Today

- Custom/Semi-custom Routing
- Slicing
- Channel Routing
- Over-the-Cell/Multilayer

Routing Problem

- Where to wires run?
- Once know where blocks live,
  - where do the wires go?
    - In such a way as to:
      - Fit in fixed resources
      - Minimize resource requirements
        - (channel width $\rightarrow$ area)

Variants

- Gate-Array
- Standard-Cell
- Full Custom

Gate Array

- Fixed Grid
- Fixed row and column width
- Must fit into prefab channel capacity

Gate Array

- Opportunities
  - Choice in paths
  - How exploit freedom to:
    - Meet channel limits
    - Minimize channel width
Gate Array

- Opportunities
  - Choice in paths
  - How exploit freedom to:
    - Meet channel limits
    - Minimize channel width

Semicustom Array

- Float Channel widths as needed
- Becomes a questions of minimizing total channel widths

Row-based Standard Cell

- Variable size
  - Cells
  - Channels
- Primary route within row
- Vertical feed throughs

Standard Cell Gates

- IOs on one or both sides
- Design in Feed-thru

Full Custom / Macroblock

- Allow arbitrary geometry
  - Place larger cells
    - E.g. memory
  - Datapath blocks

Channel Routing

- Key subproblem in all variants
- Psuedo 1D problem
- Given: set of terminals on one or both sides of channel
- Assign to tracks to minimize channel width
Gate Array → Channel
• Global route first
  – Decide which path each signal takes
  – Sequence of channels
  – Minimize congestion
    • Wires per channel segment

Gate Array → Channel
• Then Channel route each resulting channel

Std.Cell → Channel Route
• Plan feed through
• Channel route each row

Macroblock → Channel Route
• Slice into pieces
• Route each as channel
• Work inside out
• Expand channels as needed
• Complete in one pass

Not all Assemblies Sliceable
• No horizontal or vertical slice will separate
• Prevents ordering so can route in one pass

Switchbox Routing
• Box with 3 or 4 sides fixed
• Try to route signals with
Switchbox Route
- Terminals on 4 sides
- Link up terminal

Phased Routing
- After placement…
- Slice (macroblock case)
  - And order channels
- Global Route
  - Which channels to use
- Channel Route
- Switchbox Route

Channel Routing

Trivial Channel Routing
- Assign every net its own track
  - Channel width > N (single output functions)
  - Chip bisection $\propto N \rightarrow \text{chip area } N^2$

Sharing Tracks
- Want to Minimize tracks used
- Trick is to share tracks

Not that Easy
- With Two sides
  - Even assigning one track/signal may not be enough
Not that Easy

• With Two sides
  – Even assigning one track/signal may not be enough

Bad assignment
Overlap:
A, B
B, C

Not that Easy

• With Two sides
  – Even assigning one track/signal may not be enough

Valid assignment avoids overlap

Not that Easy

• With Two sides
  – Even assigning one track/signal may not be enough

i.e. there are vertical constraints on ordering

Vertical Constraints

• For vertically aligned pins:
  – With single “vertical” routing layer
  – Cannot have distinct top pins on a lower track than bottom pins
    • Leads to vertical overlap
  – Produces constraint that top wire be higher track than lower
  – Combine across all top/bottom pairs
    • Leads to a Vertical Constraint Graph (VCG)

Channel Routing Complexity

• With Vertical Constraints
  – Problem becomes NP-complete
• Without Vertical Constraints
  – Can be solved optimally
    – Tracks = maximum channel density
    – Greedy algorithm
No Vertical Constraints

Good for:
• Single-sided channel
  – (no top and bottom pins)
• Three layers for routing
  – Two vertical channels allow top and bottom
    pins to cross
  – May not be best way to use 3 layers...

Left-Edge Algorithm

1. Sort nets on leftmost end position
2. Start next lowest track; end=0
3. While there are unrouted nets with lowest
   left position > end of this track
   – Select unrouted net with lowest left position >
     end
   – Place selected net on this track
   – Update end position on this track to be end
     position of selected net
4. If nets remain, return to step 2

Greedy, optimal.

