CS294-7: Reconfigurable Computing

Day 17: Tuesday, March 18

Interconnect

- Admin
- Routing

Admin

- **Project 1**

  `http://www.cs.berkeley.edu/~amd/CS294S97/project1.html`

- **Project 2**
  - Proposals Due Today
  - Review meetings Today and Wednesday

Routing

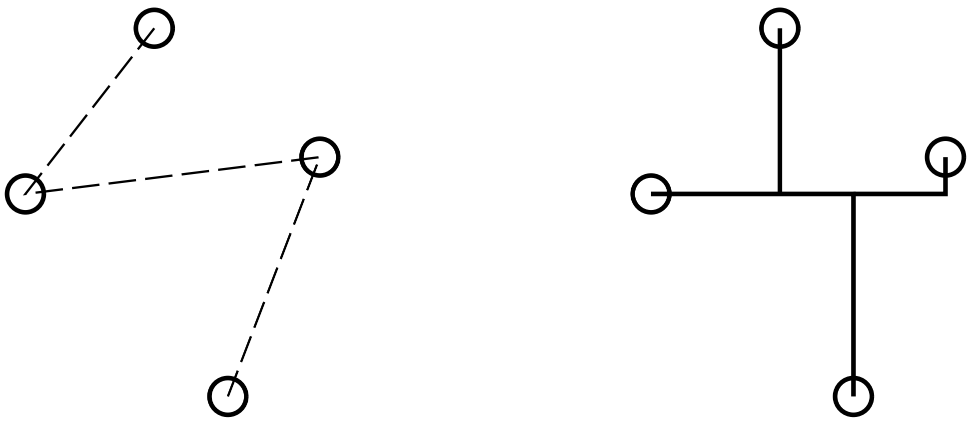
Placement Goals Included:

- minimize total wiring capacity needed (placing interconnected blocks "close" together)
- minimize max channel density (distribute wiring requirements evenly across channels)

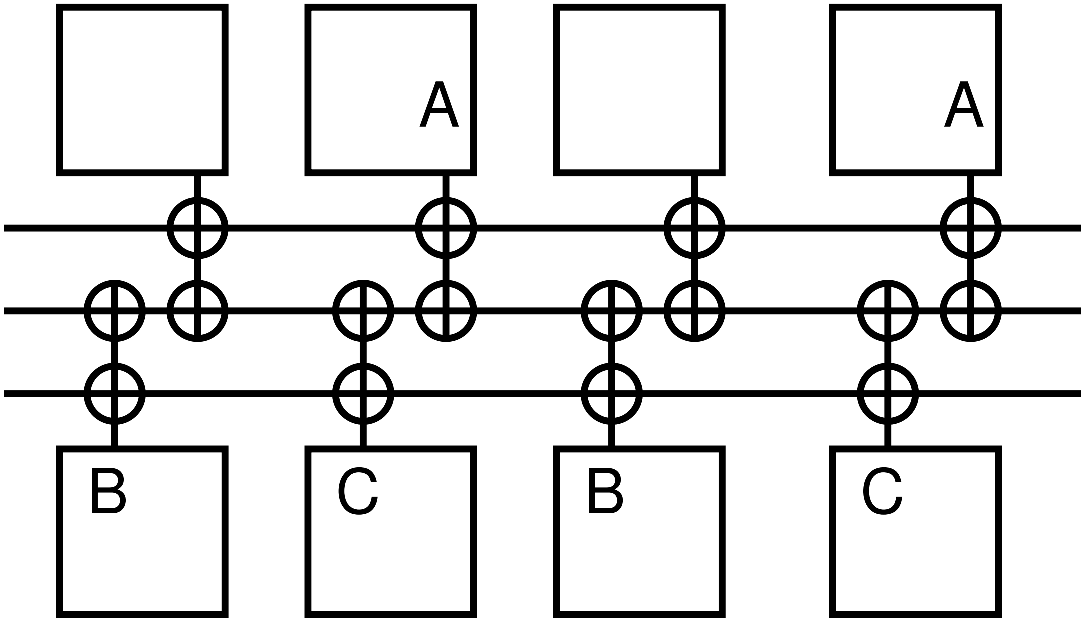
Problem: Allocate wires for nets after placement

