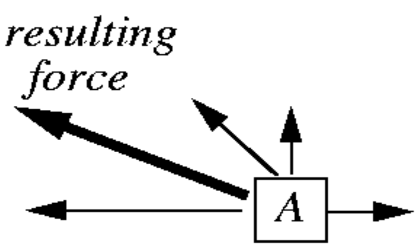
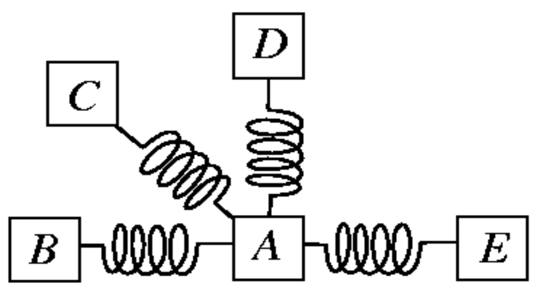
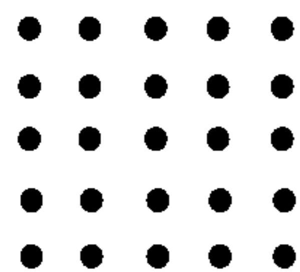
Placement by the Force-Directed Method

- Hanan & Kurtzberg, “Placement techniques,” in *Design Automation of Digital Systems*, Breuer, Ed, 1972.
- Quinn, Jr. & Breuer, “A force directed component placement procedure for printed circuit boards,” *IEEE Trans. Circuits and Systems*, June 1979.
- Reducing the placement problem to solving a set of simultaneous linear equations to determine equilibrium locations for cells.
- Analogy to Hooke’s law: $F = kd$, F: force, k: spring constant, d: distance.
- Goal: Map cells to the layout surface.





$F = kd$



layout surface

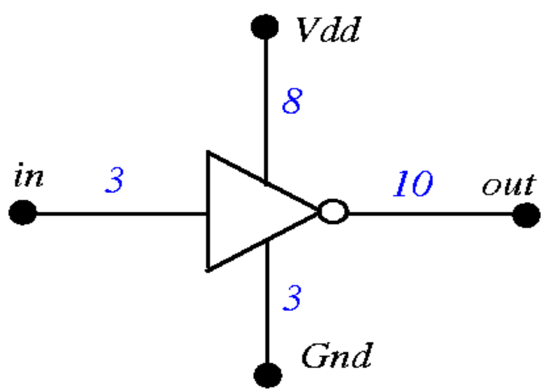
Finding the Zero-Force Target Location

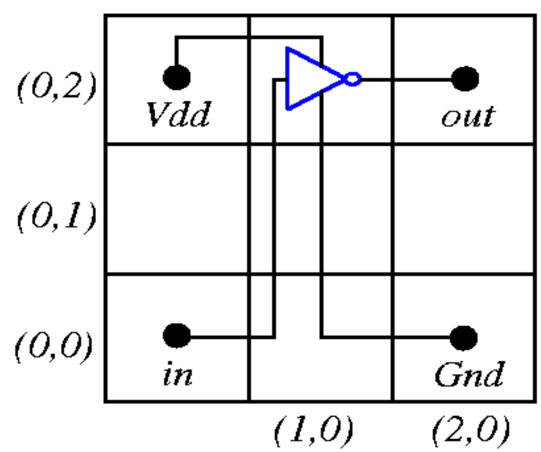
- Cell i connects to several cells j's at distances d_{ij}'s by wires of weights w_{ij}'s. Total force: $F_i = \sum_j w_{ij} d_{ij}$
- The zero-force target location $(\hat{x}_i, \hat{y}_i)$ can be determined by equating the x- and y-components of the forces to zero:

$$\sum_j w_{ij} \cdot (x_j - \hat{x}_i) = 0 \quad \Rightarrow \quad \hat{x}_i = \frac{\sum_j w_{ij} x_j}{\sum_j w_{ij}}$$

$$\sum_j w_{ij} \cdot (y_j - \hat{y}_i) = 0 \quad \Rightarrow \quad \hat{y}_i = \frac{\sum_j w_{ij} y_j}{\sum_j w_{ij}}$$

- In the example, $\hat{x}_i = \frac{8*0+10*2+3*0+3*2}{8+10+3+3} = 1.083$ and $\hat{y}_i = 1.50$.



