Sequential Equivalence Techniques for High Performance Design

Abstract: Quite often in semiconductor industry, when a product is nearing its launch date, most of us have had the déjà-vu situation of performance to time-to-market trade-offs; especially in high-performance designs. Sequential Equivalence Checking opens up possibilities in this area, by enabling performance-tuning related sequential micro-architectural changes to be verified with significantly lower impact on effort estimates and risk. This nascent technology promises to change the way we look at eleventh hour changes.

Sini Balakrishnan

Senior Design Engineer, Infineon Technologies

sini.balakrishnan@infineon.com

Phone: +91.80.41392425 | Fax: +91.80.41392

Sequential Equivalence Techniques for High Performance Design

Introduction

With ever increasing design performance requirements and complexity thereof coupled with the limited time to market, Sequential Equivalence Checking technology is receiving a lot of attention in recent years. SEC minimizes the risk involved in making RTL sequential changes late in the development cycle, without changing the functional behavior.

This paper presents the need for Sequential Equivalence Checking method in our design flow and different sequential micro architectural modifications done on the RTL at the end of design cycle to meet high performance design goals related to area, timing and power. Use of Sequential Equivalence Checking methods at different stages in the high performance design and limitations of SEC are also discussed.

SEC - Role in Design Flow

Sequential Changes in the RTL

Functional verification is a major bottleneck for the designers to refine or modify RTL to meet timing, area and power goals, close to tape-out. This is especially critical since sequential modifications such as FSM re-encoding, retiming and the like to meet high performance design goals will lead to changes in the flops. Different models with same functionality and sequentially different implementations cannot be verified using traditional Combinational Equivalence Checkers (CEC), as it requires one-to-one mapping of I/O's and States. This implies that each flop in the first design must have a mapping counterpart in the other design to apply CEC. Further, simulation based verification of sequentially different new RTL requires modifications in the test-benches and test-suites. It is a time consuming task and may skip corner case bugs in the design.

Sequential equivalence checkers can verify the above mentioned structurally different implementations which do not have one-to-one flop mapping. Assuming we have a Reference Model which is already verified using either simulation based verification or property checking, verification of revised design which has undergone micro architectural sequential changes can be done using SEC, with significantly less effort and risk. SEC proves that starting from initial states all mapped outputs are same for all input sequences. In other words, it is able prove that the revised design is functionally equivalent with the golden reference model.

Fig.1 gives us the overall picture about the usage of SEC to meet high performance design goals. The Golden RTL is a high performance design which met all functional requirements as in the specification and which has been verified

using simulation based verification and/or using property checking. To meet non-functional performance requirements, the golden RTL is SEC compared with the refined RTL to establish functional equivalence.

On gate level implementation backend designer tries to meet high performance goals related to area, power and timing as mentioned in the design spec. If it is difficult to meet performance goals in the back-end stage, golden RTL needs to be modified. The verification for these micro architectural changes is a complicated task without Sequential Equivalence Checking method.

Bounded and Unbounded equivalence checking are two different sequential checking methods. Bounded SEC is used for identifying bugs. Bounded SEC approach is to check equivalency for a bounded number of steps starting from the initial state.

