FastRoute: Adapting Academic Code for Use in the Design of Production ICs

Eder Monteiro and Ricardo Reis

Introduction

We present:
- FastRoute history: the authors, the original routing flow and its limitations
- The OpenROAD project: brief introduction to the project and to the RTL-to-GDSII flow
- Adaptations and enhancements to integrate FastRoute to the OpenROAD flow
- New features and ongoing work

The original tool

Academic code
Developed originally by Pan et al.1 for a VLSI CAD contest.
There are five versions2 of FastRoute, each adding new techniques to improve quality of results → we have adapted the latest version3
Input and output file formats are from the ISPD 2008 Global Routing Contest4
Routing flow and techniques
Congestion-driven and via-aware Steiner tree construction
Multi-source multi-sink maze routing
Spiral layer assignment using dynamic programming
Virtual capacity adjustment

The OpenROAD Project

An ambitious project that aims to develop an open-source 24-hour RTL-to-GDSII flow.
Composed of dozens of contributors, mostly students from different universities around the world and a few industry veterans
Different teams divided according to the universities, each one responsible for different tools of the flow
The Brazilian team gives support for four tools in the flow, at the steps of floorplan, clock tree synthesis and routing

Enhancements

Initial limitations
Tool usage not “user-friendly” → focus on the VLSI CAD contest
Hard-coded max number of pins, routing layer direction, grid size, pin layers
Input and output file formats not matching the formats adopted in the OpenROAD flow

Implemented features
Interface between the formats used in the project and FastRoute original code, integrating it with the OpenROAD project database7
Scalability: removed hard-coded information, allowing nets with any number of pins, routing layer direction according technology, and dynamic grid size
Routing resources model: correct calculation of “true routing resources” according to both technology attributes (spacing rules, transition layers, track pitches) and design attributes (routing obstacles, macro blocks, pin geometries)
Control of the global routing resources: API to define per-layer and per-region resources reductions + studies and experiments of to find proper resources configuration for each technology node studied in the project
Routing guides example.
The first routing layer is usually avoided, used only for pin access. In this example, layers two (red), three (green) and four (orange) are used

New features and future work

Clock net routing
Allows specifying a different configuration for clock nets
Routing layer range sets the minimum and maximum routing layers
Routing topology created by FLUTE7 or PDRRev8
Antenna repair
Preemptive approach to mitigate antenna rule violations
Long routing segments may created antenna violations in the detailed routing → they can be inferred in the global routing result
Use of an antenna checker tool to find the nets with violations → fix the violated nets by inserting diodes
Parasitics estimation
Used by the gate resizer and static timing analysis tool to identify transition and capacitance violations from global routing result
Future work: use FastRoute for congestion- and timing-driven placement and solidify tool support for different technologies

References