CAD Algorithms

Placement

Mohammad Tehranipoor

ECE Department

---

Placement and Floorplanning

- Layout maps the structural representation of circuit into a physical representation

- Physical representation:
    - Geometric coordinates for all the circuit elements and the wiring that interconnects the elements

- A circuit is represented by:
    - list of circuit elements
    - list of signals (nets) indicating terminals to be connected together

Important Features

- Layout has two important functions:
  - Positioning of components (**Placement**)
  - Interconnecting the components (**Routing**)
- Placement and Routing are interdependent
  - A fully auto place-n-route (PNR) system should be able to perform task simultaneously.
  - Each problem is so complex.
 - At most, iterative approach is practical.

```
place
route
good ──── no
yes
```
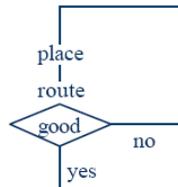
Placement Problem

- Placement is a key step in physical design cycle.
- Definitions:
  - **Position of component** is its physical location and an orientation.
  - **Pins on components** define locations where circuitry within component connects the interconnect between components.
  - Subset of pins that are to be electrically connected form a **net**.

Internal Circuitry

Component

Pins
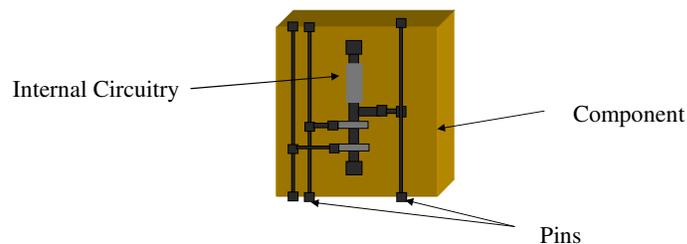
Placement Problem

- Depending on the IC design style Placement has different goals:
    - Placement can affect Router
    - Placement should concentrate on minimizing the **length of interconnection** or **area**
    - **Delays** and **design performance** greatly depend on placement

Input/Output of Placement

- **Input**:
    - Set of fixed blocks
    - Number and location of terminals for each block
    - Netlist
    - Block's dimensions
- **Output**
    - Exact location of blocks
    - Location of pins assigned to each component

Placement at Various Levels

- **System Level:**
  - Place all the PCBs together so that
    - The area occupied is minimum.
    - Heat is dissipated properly.
- **Board Level:**
    - All the chips must be placed within a fixed area of the PCB.
    - All blocks are fixed and rectangular/square in shape.
    - Some chips may be pre-placed
    - Minimize routing layers
    - Heat dissipation
- **Chip Level:**
    - Limited number of layers
    - Minimum area
    - Performance

Objectives

- Normally we are given the maximum chip area
- Placement should define locations such that
  - All elements get connected (routability)
  - Routing area is minimized
  - *Routability* and *Performance* are the measures of good Placement

- Routability:
  - Depends on number of factors such as Router being used.
  - Minimum routing area can *only* be estimated
- Performance:
  - Depends on wirelengths.

Objectives of a Good Placement

- Minimize area (total wiring area)
- Ensure routability
- Avoid signal interference
- Distribute heat
- Maximize performance

Poor Placement

- Consumes large areas
- Results in performance degradation
- Results in difficult and sometimes impossible tasks (Routing)

- An ill-placed layout cannot be improved by high quality routing.

Placement Methodology

Goals: Time, Performance, …	**Placement** →	Minimize Wirelength, Maximize Routability

- Assume d_s = total wirelength estimated for each net

- $\sum\limits_{allnets} d_i = N_d$ = total estimated wirelength of the layout

- **Objective**:

  - *Place such that N_d is to be minimized*

Placement Problem Formulation

- Given an electrical circuit consisting of fixed blocks and a netlist interconnecting terminals on the periphery of these blocks
  - Let B_1, B_2, …,B_n be the blocks to be placed.
  - Each B_i, $1 \leq i \leq n$, has h_i and w_i.
  - Let $N = \{N_1, N_2, …,N_m\}$ be the set of nets.
  - Let $Q = \{Q_1, Q_2, …, Q_k\}$ be the rectangular empty areas allocated for routing between the blocks.
  - Let L_i denote the estimated length of net N_i, $1 \leq i \leq m$.

