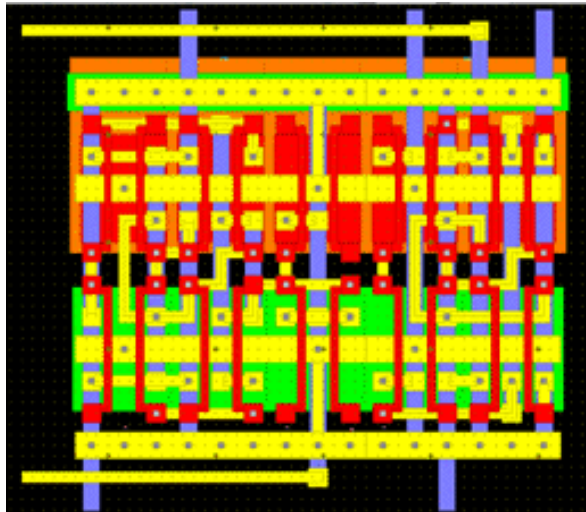# Hierarchical Layout Operations

Yasmine Badr
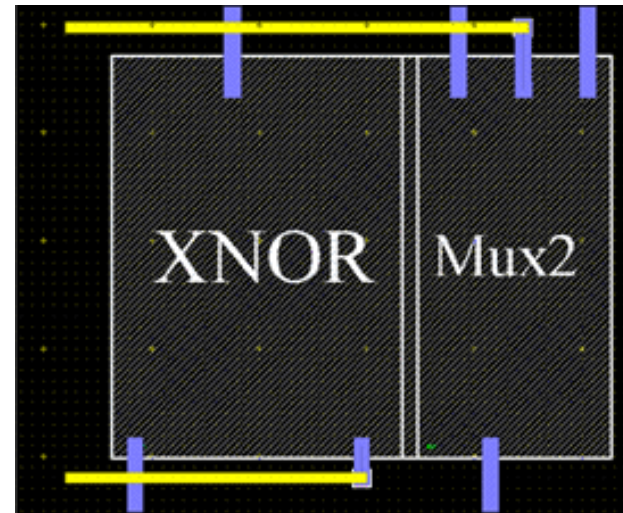
NanoCAD Lab
UCLA

Hierarchical vs. Flat Layout

Flat

Hierarchy

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 ➜ different sets of constraints BUT ONE output cell
  - Working on **hierarchical** view ➜different results from working on **flat** view of same layout

Challenges in Hierarchical Processing (cont'd)

- Hierarchical DRC is NP-complete [2]

Example [2]

Minimum spacing

(a)

(b)

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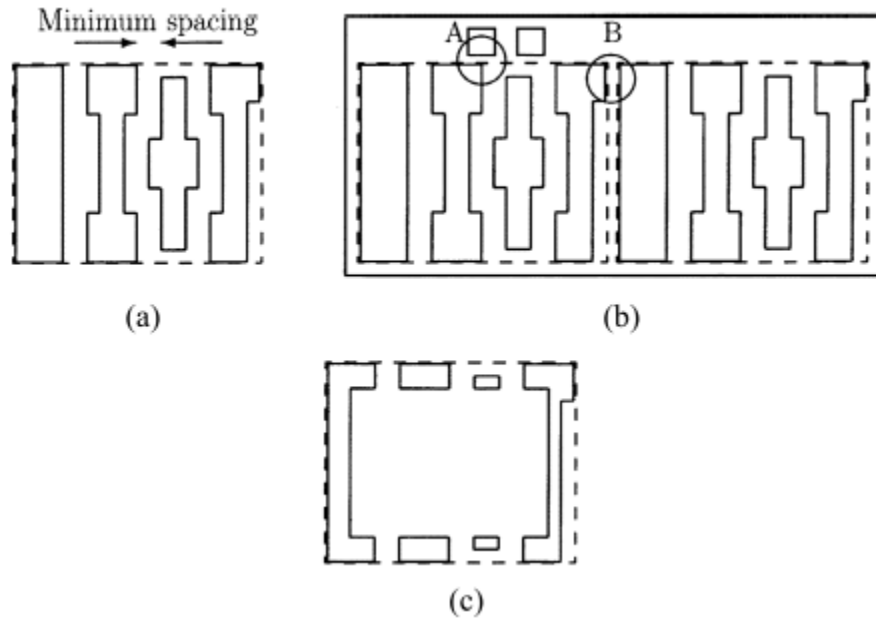
a: cell
b: hierarchical layout
c: Abstract of cell

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

[1] Telle Whitney; "A Hierarchical Design Rule Checker"; M.Sc. Thesis, Caltech, 1981

[2] Louis K. Scheffer; "Some Conditions Under Which Hierarchical Verification is O(N)"; TCAD; 2003

[3] http://www.semiwiki.com/forum/showwiki.php?title=Semi+Wiki:Design+Rule+Checking+DRC+Wiki

[4] "Hierarchical Layout Verification"; Todd J. Wagner, Intel; DAC; 1984