Novel Use of Symbolic Map Based Constraints for Synchronization Block Verification Using Formal Methods

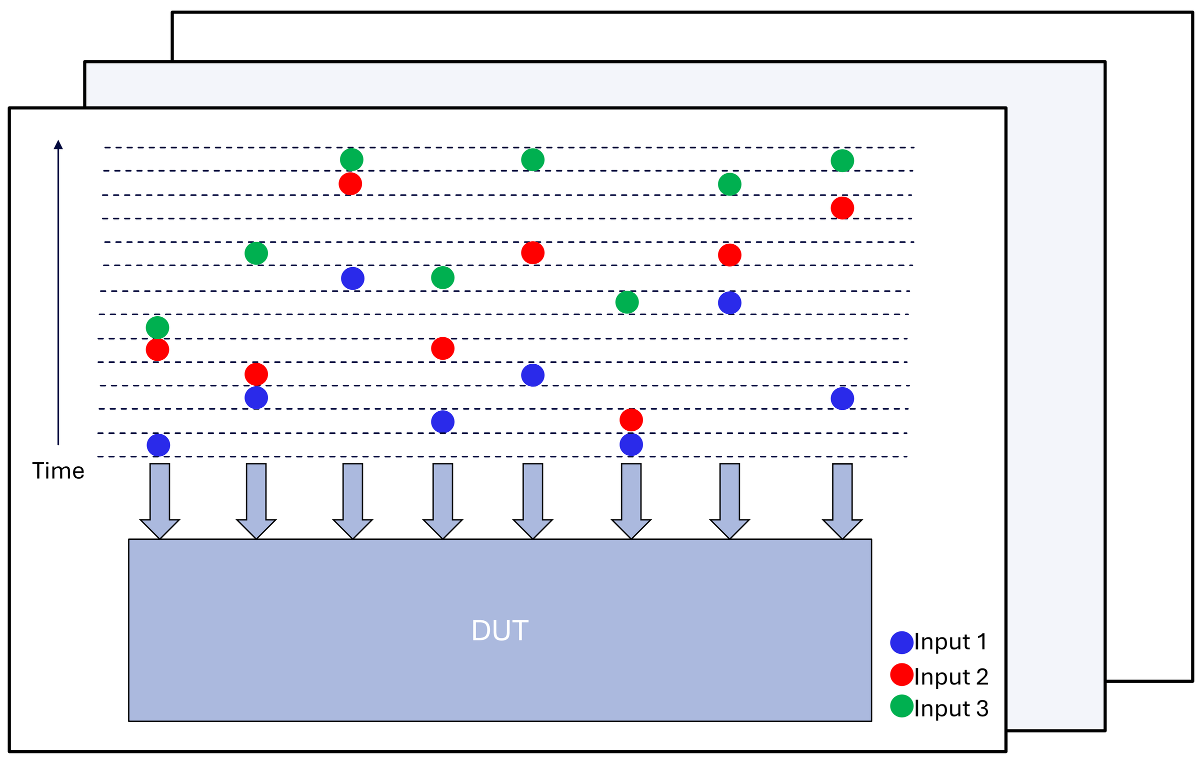
Vikas Vikram Singh, Sanjana Jain, Patnala Naga Sai, Ipshita Tripathi

