

# CS137: Electronic Design Automation

Day 21: November 28, 2005  
Routing 1



CALTECH CS137 Fall2005 -- DeHon

## Today

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

CALTECH CS137 Fall2005 -- DeHon

2

## 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)

CALTECH CS137 Fall2005 -- DeHon

3

## Variants

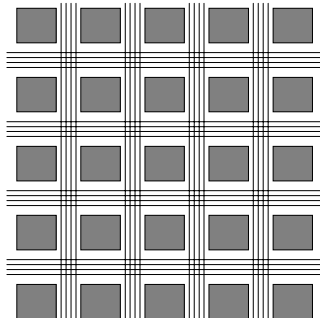
- Gate-Array
- Standard-Cell
- Full Custom

CALTECH CS137 Fall2005 -- DeHon

4

## Gate Array

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

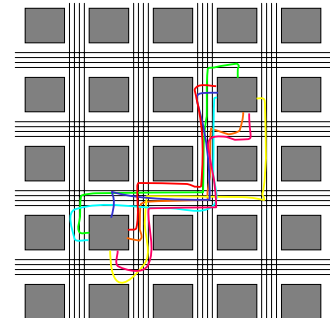


CALTECH CS137 Fall2005 -- DeHon

5

## Gate Array

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

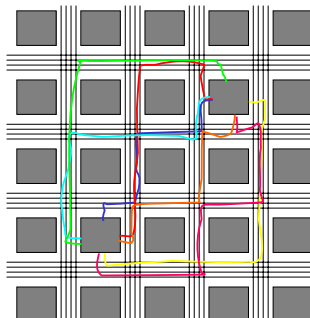


CALTECH CS137 Fall2005 -- DeHon

6

### Gate Array

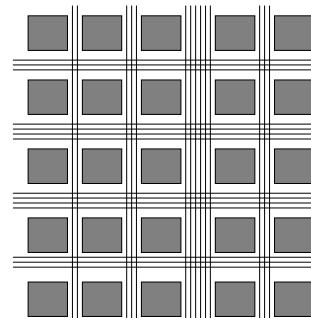
- Opportunities
  - Choice in paths
  - How exploit freedom to:
    - Meet channel limits
    - Minimize channel width



CALTECH CS137 Fall2005 -- DeHon 7

### Semicustom Array

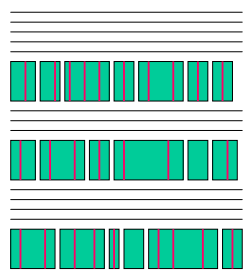
- Float Channel widths as needed
- Becomes a questions of minimizing total channel widths



CALTECH CS137 Fall2005 -- DeHon 8

### Row-based Standard Cell

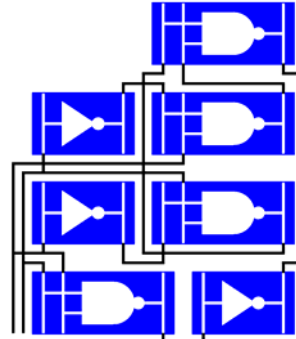
- Variable size
  - Cells
  - Channels
- Primary route within row
- Vertical feed throughs



CALTECH CS137 Fall2005 -- DeHon 9

### Standard Cell Gates

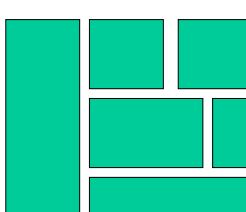
- IOs on one or both sides
- Design in Feed-thru



CALTECH CS137 Fall2005 -- DeHon

### Full Custom / Macroblock


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



CALTECH CS137 Fall2005 -- DeHon 11

### Channel Routing

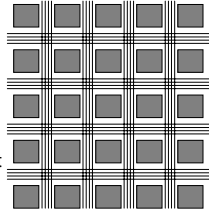
- Key subproblem in all variants
- Psuedo 1D problem
- **Given:** set of terminals on one or both sides of channel
- Assign to tracks to minimize channel width



CALTECH CS137 Fall2005 -- DeHon 12

## Gate Array → Channel

- Global route first
  - Decide which path each signal takes
  - Sequence of channels
  - Minimize congestion
    - Wires per channel segment

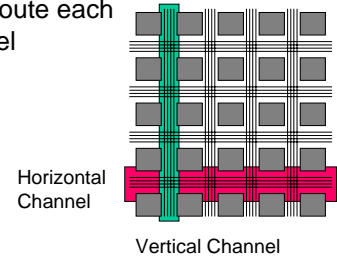


CALTECH CS137 Fall2005 -- DeHon

13

## Gate Array → Channel

- Then Channel route each resulting channel

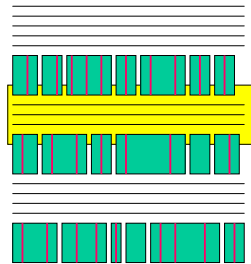


CALTECH CS137 Fall2005 -- DeHon

14

## Std.Cell → Channel Route

- Plan feed through
- Channel route each row

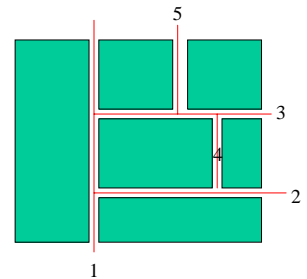


CALTECH CS137 Fall2005 -- DeHon

15

## Macroblock → Channel Route

- **Slice** into pieces
- Route each as channel
- Work inside out
- Expand channels as needed
- Complete in one pass

