Transforming Arbitrary Structures into Topologically Equivalent Slicing Structures

Olivier Peyran, Wenjun Zhuang,
Institute of High Performance Computing, Singapore

Abstract

Floorplanning is an important step of IC design. Traditionally, floorplan representation has been segregated between slicing and non-slicing structures. We present a heuristic that translates any arbitrary structure into a slicing one, topologically equivalent to the initial one after a 1-D compaction.

1. Introduction

Very Deep Sub Micron technologies have changed the criteria of circuit optimisation. IC designers need reliable physical information at early design stages. In this context, floorplanning has become a critical step in the design flow.

Floorplan representations can be segregated in two categories: slicing and non-slicing. M. Lai and D.F. Wong proved that slicing tree is a complete representation for *maximally* compacted floorplans [1]. We propose a heuristic to transform *any* non-slicing floorplan into a slicing one that would be equivalent to the initial structure after a 1-D compaction. The two-stage algorithm (slicing structure generation and slicing structure modification) is presented in the next sections

2. Slicing structure generation

An arbitrary placement is usually not a slicing structure. To generate the slicing structure, we recursively force the creation of slices by selecting a scan line and moving its consisting blocks in either side of the line. The selected line is called the cut line. In order to have a global approach of the problem, we look for "slicing patterns" in the initial placement. The patterns we are searching for are block clusters bounded by a rectangular box. We call such clusters bounding boxes (see Fig. 1).

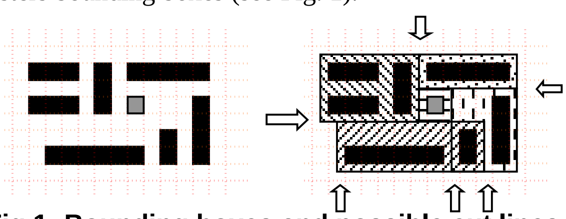

Fig.1: Bounding boxes and possible cut lines

Since bounding boxes should eventually become the partitions of the slicing structure, we recursively look at the extremities of bounding boxes for possible cut lines. The final choice of the cut line is not detailed here. Intuitively, the inner boxes that are not considered for a given slice will be considered on a subsequent step.

3. Slicing structure modification

The "search-and-move" algorithm searches for good candidates among the slicing structures previously generated and modifies them until equivalence is reached. The main steps of the algorithm are to find a good candidate, modify it, and then evaluate the results.

The candidate selection is made according to its similarity with the original structure. The candidate critical path is then studied and the blocks that should be moved are identified. The glitch in the critical path is removed by moving the problematic blocks to a new position (*i.e.* relatively to different blocks) where they are less likely to modify the structure. The result is then evaluated. If the result is not satisfying, new candidates are searched for.

4. Results

Fig. 2 shows a result. Our approach managed to find an equivalent structure in 98% of 1700 placements. More detailed algorithms and results are available on request [2].

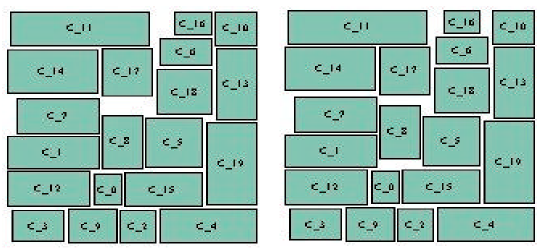

Figure 2: Minimally compacted floorplan. Left: targeted structure, right: result

References

[1] M. Lai, D.F. Wong, "Slicing Tree Is a Complete Floorplan Representation", proceedings of DATE'2001.

[2] O. Peyran, W. Zhuang "Transforming Arbitrary Structures into Topologically Equivalent Slicing Structures", internal report.