Qualcomm India Pvt Ltd

accellera SYSTEMS INITIATIVE™

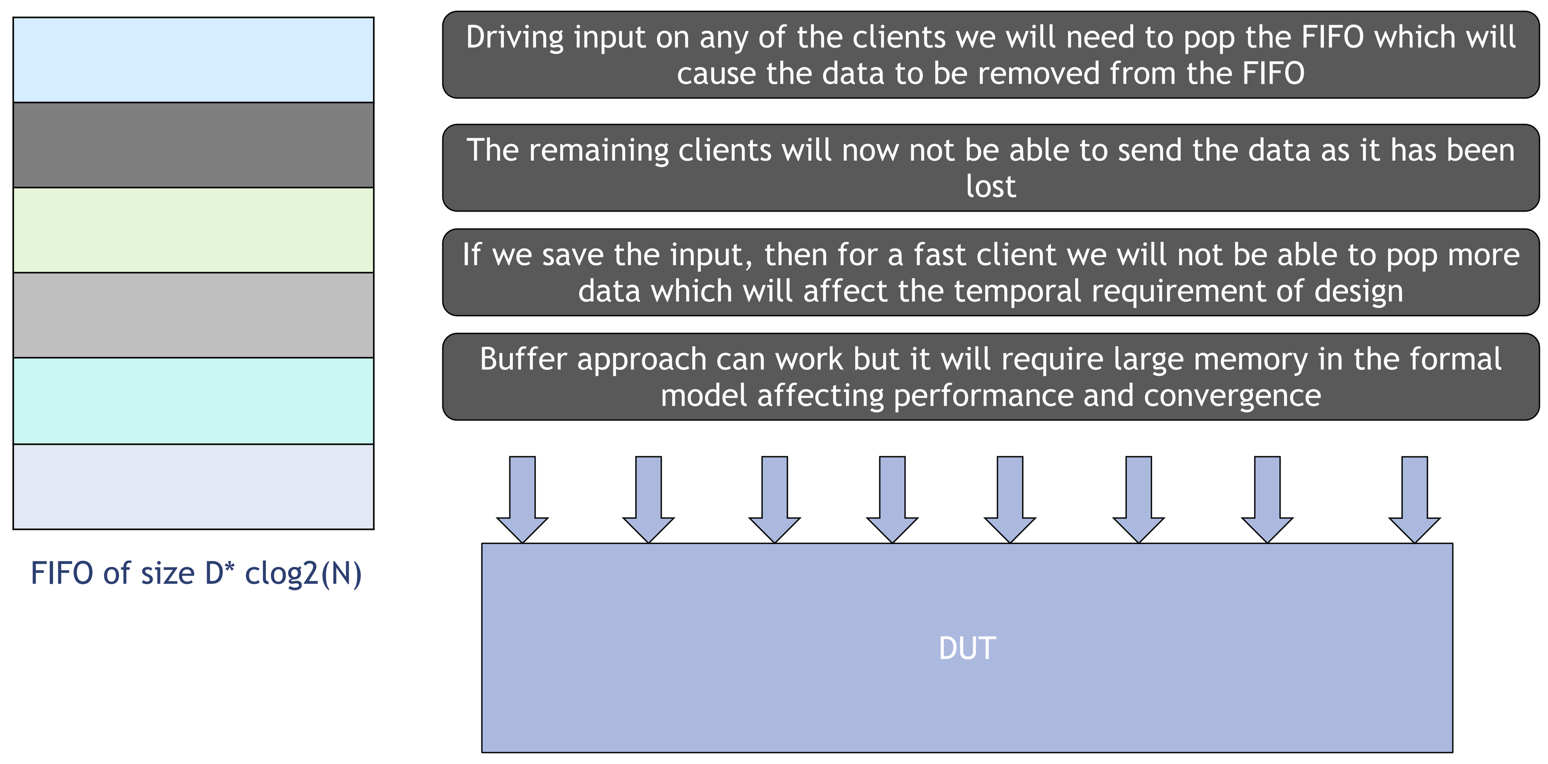
2025 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION INDIA

Problem Statement/Introduction



- Design requirement to get exact same sequence of inputs on all N interfaces but they can arrive temporally at different time
- FPV environment is good at randomizing the inputs at the interfaces but here we need to maintain strict ordering of inputs else design will hang
- Regular FIFO based or simple symbolic based inputs will not be useful in constraining the inputs

Why will a FIFO or Buffer-based approach will not work?

