RTL2RTL Formal Equivalence: Boosting the Design Confidence

M, Achutha KiranKumar V, Aarti Gupta, Ss Bindumadhava

Intel Technologies, Bangalore, India

accellera SYSTEMS INITIATIVE™

2014 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE

INTRODUCTION

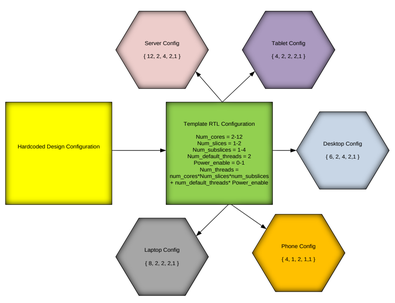
- Increasing design-complexity and Time to Market (TTM) constraints, forces a faster design and validation closure
- Novel ways of identifying and debugging behavioral inconsistencies early in the design cycle mandated
- Addition of incremental features and timing fixes is usually accompanied with the risk of tampering the existing legacy design behavior and insertion of undesirable bugs
- Any number of Dynamic Validation (DV) regression tests can't guarantee complete coverage and mitigate risks.
- DV is convenient but not exhaustive

OBJECTIVES

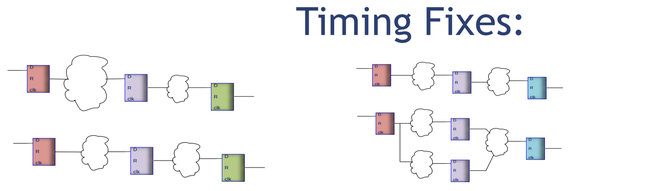
- Provide a static validation methodology which is exhaustive and easy to use
- Formal Verification (FV) techniques to provide a complete coverage of the design with the available resources.
- Formal Equivalence Verification (FEV) to be applied on a wide variety of problems ranging from simple pipeline optimizations to state matching designs to complex logic redistributions.
- Sequential Equivalence mode of checking FEV to enable formal on many more design problems
- Common application of FEV is between the RTL and its synthesized netlist, but RTL2RTL equivalence has much wider scope.

APPLICATIONS

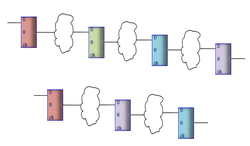
Parameterization:



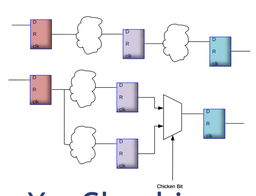
Timing Fixes:



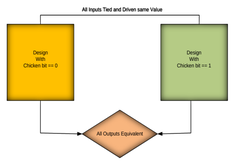
Pipeline Optimizations



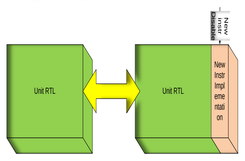
Chicken bit Validation



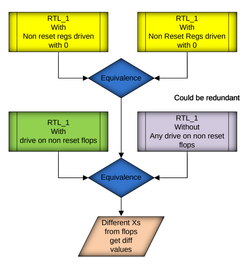
Clock Gating Verification



Legacy behavior checks

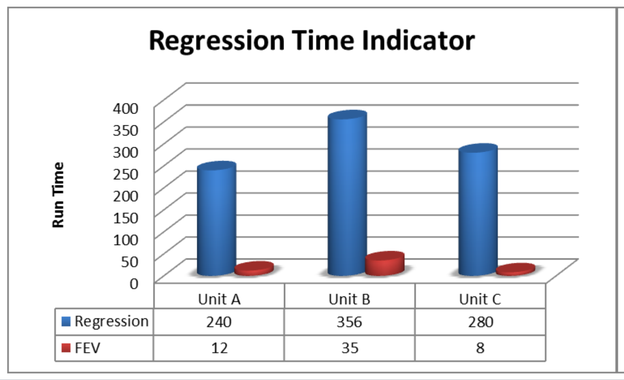


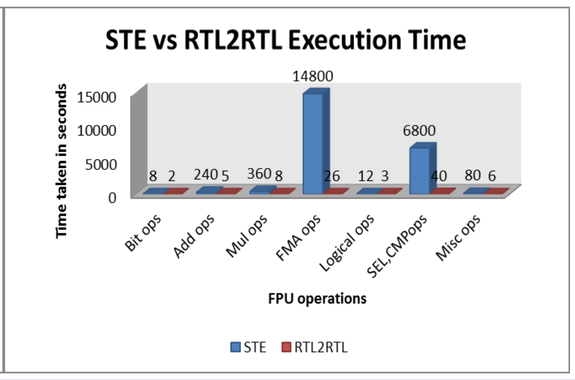
X - Checking



RESULTS

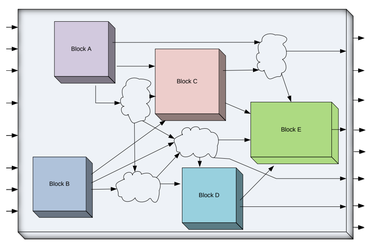
- Very successfully applied RTL2RTL in Intel Graphics design.
- RTL2RTL FV successfully replaced DV regressions for timing fixes, chicken bits, clock gating validation, and legacy checks.
- Replaced weekly 5-day STE regression on netbatch with a 45minute RTL2RTL FV on single machine





COMPLEXITY REDUCTION TECHNIQUES

- Divide and Conquer
- Careful Logic Carving
- Inputs Pruning
- Case Splitting
- Intermediate equivalence
- State Splitting
- Design Abstractions



CONCLUSIONS

- RTL2RTL FV has many more application facets
- Equivalence FV maximizes ROI compared to the investment on DV with minimal resources.
- Stronger equivalence checking tools can improve the gamut of applications where formal can be applied.
- Reduces the barrier for entry into formal world.
- Future work to evangelize the benefits of RTL2RTL

REFERENCES

[1] C. Pixley, *"A theory and implementation of sequential hardware equivalence*", IEEE Transactions on Computer-Aided Design, Dec. 1992, pp. 1469-1478.

[2] K.N. Lalgudi and M.C. Papaefthymiou, "*Fixed-Phase Retiming for Low Power Design*", Proceedings of the International Symposium on Low Power Electronics Design, Aug 1996.

[3] Z. Khasidashvili, M. Skaba, D. Kaiss and Z. Hanna, "*Theoretical Framework for Compositional Sequential Hardware Equivalence in Presence of Design Constraints*", ICCAD 2004, pp. 58-65.

[4] Orly Cohen, Moran Gordon, Michael Lifshits, Alexander Nadel, and Vadim Ryvchin *"Designers Work Less with Quality Formal Equivalence Checking"* DVCON'2010

[4] Nikhil Sharma, Gagan Hasteer and Venkat Krishnaswamy, "*Sequential equivalence checking for RTL models*" EETimes Now.

[5] Carlos Ivan Castro Marquez, Marius Strum , Wang Jiang Chau "*Formal Equivalence Checking between High-Level and RTL Hardware Designs"*

[6] R. Brayton, G. Hachtel, A. Sangiovanni-Vincentelli, "Multilevel logic synthesis", Proc. IEEE, Vol. 78, Feb.1990.

[7] C.A.J. van Eijk, *"Sequential Equivalence Checking without State Space Traversal" DATE 1998*

[8] M Achutha KiranKumar V, Aarti Gupta, and Rajnish Ghughal, "*Symbolic Trajectory Evaluation: The primary validation Vehicle for next generation Intel® Processor Graphics FPU*", Proceedings of the 12th Conference on Formal Methods in Computer-Aided Design (FMCAD 2012).

ACKNOWLEDGEMENTS

Erik Seligman, FVCOE , Intel , Portland

Pradeep Raghavendra, Intel India, Bangalore

AUTHOR CONTACTS

achutha.kirankumar.v.m@intel.com

aarti.gupta@intel.com

bindumadhava.ss@intel.com