CS286.5: EDA

Day 12: Tuesday, February 15

Routing (intro+channel)

Today

- Routing Styles
- Channels and Switchboxes
- Channel Routing
  - Vertical Constraint Graphs
  - Unconstrained Left-Edge
  - Constrained Left-Edge
  - Dogleg

Routing Styles

- Gate Array
- Std. Cell-based (row-based)
- Std. Cell-based (row-based), multilevel
- Macrocell

Channels and Switchboxes

- Channels – pins on two sides; minimize tracks
  - assign each channel to single track
- Switchbox – pins on four sides; routable? (minimal expansion?)
  - junction of switchboxes

Trivial Channel Routing

Assign every wire to a horizontal track.

Width of channel =
number of unique signals.

Compare:
channel lower bound=max channel density

(doesn't even work...vertical constraints)

Vertical Constraints

For vertically aligned pins:

- With single vertical routing layer
- Cannot have distinct top pins on "lower" tracks than bottom pins (overlap vertical)
- Produces constraint that top wire be "higher" track than lower
- Combine across all top/bottom pairs, get constraint graph
- (may be cyclical)

Special Case: No Vertical Constraints

- No Vertical Constraints
  - Singled Sided Channel
  - VHV routing (two layers for vertical connections)
- Can be solved optimally
  (tracks = maximum channel density)
- With vertical constraints:
  channel routing NP-complete

Unconstrained Left-Edge Algorithm

1. Sort nets on leftmost end position
2. Start next lowest track
3. Select net with lowest left position
4. Place on current track
    - update end position on current track
    - delete net from list
5. While there are nets which start after last position used
    - place net on track and update end position
    - delete net from list
6. if there are still nets left, return to 2

Example: Left-Edge

Top	0	1	6	1	2	3	5
Bottom	6	3	5	4	0	2	4

Constrained Left-Edge Algorithm

1. Construct VCG
2. Sort nets on leftmost end position
3. Start next lowest track
4. Select net with lowest left position
5. If net has no decendents in VCG:
   - Place on current track
   - update end position on current track
   - delete net from list and VCG
6. While there are nets which start after last position used *and* have no descendents in VCG
   - place net on track and update end position
   - delete net from list
7. if there are still nets left, return to 2

Cycles in VCG?

Top	1	1	2
Bottom	2	0	1

- Cannot find any channel ordering to satisfy
- Must relax *artificial* constraint of single horizontal track per signal
- *Dogleg* – splitting of horizontal runs into multiple track segments
- in general, can reduce track requirements

Top	1	1	2	2	0
Bottom	0	2	0	3	3

Dogleg Algorithm

- Break net into segments at pin positions
- Build VCG based on segments
- run unconstrained on segments rather than full wires

Top	1	1	2	0	2	3
Bottom	2	3	0	3	4	4

Exploit Freedom

Goal:

- Reduce channel density
- Eliminate VCG cycles
- Reduce/eliminiate veritical constraints

Freedom:

- Swap equivalent pins
- Mirror cell (?)
- Choose instance of cell (permute pins)

More Metal Layers

- VHV – can achieve channel density (eliminate vertical constriants)
- HVH – theoretical minimum of max
  - half channel density
  - height of vertical constraint graph
- OTC – w/ one layer
  - planar route nets over cell
  - maximum independent set (special circle graph, solveable in poly time)

Other Routing Problems

- Steiner Trees
- Maze Routing
- Switchboxes

Summary

- Routing Models
- Channels
- Left-Edge (optimal, greedy algorithm...for restricted case)
- Relax single track constraint (doglegs)
- More wires (flavor)

Big Ideas

- Decompose problem (divide-and-conquer)
- Interrelation of components (*e.g.* pin positions)
- Structure: special case, solve optimally
- Technique: greedy algorithm
- Use as basis for more general algorithm