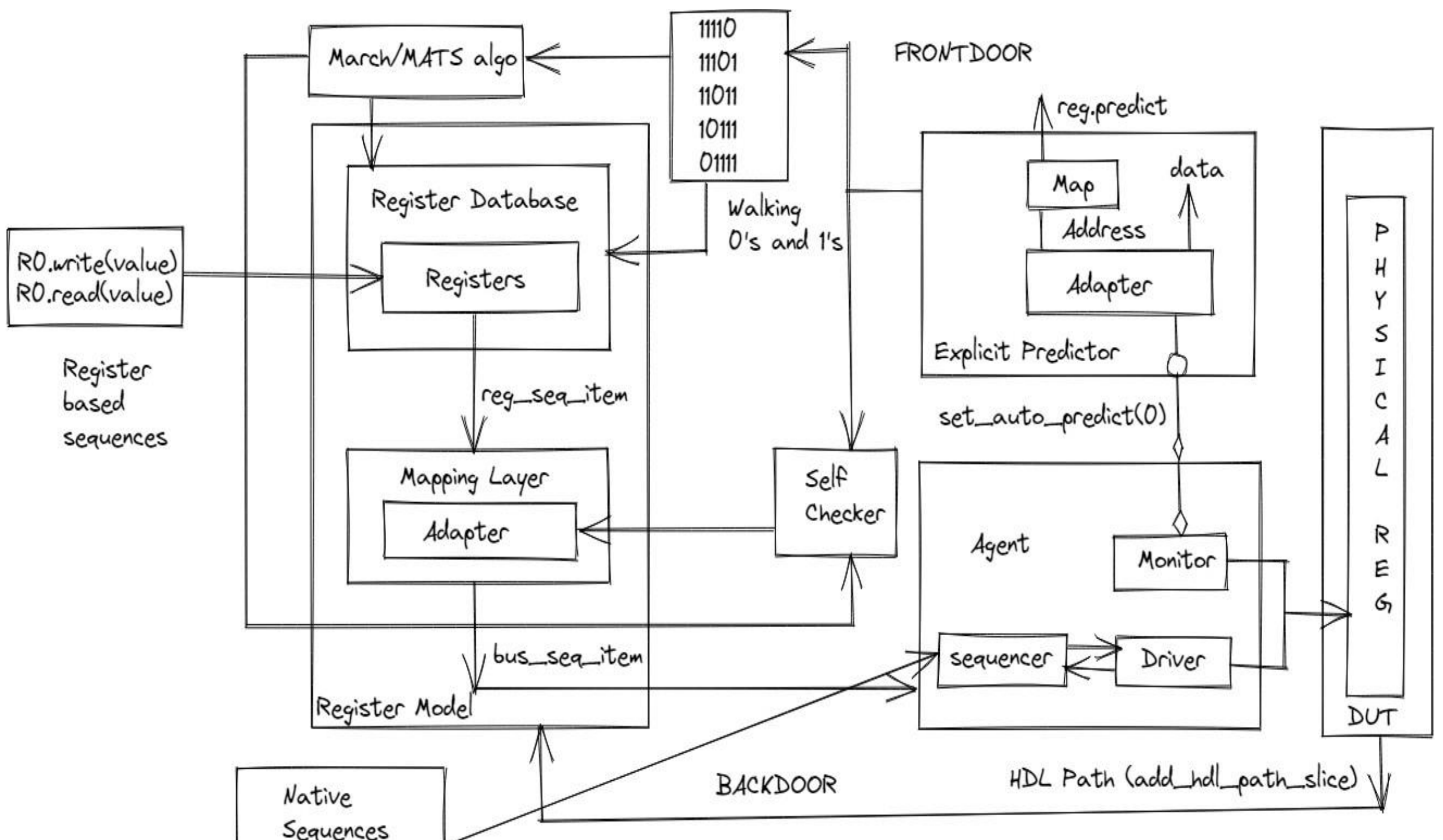


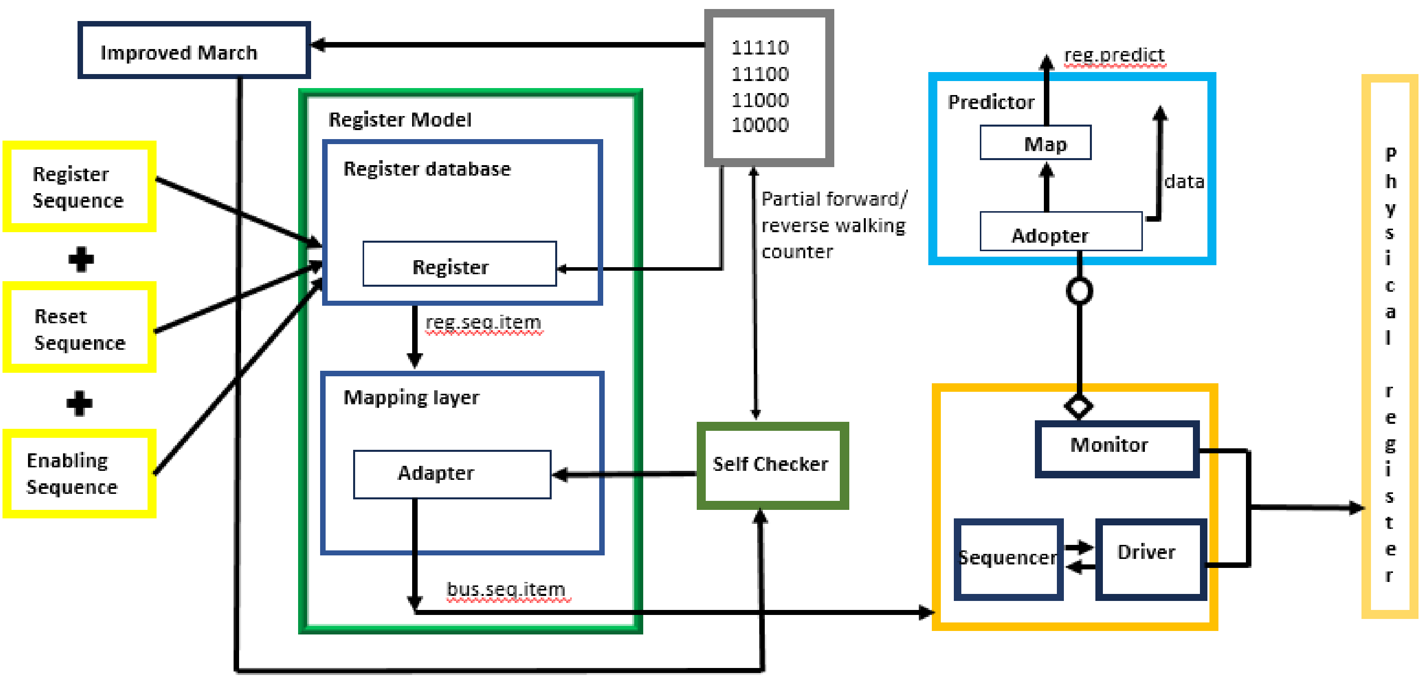
Figure: Custom (March-Bash) Algorithm

March-Bash Execution Steps

- ➢ Write data to trigger the register database within the register model.
- ➢ The register sequence item is converted to a bus sequence item with the help of the Adapter.
- ➢ Write data in the form of a register list pattern to make sure each register within the set is accessed along with the stream of walking 0/1.
- ➢ The data is passed through the DUT interface and accesses the physical layer.
- ➢ Compare the written data with the read data with the help of predictor.
- ➢ Write 0 in increasing and decreasing addressing order.
- ➢ Read 0 with increasing addressing order and Write 1 in increasing addressing order.
- ➢ Read 1 with decreasing addressing order and Write 0 in decreasing addressing order
- ➢ Read 0 in increasing and decreasing addressing order.
- ➢ The data is checked with the predicted data, and then update mirror value.

Advantages of March-Bash and Improved March-Bash

- ➢ Each signal of RTL is being checked for Stuck at Fault 0 and 1
- ➢ Resolves the problem of Transition and Addresses decoding bugs.
- ➢ Reduces the debug effort and improves the efficiency of the testbench.
- ➢ Registers can be claimed as regions in the form of clusters.



Register Verification Checklist

1	Check if all Register R/W access are covered in latest RTL
2	To verify the hard reset value of the registers are covered with latest RTL
3	If a register has only RO accessibility, verify its correctness by writing on it
4	If a register has only WO accessibility, verify the register by reading data from it
5	Check if any register within the register model are failing with stuck at 0 and stuck 1 level issues
6	Check if the invalid or unmapped SFR region is accessed and the system is not stuck
7	Verify the aliasing effect of allthe register --> A particular register can be accessed from multiple address within the same address submap
8	Verify the broadcasting or shared access effect of all the register --> A particular register can affect multiple other subset of register or in short it can be accessed from multiple address maps
