Hierarchical Layout Operations

Yasmine Badr

NanoCAD Lab
UCLA
Hierarchical vs. Flat Layout

Flat

Hierarchy

XNOR
Mux2
Why Hierarchy?

• Less memory
  – Same cell instantiated multiple times

• Less processing time
  – Repeated structure verified “once”
  – To determine which geometries are close enough for a DR violation ➔ sorting

• Chip with \( n \) polygons
  – Flat: \( O(n \log (n)) \)
  – Hier on 2 cells: \( 2((n/2)\log(n/2)) + \) cell interaction overhead
Why Hierarchy (cont’d)

• Better results in verification (designer perspective)
  – E.g. DRC: reporting a much smaller set of violations ➔ easier debugging
Challenges in Hierarchical Processing

• Cell neighborhood/instantiation affect results
  – Also Cells may overlap
  – Different orientations of cells

• Migration/Compaction and DP:
  – Multiple instances of same cell \(\rightarrow\) different sets of constraints BUT ONE output cell
  – Working on **hierarchical** view \(\rightarrow\) different results from working on **flat** view of same layout
Challenges in Hierarchical Processing (cont’d)

• Hierarchical DRC is NP-complete [2]
Example [2]

a: cell
b: hierarchical layout
c: Abstract of cell

Y.Badr
Hierarchical Verification Flow

1. Check all leaf cells.
2. For each cell
   – build an abstract:
     a (hopefully) simpler version of the cell that only contains features that are needed for checking cell interactions.
3. Start at hierarchy level 1 (from leaf)
4. Verify cells of current hier. level:
   a. Substitute with cell abstracts
   b. Run flat verification algorithm on resulting data
5. Prepare abstract for the next higher level
6. Repeat till top of hierarchy
References