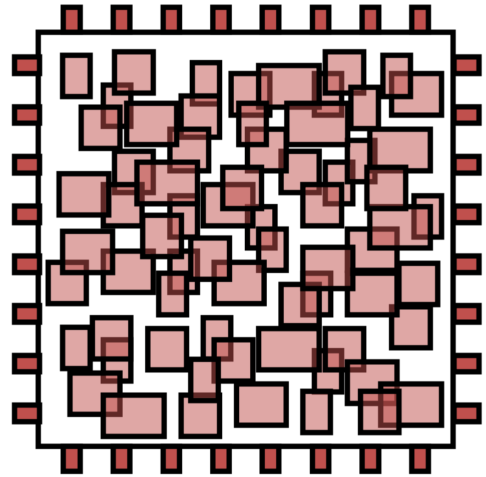


Force-Directed Placement

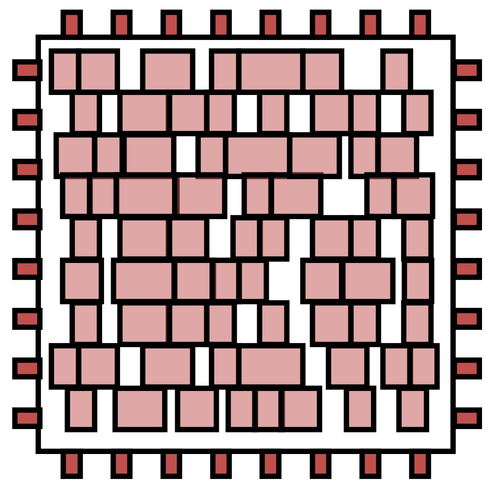
- An iterative improvement approach:
  - Start with an initial placement.
  - Repeat the following until convergence.
    - Select a “most profitable” cell p (e.g., maximum F, critical cells) and place it in its zero-force location.
    - “Fix” placement if the zero-location has been occupied by another cell q.
      - Options:
        - **Ripple move**: place p in the occupied location, compute a new zero-force location for q, …
        - **Chain move**: place p in the occupied location, move q to an adjacent location, …
        - Move p to a free location close to q.

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

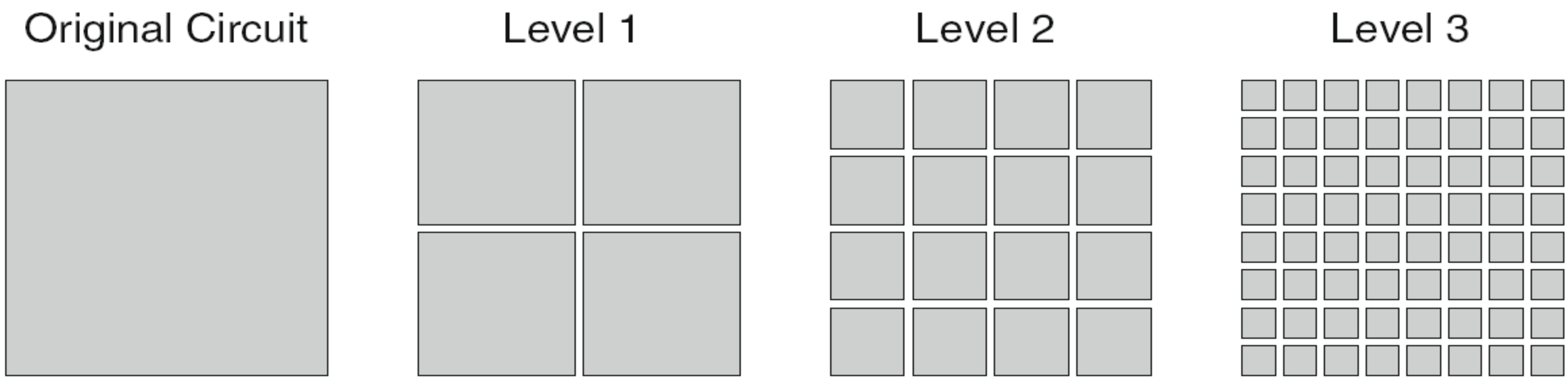
Hybrid Approach: Partitioning + Simulated Annealing

- 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



Analytical Approach

- Write the placement objective and possibly placement constraints as *analytical* functions of module coordinates
- Therefore, formulate the placement problem as a mathematical program
- Many variations
  - Different ways to formulate the problem
  - Different ways to solve the resulting mathematical program