Example: Left-Edge

• Top: a b g b c d f
• Bottom: g d f e a c e
• Sort Left Edge:
  – a: 1—5
  – b: 2—4
  – c: 5—6
  – d: 2—6
  – e: 4—7
  – f: 3—7
  – g: 1—3

• Track 0: a: 1—5
  – End 0
  – Add a: 1—5

• Track 1:
  – g: 1—3
  – b: 2—4
  – d: 2—6
  – f: 3—7
  – e: 4—7
  – c: 5—6

Note: nets (shown as letters here) show up
as numbers in conv. channel
routing file formats.
Example: Left-Edge

- Top: a b g b c d f
- Bottom: g d f e a c e

Sort Left Edge:
- Track 0: a: 1—5
- Track 1: g: 1—3, e: 4—7
- d: 2—6
- b: 4
- f: 3—7
- c: 5—6

Track 0: a: 1—5
Track 1: g: 1—3, e: 4—7
Track 2: b: 4
Track 3: d: 2—6
Track 4: f: 3—7

Example: Constrained Left-Edge

1. Construct VCG
2. Sort nets on leftmost end position
3. Start new track; end=0
4. While there are nets that have
   - No descendents in VCG
   - And left edge > end
     1. Place net on track and update end
     2. Delete net from list, VCG
5. If there are still nets left to route, return to 2

Example: Constrained Left-Edge

- Track 0: a: 1—5
- Track 1: g: 1—3, e: 4—7
- d: 2—6
- f: 3—7
- c: 5—6
- b: 2—4
Example: Left-Edge

• Top:  \( a b g b c d f \)
• Bottom:  \( g d f e a c e \)
• Nets:
  - a: 1—5
  - b: 2—4
  - c: 5—6
  - d: 2—6
  - e: 4—7
  - f: 3—7
  - g: 1—3

Example 2: ...

• Top:  \( a a a b d e g c \)
• Bottom:  \( b c d e f g f f \)
• Sort Left Edge:
  - b: 1—4
  - a: 1—3
  - c: 5—6
  - e: 4—5
  - f: 3—7
  - g: 6—7

VCG Cycles

• No channel ordering satisfies VCG
• Must relax artificial constraint of single horizontal track per signal
• Dogleg: split horizontal run into multiple track segments
• In general, can reduce track requirements

Dogleg Cycle Elimination

• Top:  \( 1 1 2 \)
• Bottom:  \( 2 3 1 \)
• VCG:
Dogleg Cycle Elimination

- Top: 1a 1a/1b 2
- Bottom: 2 3 1b
- VCG:

Dogleg Algorithm

1. Break net into segments at pin positions
2. Build VCG based on segments
3. Run constrained on segments rather than full wires

Dogleg Example (no cycle)

- Top: 1 1 2 - 2 3
- Bottom: 2 3 - 3 4 4

No Dogleg

- Top: 1 1 2 - 2 3
- Bottom: 2 3 - 3 4 4

With Dogleg

- Top: 1 1 2a/2b - 2b 3b
- Bottom: 2a 3a - 3a/b 4 4

Other Freedoms

- Swap equivalent pins
  - E.g. nand inputs equivalent
- Mirror cells
  - if allowed electrically
- Choose among cell instances
  - Permute pins

...
Exploit Freedom To

- Reduce channel density
- Reduce/Eliminate vertical constraints
  - Cycles
  - VCG height

Over The Cell

- Limit cell to lower metal
  - Maybe only up to M1
- Can route over with higher metal

Example: OTC

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

Over The Cell

- Compute maximal independent set
  - To find nets can be routed in 1 layer (planar) over cell
  - MIS can be computed in $O(n^2)$ time with dynamic programming
- Then route residual connections in channel
- Works on 2-metal if only M1 in cell
  - Feedthrus in M1

Multilayer

- With 3 layer
  - Can run channel over cells
  - Put Terminals in center of cell

Channel Over Cell
Route Over Cells

- If channel width < cell height
  - Routing completely on top of cells
- If channel width > cell height
  - Cell area completely hidden under routing channel
  - More typical case
    - Especially for large rows

Summary

- Decompose Routing
- Channel Routing
- Left-Edge
- Vertical Constraints
- Exploiting Freedom
  - Dogleg, pin swapping
- Routing over logic

Admin

- Wednesday: no class
- Friday last lecture
  - EAS course feedback forms
- Read: Pathfinder for Friday
  - online

Big Ideas

- Decompose Problem
  - Divide and conquer
- Interrelation of components
- Structure: special case can solve optimally
- Technique: Greedy algorithm
- Use greedy as starting point for more general algorithm