Cont.

- Construct a layout indicating the positions of each block such that:
    - No two rectangles overlap
    - Placement is routable
    - The total area of bounding box is minimized
    - The total estimated wirelength is minimized
    - In the case of high performance circuits
        - The length of longest net is also minimized. This problem is known as the *performance driven placement problem*.
        - The algorithm must ensure that the known critical paths lengths are shorter than a predefined value.
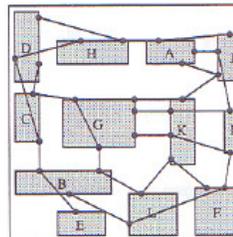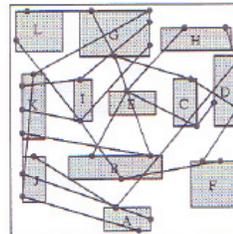
- Placement problem is NP-complete.

Topological Congestion

- The layout area and routability of the layout are approximated by *topological congestion*.

Less Congestion

7
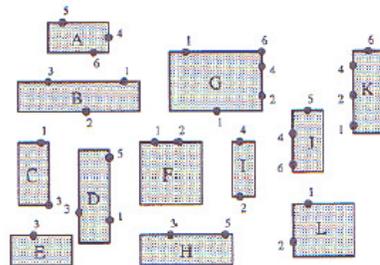
Wirelength

- The actual wiring paths are not known at the time of placement.

- **Minimum spanning tree** representations are the most commonly used structures to connect a net in the placement phase.
  - MST is also used in Global routing phase.

- Optimal wiring paths for a net can be obtained using **Rectilinear Steiner Tree**.
  - RST is used in detailed routing phase.

Design Style Specific Placement

- Different design styles impose different restrictions on the layout and have different objectives in placement problems.
- **Full Custom:**
  - No restriction on how the blocks can be placed.
  - No two blocks overlap.
  - Minimize the total layout area.
  - The irregularity of block shapes is the main cause of unused areas (dead space).
    - Minimize the unused areas.
  - The objective of minimizing the layout area sometimes conflicts with the objective of minimizing the maximum length of a net.
  - In high performance circuit design, additional constraints on net length are considered.

Design Style Specific Placement

- **Standard Cells:**
  - Simpler than the full custom placement problem
    - Cells have same height.
    - Cells are placed in rows.
  - Minimizing area is equivalent to minimizing the summation of channel heights and minimizing the width of the widest row.
  - All rows usually have equal widths.
    - This may not be the case when using large standard cells.
  - Total area:
    - Required area for cells
    - Required area for channels

- With advent of over-the-cell routing, the channels in the standard cells have almost disappeared (*channel-less standard cell designs*).
- Same for metal layers

Design Style Specific Placement

- **Gate Array**:
  - We already have
    - Logic prefabricated
    - Limited routing area
  - Placement tries to find mappings for circuit elements that ensures routability in limited space
    - Minimizing routing area is not as important as ensuring routability by minimizing congestion

Placement Algorithms

- Partitioning-Based Algorithms

- Simulation-Based Algorithms
  - Simulated Annealing
  - Genetic Algorithm

- Performance-Driven Placement Algorithm

Partitioning-Based Algorithms

Breuer's Algorithms:

- The main idea for Breuer's algorithm is to reduce the number of nets being cut when the circuit is partitioned.

- Given
  - $E=\{e_1,e_2,...,e_n\}$ a set of elements

  - $S=\{s_1,s_2,...,s_m\}$ a set of signals (nets)

  - $L=\{l_1,l_2,...,l_p\}$, $p \geq n$ a set of locations

  - Assign each element to unique location such that *something* is minimized.