Routing Sub-Problems

1. Shortest Path (two-pin nets – $O(N^3_{LUT})$)
2. Steiner Tree
   (easy n-pin $n \leq 5$; NP-complete in general)
3. Compatibility (NP-complete)

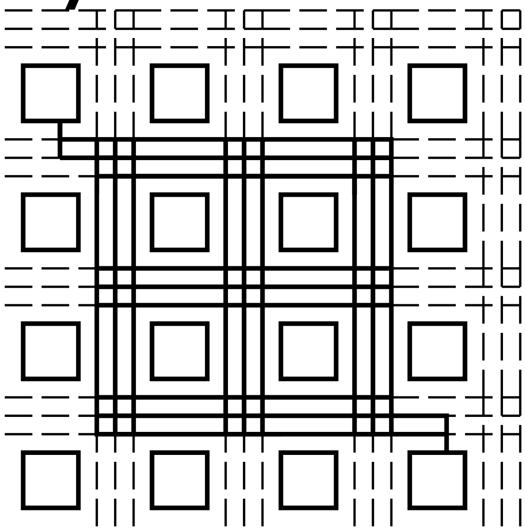


Routing Compatibility



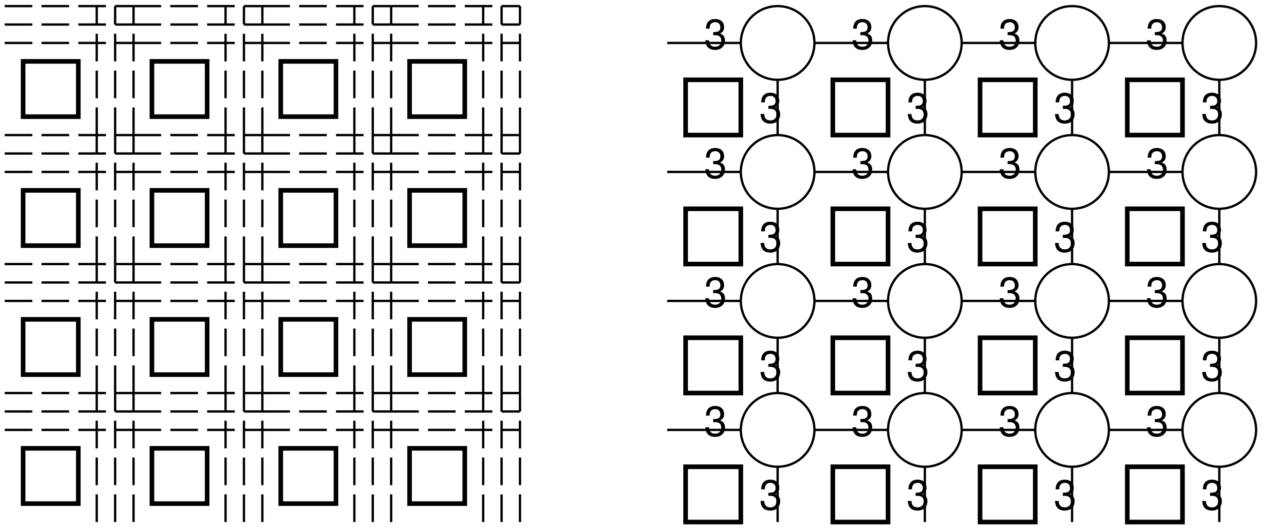
Standard Approach

1. Only (primarily) consider minimum length routes (limit number of bends?)
2. (reduce multipoint nets to sets of two-point nets)
3. iterative refinement for compatibility
   - sequential route and reroute
   - swapping/anneal



4. pragmatic decomposition
   - abstract channel/connection details (global route)
   - detailed connections (detail route)