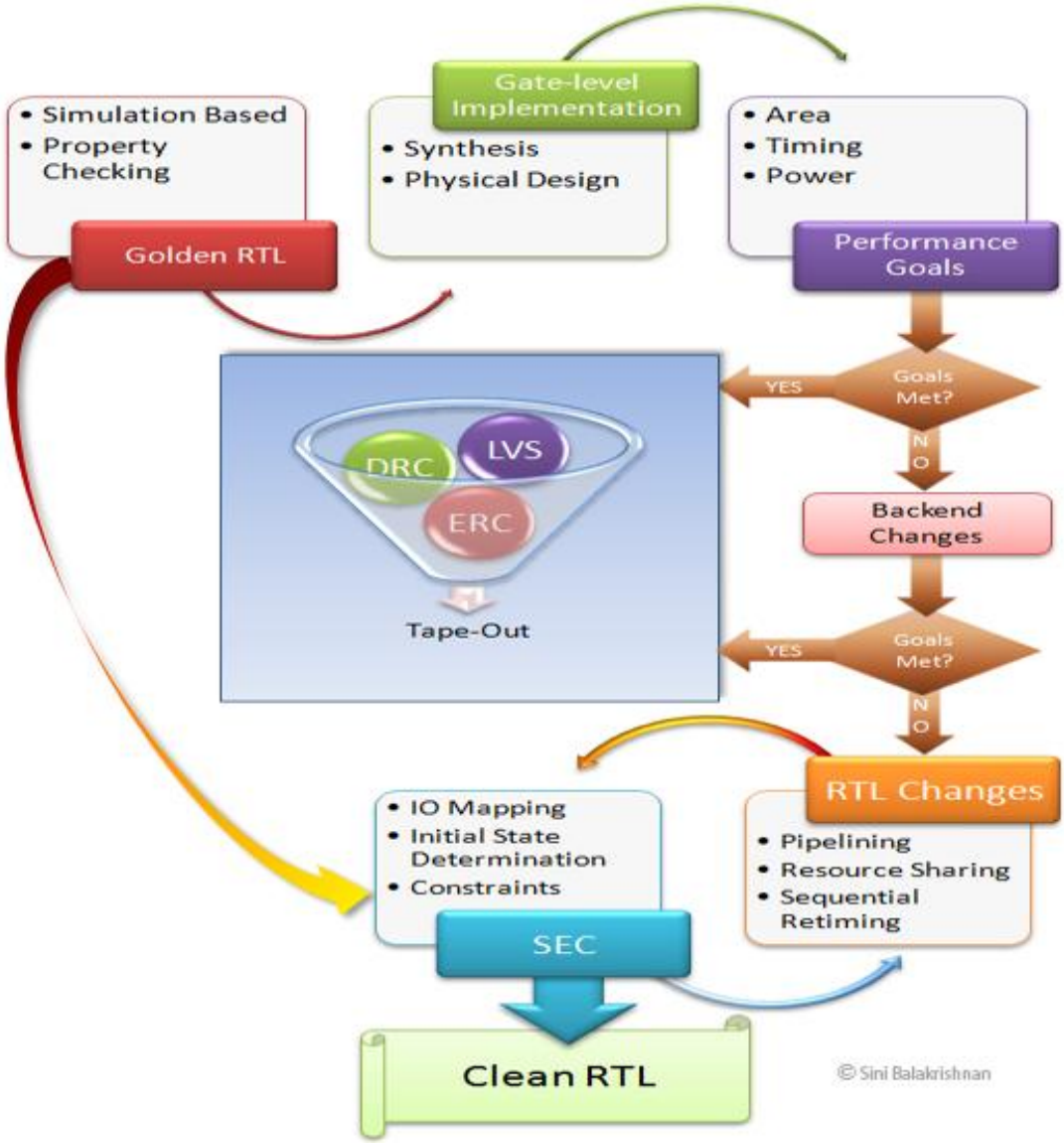

Fig 1: Verifying sequential changes in the RTL using SEC

Behavioral RTL to low level RTL

Behavioral RTL model is written to get high simulation performance. And low level RTL derived from the behavioral model can be verified using SEC which reduces the verification cycles and gives better confidence to the designers.

The simulation based functional verification on the modified RTL eats up lots of verification cycles and it may require modifications of test-suites and test-benches which we had used for verifying behavioral RTL. Further, Combinational Equivalence Checkers cannot handle sequentially different models or where there is no one-to-one mapping for states.

System level model to RTL

If an already verified system level model in SystemC/C/C++ is available, RTL developed using the same specification can be verified using Sequential Equivalence Checking technology. The system level model will be the golden reference model and SEC checks the equivalence of the revised RTL model with the golden reference model.

