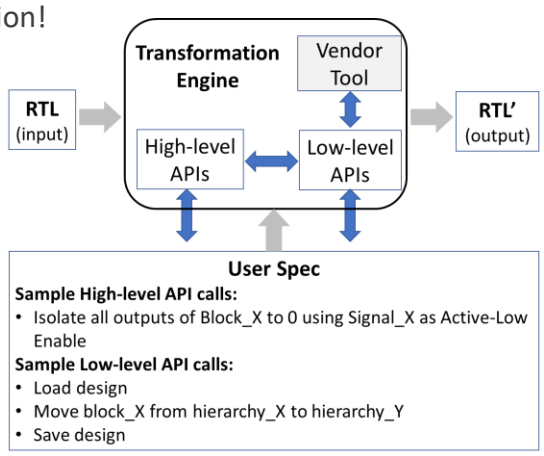
RTL Transformation Methods to Achieve Order of Magnitude TAT Improvement in VLSI Design

Ashfaq Khan, Shuhui Lin, Dan Standring, Adam Cajiao Campos, Soowan Suh, Satish Venkatesan
Intel Corporation, USA

accellera SYSTEMS INITIATIVE
2024 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

INTRODUCTION

With the advancements in the EDA tools, many RTL design activities can now be performed in the form of automated RTL transformation!

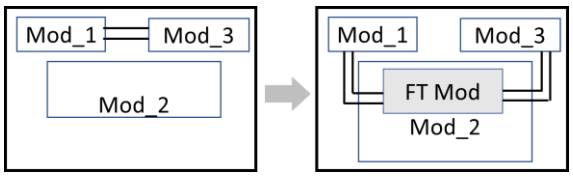


Spec-based automated RTL Transformation

UPFRONT CONSIDERATIONS AND MAJOR APPLICATIONS

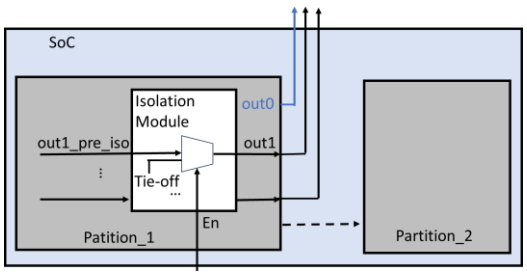
- Upfront considerations
  - Repetitive work with simple spec
  - Maturity of the underlying tool
  - In-house development of high-level APIs
  - Validation of tool-generated RTL
- Major applications
  - Hierarchy Restructuring
  - Feedthrough and Tie-off Insertion
  - Clock/Reset Re-distribution
  - Topology Restructuring
  - Fuse Isolation Insertion

APPLICATION EXAMPLES



Pseudo spec:
From: Mod_1
To: Mod_3
Signals: All port2port connections
Via: Mod_2

Feedthrough insertion



Pseudo spec:
Output hierarchy: SoC
Origin: [partition_1]
Enable: En
Exception: [out0]

Fuse Isolation insertion

RESULTS

Presented methods were deployed for production use in the development of the latest two generations of Visual Computing segment products of our company

For most of the activities, the TAT reduction was generally over 10x. There was also corresponding reduction in NRE cost, which are not reported here.

Examples of TAT reduction due to automated RTL transformation in a production VLSI design

RTL Transformation	TAT Reduction
Hierarchy Restructuring	4-6 Weeks
Feedthrough and Tie-off Insertion	5-7 Weeks
Fuse Isolation Insertion	2-3 Weeks

CONCLUSIONS

RTL transformation results in a paradigm shift in the entire design process, resulting in order of magnitude TAT improvements.

The EDA industry is at a point where these methods can be scaled to many designs to yield similar improvements.

Some of the major remaining challenges include handling latch-on content, collaterals that accompany RTL and a need to remain up to date as the RTL changes and matures, e.g., UPF.

Consuming RTL content from various sources and generating add-ons to RTL (e.g., post-processing needed to add any ifdef statement in RTL) is also a challenge that sometimes require ad hoc efforts on the users' part.

REFERENCES

[1] "RTL Restructuring Issues", Ed Spering, July 18th,2023. https://semiengineering.com/rtl-restructuring-issues [Website visited on Aug 18, 2023]

[2] "System to catch Implementation gotchas in the RTL Restructuring process", DVCon 2016

ACKNOWLEDGMENT

We thank multiple Graphics teams at Intel that pioneered various RTL transformation technologies described in this poster.