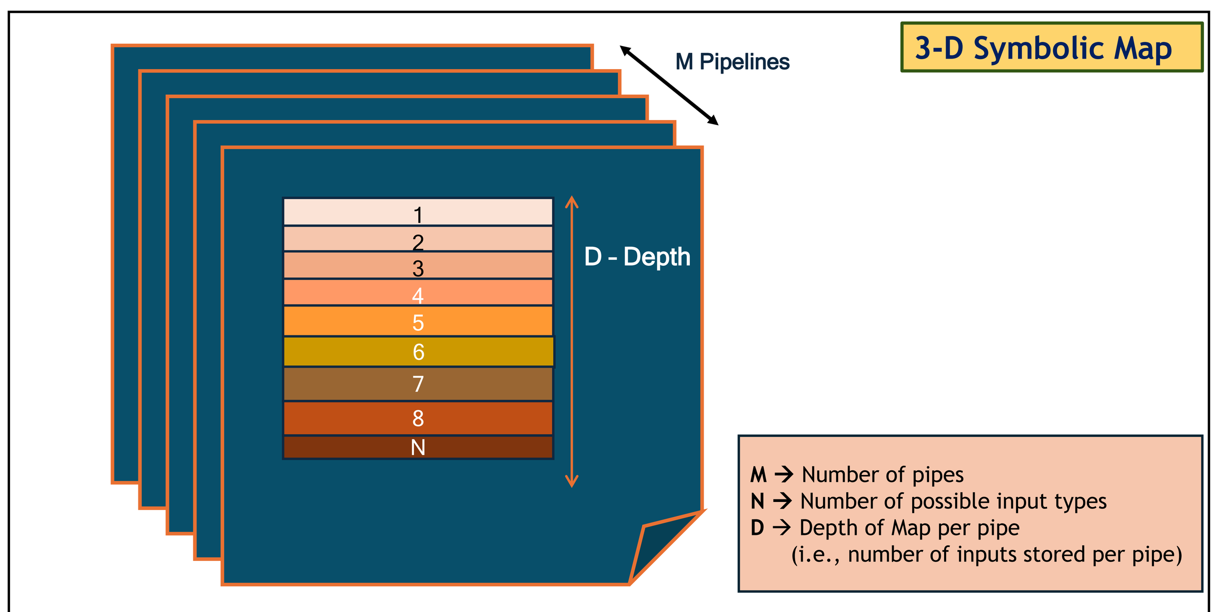


Why not a buffer-based approach?

- Buffer based approach involves addition of a buffer which contains a random sequence of inputs preloaded
- It would have ensured the completeness of the state space but have added a lot of complexity because of the number of flops needed thereby causing convergence issues and affect the performance of the formal testbench