Common Sequential Changes

As we are aware, it is difficult to achieve aggressive timing, area and power goals for high performance design. If the back-end designer is not able to achieve one of them, the fall-back option would be to modify the RTL by identifying the cause, without changing the functional behavior. Often, we have seen performance goals being ignored as a trade-off vis-à-vis time to market, as it poses a big risk in going back to RTL at final stages of development lifecycle.

Some of the commonly used methods are retiming, pipelining, re-encoding of finite state machines, resource sharing and register duplication and merging.

Retiming is a method for moving or rearranging of registers across combinatorial logic in a design in order to meet maximum timing goal. This is one of the common sequential modifications done on RTL.

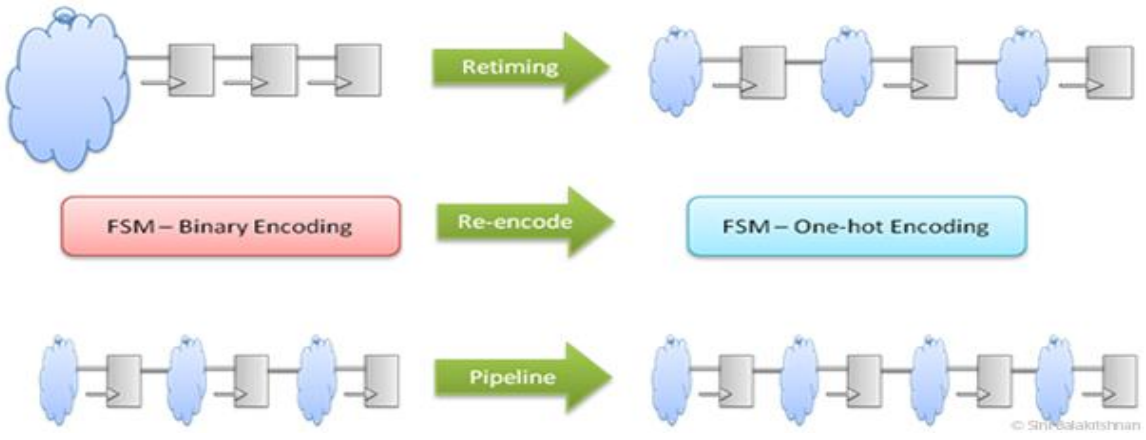

Fig 2: Retiming, FSM re-encoding and adding pipeline stages

Pipelining is another timing related tactic employed by designers. Basic approach is to insert or modify number of pipeline stages to meet timing.

FSM Re-encoding too is used to meet non-functional requirements. To get some of the optimizations on area, timing and power FSM may have to be re-encoded, which also undergo sequential modification on RTL.

All the above changes done on RTL will modify one-to-one state mapping for reference model to modified RTL. Hence Sequential Equivalence Checking, the advanced formal verification methodology is recommended, as conventional methods cannot help in this scenario without making substantial variance on effort estimates.

OneSpin 360º EC™ from OneSpin Solutions and SLEC from Calypto Design Systems are some of the prominent Sequential Equivalence Checking tools available in the market.

Known Limitations

SEC is resource intensive in terms of memory and CPU. SEC tools currently have limitations in handling multimillion gate designs as the technology is still evolving. However, this limitation is almost nullified by the fact that Hierarchical Verification can be done on such designs by verifying the sub-models and verifying the whole by black-boxing the sub-models.

Conclusion

Sequential Equivalence Checkers give designers the flexibility to make design changes without changing the functionality, late in the development process by reducing verification cycle and increasing design confidence. It can also be used to verify the equivalence of behavioral RTL to low level RTL and System level model to developed RTL.

With this advanced formal verification technology to aid, business and technology can reach a more amicable consensus in the perpetual balancing act between perfection and market realities.

References

i. http://www.onespin-solutions.com/360ec.php

ii. http://www.calypto.com/products/SLEC_RTL.html