9	Verify and resolve the transition effect, addressing decoding issues of all the registers in different address maps
10	Check if all the SFR are accessed with both HWACG enabled and disabled
11	Check if all the SFR are accessed with both Secure and Nonsecure attribute
12	Check if all the auto cleared registers are verified
13	Check if all the status register are accessed and their operation are verified
14	Check if all the quirky or special register are accessed and verified
15	Verify the accessibility of the register using reglist (Frontdoor write --> Frontdoor read)
16	Verify the accessibility of all the register using uvm_reg_access_seq (Mix of frontdoor and backdoor)
17	Verify the accessibility of all the register using only HDL path slice
18	Check if initial value of register is same as FMAP after H/W reset

Conclusion

1. The efficiency and throughput of the register verification in frontdoor is increased by roughly 40% (*calculated based on the feature an algorithm supports*) because it provides additional benefits of hitting more corner case bugs, finding the transition, stuck at fault, and address decoding bugs.
2. The reusability within the testbench is increased by 20% (*calculated based on plug-and-play feature and based on change in address space*).
3. The Improved March-Bash Algorithm reduces the simulation time to ~10% as compared to March-Bash due to the introduction of partial forward / reverse walking 0 / 1.
4. The thoroughness of the Improved March-Bash and the DV checklist is helpful in both functional and code coverage improvement of SFR coverage in an IP from 87.72% to 91.23% (code coverage), and corner case bins are hit while calculating functional coverage.
5. The paper suggested two novel algorithms to access the registers, but for the completeness of the register verification, other inbuilt sequences also need to be executed

REFERENCES

[1] Universal Verification Methodology (UVM) 1.2 User's Guide by Accellera
[2] A unique March Test Algorithms for the widespread realistic memory faults in SRAM – IEEE – A.Benso

SAMSUNG