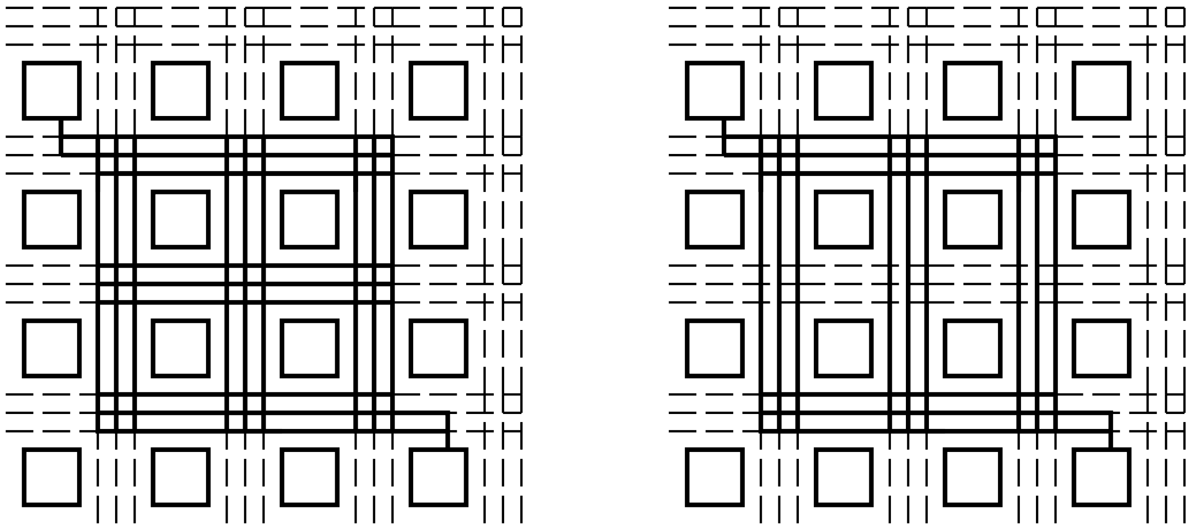
Global Routing



Global Routing – Bends

Why limit bends:

- Standard cells and PC Boards – minimize vias
- Segmented FPGA – maximized potential use of long lines (minimize switches)



Why not: Less routing choices $\rightarrow$ increased compatibility pressure

Global Routing – Iterative

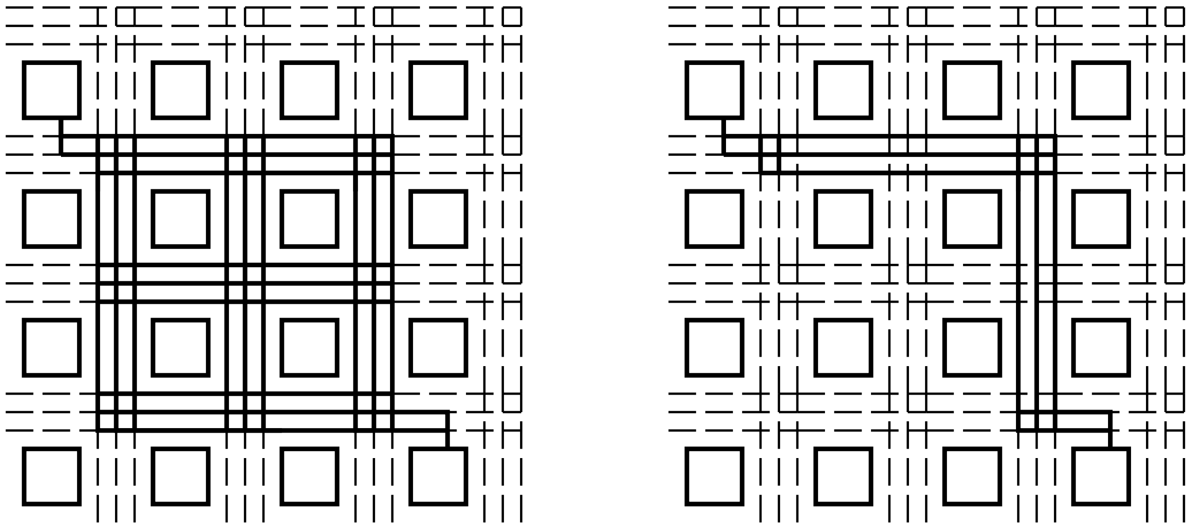
1. **Pick a net (most critical, most congested, ...)**
2. **Unroute net, if necessary**
3. **Route net (allocate a path)**
4. **Repeat 1-3 until all routed**
   (achievable if don't limit channel size
   – some may overflow)
5. **Repeat 1-4 until no improvement**
   (achieve target channel size)
   keep routed nets for successive run (or, at least, information on channel congestion and nets forced to use overflow channels)

