Experience of using Formal Verification for a Complex Memory Subsystem Design

Sujeet Kumar,Vandana Goel,Hrushikesh Vaidya,
Ronak Sarikhada
INTEL

accellera SYSTEMS INITIATIVE™

2020 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

INTRODUCTION

➢Memory subsystem is essential part of any SOC, traditional verification doesn't guarantee of catching design corners scenarios.

➢Usage of formal verification for a complex memory subsystem design is not an easy task because of its huge state space of the design.

➢Solving the problems of subsystem verification in structured approach.

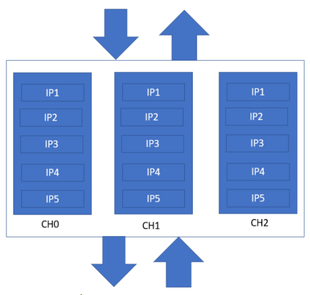
➢Filling the gap of dynamic simulation using formal simulation.

➢Usage of advance approaches to verify a memory subsystem.

➢Faster approach for early design bring up.

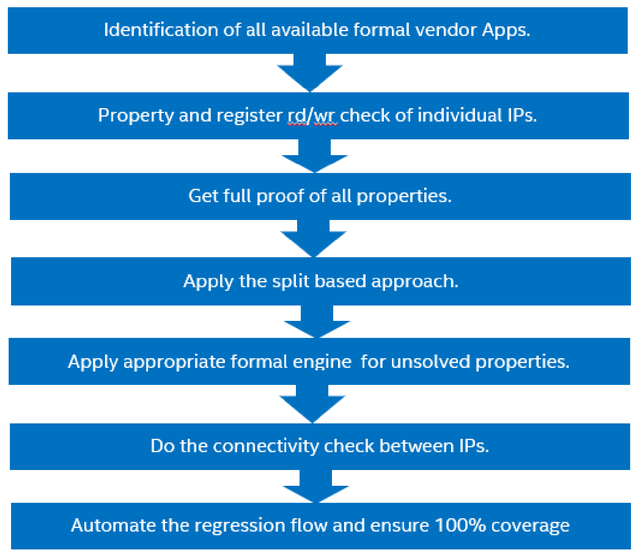
OBJECTIVES

To do formal verification Memory Subsystem



➢ Memory subsystem consists of five IPs. The role of these IPs is to ensure the data transfers from the processor to different flavors of SRAMs.

VERIFICATION FLOW



FORMAL METHODS USED FOR VERIFICATION

- Property Check
- Connectivity Check
- Register Check
- Sequence Equivalence Check
- ABVIP Based Verification
- Split based approach to solve the formal properties

SPLIT BASED APPROACH

FV task can be subdivided for complexity through case splitting. The problem can be defined as

$f(x,y,z\ldots)=a\_x+b\_y+c\_(z\ldots.)$

Function is dependent on its variables.

Splitting is done in such a way that the range of x is kept random when y and z are constant and vice versa. By doing this the problem complexity is minimized

RESULTS

IP name	Total number of properties	Number of undetermined properties						Undetermined solved using split
		RO		WO		RW		
		Before split	After split	Before split	After split	Before split	After split	
IP1	778	33	0	1	0	3	3	91.89%
IP2	652	90	72	18	0	171	63	51.61%
IP3	847	351	27	18	0	54	54	80.85%
IP4	2827	528	356	0	0	340	182	37.67%
IP5	2818	351	183	0	0	383	105	60.76%

BUGS FOUND

➢ Performance: We found a performance, which we would not have found through functional integration verification.

➢ Connectivity: Four issues were found through connectivity verification early in the project.

➢ IP Verif: Two crucial issues were found in the IP implementation. Same were later found in functional verification when scenarios were developed to test the features.

➢ Reg Verif : One issue found.

CONCLUSIONS

➢ Though Formal was applied late in the project, but we got significant results.

➢ 9 issues were found in Architecture, Performance, Register and IP Design.

➢ Automation helped us to reduce verification time.

➢ Faster verification for design bring up.

➢ We got 6x ROI with respect to functional verification

REFERENCES

[1] Erik Seligman, M Achutha KiranKumar, and Tom Schubert, "Formal Verification- An Essential Tool kit for the modern VLSI Design," Elsevier Publications.

[2] C. Jacobi ; K. Weber ; V. Paruthi ; J. Baumgartner ,"Automatic formal verification of fused-multiply-add FPUs",IEEE

Thanks Ketki Gosavi who helped on the resolving tool issues from Cadence