Min-cut Placement

- Various objective functions have been developed for this method.
  - Total net-cut (Min-cut) objective function
  - Min-max cut value objective function
  - Sequential cut line objective function

Min-cut Objective Function

- Let a layout be divided into smaller blocks using horizontal/vertical lines.
- For each line i, let C_i be the # of wires cut by line i

$$v(i) = \sum_{\text{all } i} C_i$$

- Placement procedures that minimize the value of $v(i)$ are called *Min-cut Placement Algorithms*.

Min-max Cut Value Objective Function

- In case of standard cells and gate arrays, the channel width depends on the number of nets that are routed through the channel.
- The more the number of nets the larger is the channel width.
- The objective function is to reduce the number of nets cut by the cut line across the channel.
    - This will reduce the congestion in channels.

Sequential Cut Line Objective Function

- It is very difficult to compute the minimum net cuts.

- Find cut lines $c_1, c_2, c_3, \ldots c_n$ in sequential manner such that:

    $\textbf{min}(v(c_n))|$ subject to

        $\textbf{min}(v(c_{n-1}))|$ subject to

            .

              .

              $\textbf{min}(v(c_1))$

Placement Procedures

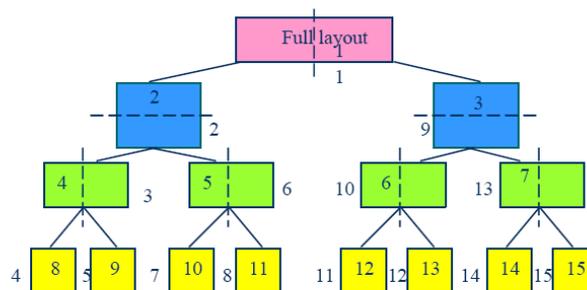- Cut Oriented Min-Cut Placement
- Quadrature Placement
- Bisection Placement
- Slice/Bisection Placement

Min-cut Placement
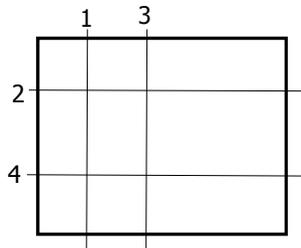
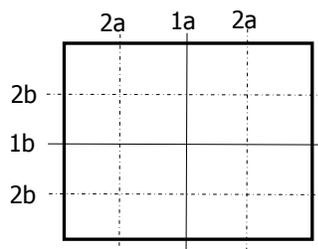- Partitioning is represented as a tree:

Cut Oriented Min-Cut Placement Alg

- This partitioning procedure is sequential and easy to implement.
- It does not provide good results.

Quadrature Placement

- QP is the most popular algorithm.
- Bisect layout by a vertical line $\{B_1, B_2\}$
- Cut two blocks by horizontal line $\{B_1\ B_2\ B_3\ B_4\}$
- Cut 4 blocks by V-line
- Stop when each element is placed
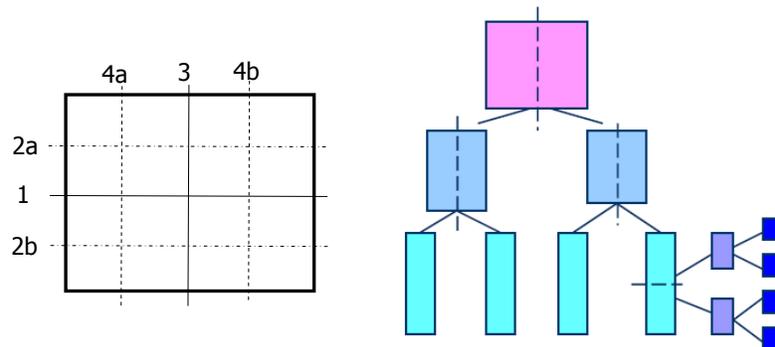
Bisection Placement Procedure

- Divide Block into columns (rows) and then into rows (columns)
- This method is usually used for standard cell placement.

