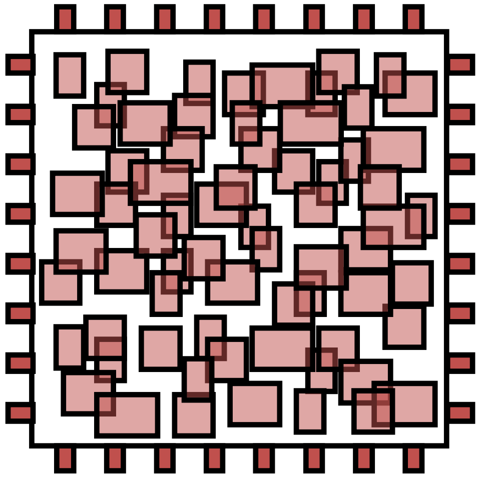
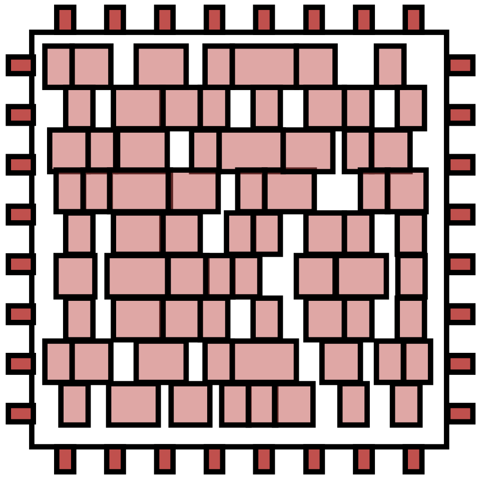
Steps for Modern Circuit Placement

- Placement is usually divided into several more manageable steps:

1. **Global placement**
   - Generating a rough placement solution that may violate some placement (e.g., non-overlapping) constraints
2. **Legalization**
   - Modifying the rough placement to satisfy all constraints by local module movement
3. **Detailed placement**
   - Further improving the legalized solution by some local techniques





Placement Objectives

- Total wirelength
- Routability
  - Congestion-driven placement
- Performance
  - Timing-driven placement
- Power consumption
  - Power-driven placement
- Heat distribution
  - Thermal-driven placement

Underlying Issues for All Formulations

- Underlying issues for all variations of placement formulations are the same:
  - **Wirelength needs to be minimized** for different objectives
  - **Modules have to be properly distributed** for different design styles and for thermal-driven placement
- Will focus on total wirelength (HPWL) minimization

Other Works in Min-Cut Placement

- **Capo** [DAC-00]
  - http://vlsicad.eecs.umich.edu/BK/PDtools/Capo/
  - Standard cell placement, fixed-die context
  - Pure recursive bisectioning placer
  - Several techniques to produce good bisections
  - Produce good results mainly because
    - Breakthroughs in mincut bisection
    - Pay attention to details in implementation
  - Implementation with good interface (LEF/DEF and GSRC bookshelf)
  - A lot of extensions and improvements since the first implementation
- **Fengshui** [GLSVLSI-01,DAC-01,ICCAD-03]

Capo

- Recursive bisection framework
  - Multi-level FM for instances with >200 cells
  - Flat FM for instances with 35-200 cells
  - Branch-and-bound for instances with <35 cells
- Careful handling of partitioning tolerance
  - Uncorking: prevent large cells from being the first modules in a bucket of the FM algorithm
  - Repartitioning: several FM calls with decreasing tolerance
  - Block splitting heuristics: Higher tolerance for vertical cut
  - Hierarchical tolerance computation: instance with more whitespace can have a bigger partitioning tolerance

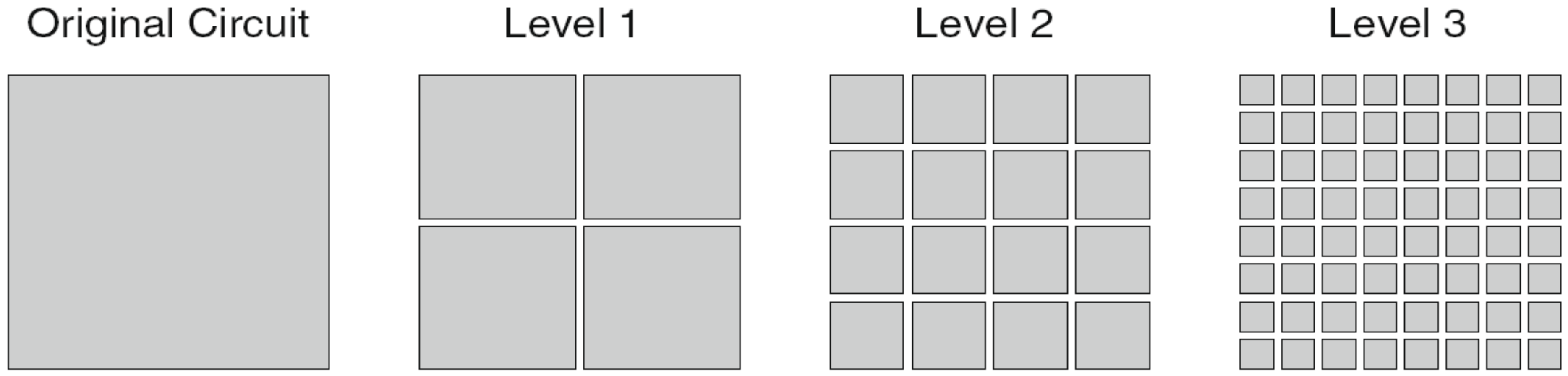
Hybrid Approach: Partitioning + Simulated Annealing

- Simulated annealing based placement produces good solution for small circuits
- But it becomes slow and poor in quality for larger circuits

- **Dragon** [ICCAD-00] takes a hybrid approach that combines simulated annealing and partitioning
  - Recursive partitioning to reduce the problem size
  - Simulated annealing to refine the solution generated by partitioning

Dragon

- Top-down hierarchical placement
  - hMetis to recursively quadrisect into 4^h bins at level h
  - Swapping of bins at each level by SA to minimize WL
  - Terminate when each bin contains < 7 cells
  - Then at the final stage of GP, switch single cells locally among bins by SA to further minimize WL
- Detailed placement is done by a greedy algorithm


Original Circuit
Level 1
Level 2
Level 3