Global Routing – SA

- Move: nets (swap net pairs)
- Cost: Channel densities (overflows), delay/length

Detailed Routing – Greedy

1. **Pick a net** (by priority from cost function)
2. **Route net on lowest cost path**
3. **Allocate channel resources, update net costs**
4. **Repeat 1-3 until done**
   (achievable if don't limit channel size – some may overflow)

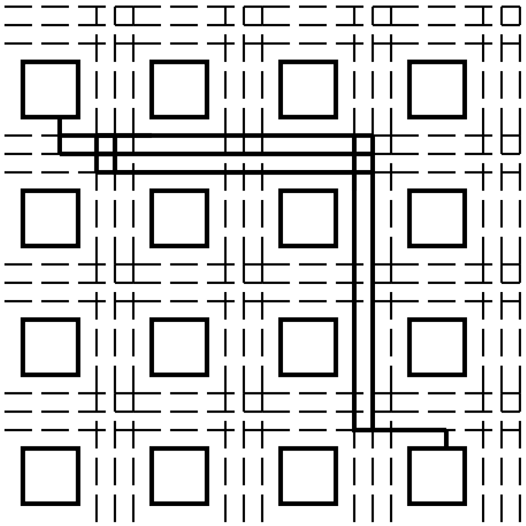


Detailed Routing – Congestion

Demand – weighted sum of potential users of segments
A net with k alternate wire segments, contributes $\frac{1}{k}$ to each wire.

$$\textbf{Demand}(w) = \sum_{w_j \in w} \left(\frac{1}{alt(w_j)} \right)$$

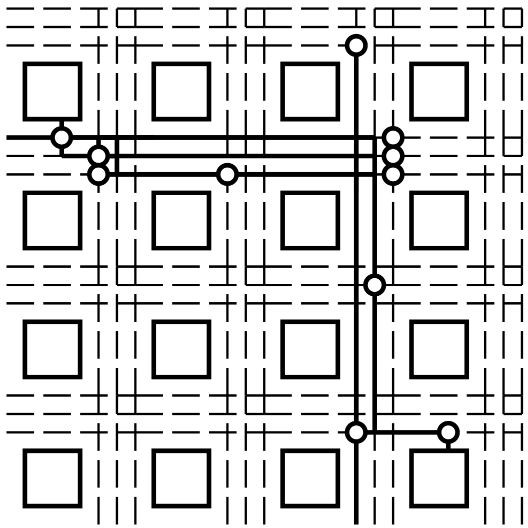
$$\textbf{Demand}(p) = \sum_{w \in p} \left(\textbf{Demand}(w) \right)$$



Detailed Routing – Delay

$$\textbf{Delay}(p) = c_1 \times \textbf{NumSeg}(p) + c_2 \times \textbf{SegLen}(p)$$

- **NumSeg = (actual-minimum)/actual**
- **SegLen = (total wasted)/(total length)**
- c_1, c_2 **– binary**



Routing Delay v/s Cost Function

Cost Function	No Bend Reduce Delay	Bend Reduce Delay
Area	16.7	12.8
Seg. Len.	19.0	16.9
Num. Seg.	14.9	11.9
Len. + Num.	16.4	13.3
RC Delay	14.8	11.9
RC + Nets	13.0	10.1

Assumes adequate number of wires of lengths 1-3.

Delay Routing v/s Area Routing

Fig. 7, Tables 5. and 6 from Brown, Khellah, and Lemieux

Summary

- Hard part of routing – compatibility problem
- Typically attack via iteration or SA
- Complicated by
  - multipoint nets
  - segmented channels
  - area-time tradeoffs with segmentation