Bisection Placement Procedure

- Not as effective as Quadrature technique
- Good for standard cell layout
- Does not guarantee the minimization of the maximum net cut per channel

Slice Bisection Placement

- Given n elements
- Divide n elements into sets of k and n-k elements
- Repeat procedure to divide remaining n-k elements into sets of size k and n-$2k$
- Continue till all rows (columns) are formed
- Assign locations to elements in each row by vertical (horizontal) bisection.
- This method is most suitable for circuits which have a high degree of interconnection at the periphery.
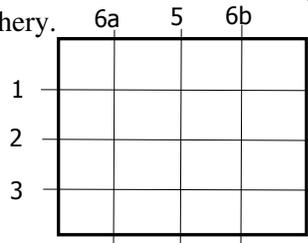
Group migration

- Group migration method can be used in the partitioning process to minimize the cut size.

Simulated Annealing

Simulated Annealing Based Placement

```
begin:
     T = T_start
     placement = init_placement;
     while (T > T_end) do
       {
           while (! equilibrium) do
             {
                     perturb;
                     delta = new_cost – cost;
                     if (delta < 0) accept;
                     else if (random (0, 1) < e^(-delta/T)) accept;
                     else reject ;
             }
           T = cool_down (T);
       }
end
```

TimberWolf 3.2

- TimberWolf is a standard cell placement algorithm based on Simulated Annealing.
- TimberWolf is one of the most successful placement algorithm.
- In this algorithm, the parameters and functions are taken as follows:
- Fixed schedule
    - T_{start} = 4,000,000   // Initial temperature
    - T_{end} = 0.1   // Final temperature
    - Cool_down = alpha (T) x T
        - alpha (T) = 0.8 → 0.95 → 0.8

TimberWolf 3.2 (Cont.)

- Perturb
    - Displace a block to a new location
    - Swap two blocks
    - Change orientation of a block

- Cost function
    Cost = α × wire length + β × block overlap + γ × row length overshoot

Automatic Schedule

- Quality of placement depends on:
    - Perturb
    - Cost function
    - Cooling schedule

- Runtime of SA placement depends on:
    - Annealing (cooling) schedule

- Quality Vs Runtime:
    - Non-linear/complex relationship

Perturb

- Displace the current solution as little as possible
    - Swaps, moves and reorients are widely used
    - Affecting too many blocks will not work

- Temperature dependant Range limiters
    - At High T, use the entire device
    - As T lowers, use a smaller window of the device for swap/move targets

Tweak Zones

- Cost function
  - Components, weights
- Perturb
  - Range limiters, moves/swaps etc
- Equilibrium
  - T_{end}

Genetic Algorithm

Genetic Algorithm

- Initial set of placement configurations is called Population, which can be generated randomly.
- The individuals in the population represent a feasible placement.

- SCORE is same as fitness.
- It can be wirelength.
- Three operations:
    - Crossover
    - Mutation
    - Selection

4 November 2008   41

Genetic Algorithm

begin
    no_pop = SIZE-POP;
    no_offspring = *no_pop* X *k*;
    pop = CONSTRUCT-POP(*no-pop*);
    for (*i* = 1 to *no_pop*) **do**
        SCORE(*pop(i)*)
    for (*i* = 1 to *no_generation*) **do**
        for (*j* = 1 to *no_offspring*) **do**
            (*x,y*) = CHOOSE-PARENT(*pop*)
            offspring(*j*) = GENERATE(*x,y*);
            SCORE(*offspring(j)*));
        pop = SELECT(*pop*, *offspring*, *no_pop*);
        for (*j* = 1 to *no_pop*) **do**
            MUTATE(*pop(j)*);
Return highest scoring configuration in population
end.

4 November 2008   42

21

Performance Driven Placement

- The delay at chip level plays an important role in determining performance of the chip.

- As the number of blocks in a chip increases, the size of the chip decreases but may result in longer interconnections between the blocks (performance degradation).

- The placement algorithms for high performance chips have to generate placements which will allow routers to route nets within the timing requirements.

- Also called *path-based approach*.

- The critical path length must be within its timing constraint.