HPWL as Analytical Function

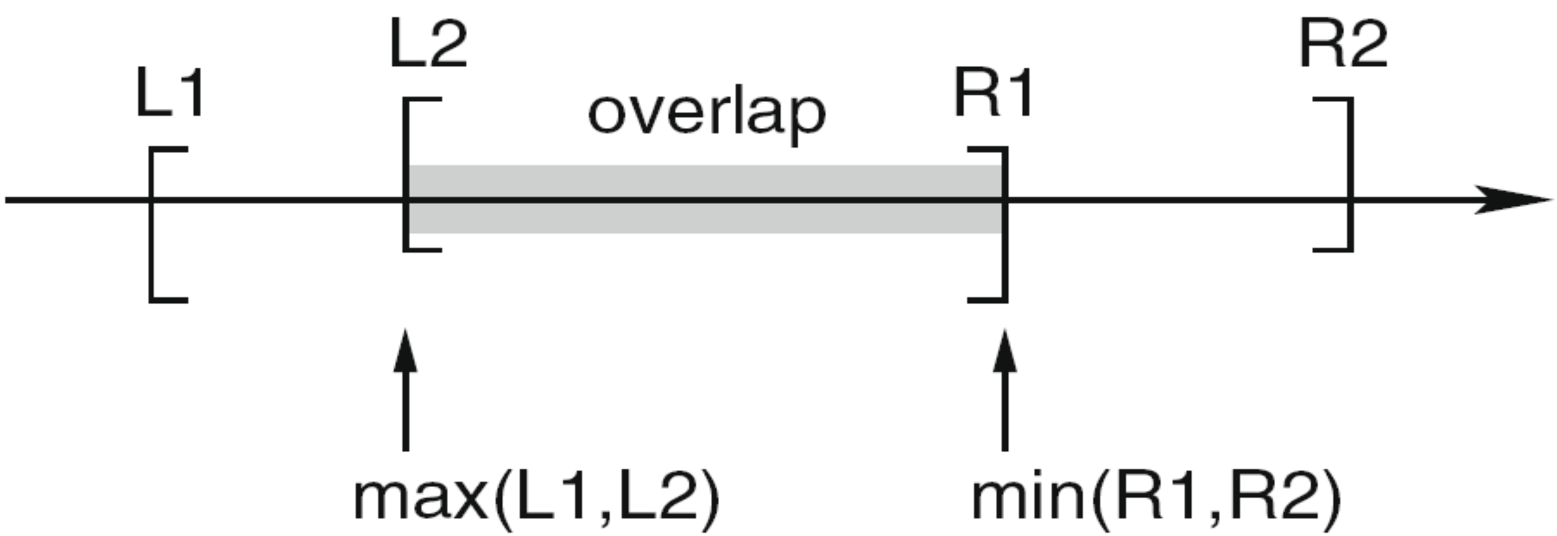
- HPWL of net e

$$\mathrm{HPWL}_e(x_1, \ldots, x_n, y_1, \ldots, y_n) = \left(\max_{i \in e}\{x_i\} - \min_{i \in e}\{x_i\}\right) + \left(\max_{i \in e}\{y_i\} - \min_{i \in e}\{y_i\}\right)$$

Overlapping Area as Analytical Function

- First, define:

$$\Theta([L1, R1], [L2, R2]) \\ = [\min(R1, R2) - \max(L1, L2)]^{+}$$



Overlapping Area as Analytical Function (cont'd)

- Overlapping area between modules *i* & *j*:

$$\mathrm{Overlap}_{ij}(x_i,\ y_i,\ x_j,\ y_j)$$

$$= \Theta\left(\left[x_i - \frac{\omega_i}{2},\ x_i + \frac{\omega_i}{2}\right], \left[x_j - \frac{\omega_j}{2},\ x_j + \frac{\omega_j}{2}\right]\right)$$

$$\Theta\left(\left[y_i - \frac{h_i}{2},\ y_i + \frac{h_i}{2}\right], \left[y_j - \frac{h_j}{2},\ y_j + \frac{h_j}{2}\right]\right)$$

An Exact (but Impractical) Formulation

Minimize $\sum_{e \in E} c_e \times \mathrm{HPWL}_e(x_1, \ldots, x_n, y_1, \ldots, y_n)$

Subject to $\mathrm{Overlap}_{ij}(x_i, y_i, x_j, y_j) = 0$ for all $i, j \in V$ s.t. $i \neq j$

$$0 \leq x_i - \frac{\omega_i}{2},\ x_i + \frac{\omega_i}{2} \leq W \text{ for all } i \in V$$

$$0 \leq y_i - \frac{h_i}{2},\ y_i + \frac{h_i}{2} \leq H \text{ for all } i \in V$$

Problems of the Exact Formulation

- The functions $Overlap_{ij}(x_i, y_i, x_j, y_j)$ for $i, j \in V$ are highly nonconvex and not differentiable
- The functions $HPWL_e(x_1, \ldots, x_n, y_1, \ldots, y_n)$ for $e \in E$ are not differentiable (although convex)
- There are $O(n^2)$ constraints
  - n is up to (tens of) millions in modern designs

Practical Analytical Placement Algorithms

- Global placement
  - Wirelength is approximated by some differentiable and convex functions
  - Nonoverlapping constraints are usually replaced by some simpler constraints to make the module distribution roughly even
- Legalization is performed to eliminate module overlaps
- Detailed placement is applied to refine the solution with a more accurate wirelength metric