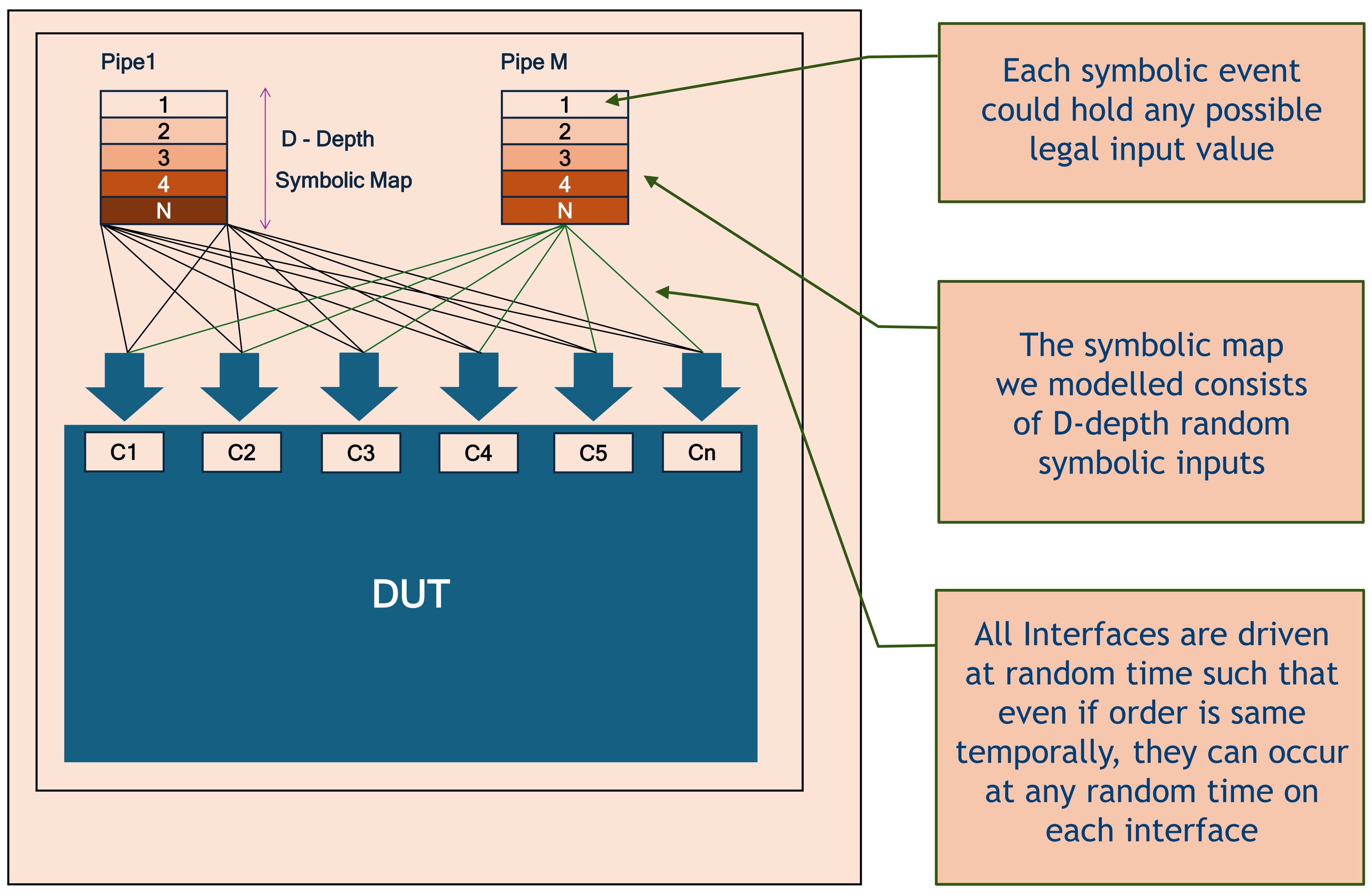
What is a Symbolic Map

- Symbolic Variable is a random constant variable which consist of a random legal values
- Symbolic map in our case is a 3-dimensional array of random constant legal values where
  - N is the number of legal inputs,
  - D represent the total number of outstanding inputs that the design can handle and
  - M represents the parallel streams or pipes of data that need to maintain ordering within the stream but can be random across the streams or pipes



Symbolic Map concept with a **random but stable input tracked across all pipes**

Implementation Details



Results Table

Using this methodology, we were able to verify the following total number of possible combinations of input sequences

Per Pipe, Per Client:

- ❑ Each of the **D** Map slots can hold any of the **N** input types.
- ❑ So, number of combinations per pipe per client: N^D

Per Client (Across M Pipes):

- ➢ Each pipe is independent, so combinations multiply:

  $(N^D)^M = N^{DM}$

- Allowing for temporal randomness allows extensive coverage of these N^{DM} combinations and provides good stress testing of design-under-test
- We ran this methodology on a legacy design which had seen multiple silicon, but we were able to find 2 legacy issues which were due to corner case scenarios

Conclusion & Future Work

- Using this methodology, we increased the coverage of possible combinations tested
- Using the symbolic map, we were able to keep the formal model light and ensure our checks reached the desired depth thereby helping us exercise all possible combinations
- In future we plan to create a reusable plugin and extend this methodology to generalize for N dimensions and allow variable depth across the N dimensions so that this methodology can be used for any case where strict ordering is required

REFERENCES

[1] Souri, A., Rahmani, A.M., Navimipour, N.J. *et al.* A symbolic model checking approach in formal verification of distributed systems. *Hum. Cent. Comput. Inf. Sci.* 9, 4 (2019)

[2] Pandey, Manish and Raimi, Richard and Bryant, Randal E. and Abadir, Magdy S. "Formal verification of content addressable memories using symbolic trajectory evaluation" Proceedings of the 34th Annual Design Automation Conference, pp. 167 - 172

[3] I.Tripathi, A. Saxena, A. Verma, P. Aggarwal, "The Process and Proof for Formal Sign-off: A Live Case Study", DVCon 2016.

[4] P. Aggarwal, D. Chu, V. Kadamby, and V. Singhal, "Planning for end-to-end formal with simulation-based coverage", Proc. Formal Methods in Computer-Aided Design FMCAD 2011, Austin, TX, USA, 2011, pp. 9-16.

[5] P. Wolper, "Expressing interesting properties of programs in propositional temporal logic", POPL '86 Proc. of the 13th ACM SIGACT-SIGPLAN symposium on Principles of programming languages", pp. 184-19

[6] I. Tripathi, A. Velayoudame, R. Gupta, V. Garipelli, "Deadlock & Livelock Detection in Power Controller using under-constrained Formal Testbench", Proc. Of the Design Automation Conference, 2019

[7] N. Kim, J. Park, H. Singh, and V. Singhal. "Sign-off with Bounded Formal Verification Proofs", DVCon 2014.