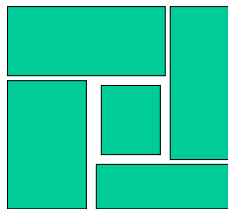


CALTECH CS137 Fall2005 -- DeHon

16

## Not all Assemblies Sliceable

- No horizontal or vertical slice will separate
- Prevents ordering so can route in one pass

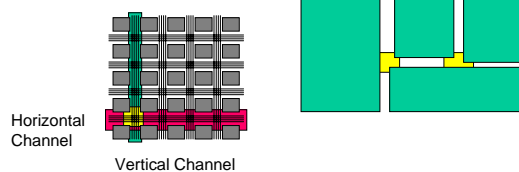


CALTECH CS137 Fall2005 -- DeHon

17

## Switchbox Routing

- Box with 3 or 4 sides fixed
- Try to route signals with

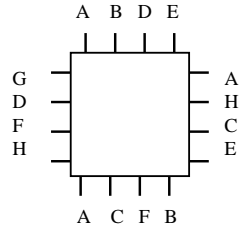


CALTECH CS137 Fall2005 -- DeHon

18

## Switchbox Route

- Terminals on 4 sides
- Link up terminal



CALTECH CS137 Fall2005 -- DeHon

19

## Phased Routing

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

CALTECH CS137 Fall2005 -- DeHon

20

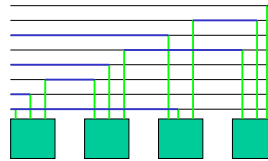
## Channel Routing

CALTECH CS137 Fall2005 -- DeHon

21

## Trivial Channel Routing

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

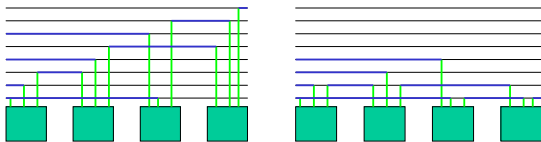


CALTECH CS137 Fall2005 -- DeHon

22

## Sharing Tracks

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

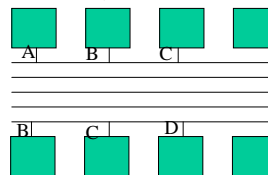


CALTECH CS137 Fall2005 -- DeHon

23

## Not that Easy

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

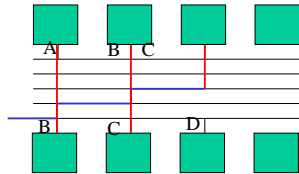


CALTECH CS137 Fall2005 -- DeHon

24

## Not that Easy

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



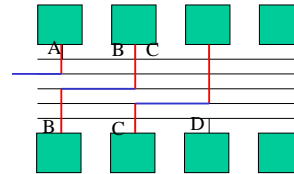
Bad assignment  
Overlap:  
A,B  
B,C

CALTECH CS137 Fall2005 -- DeHon

25

## Not that Easy

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



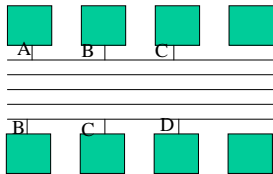
Valid assignment  
avoids  
overlap

CALTECH CS137 Fall2005 -- DeHon

26

## Not that Easy

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



*i.e. there are  
vertical  
constraints  
on ordering*

CALTECH CS137 Fall2005 -- DeHon

27

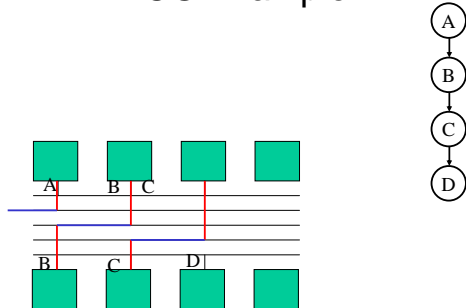
## Vertical Constraints

- 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)

CALTECH CS137 Fall2005 -- DeHon

28

## VCG Example



CALTECH CS137 Fall2005 -- DeHon

29

## Channel Routing Complexity

- With Vertical Constraints
  - Problem becomes NP-complete
- Without Vertical Constraints
  - Can be solved optimally
  - Tracks = maximum channel density
  - Greedy algorithm

CALTECH CS137 Fall2005 -- DeHon

30

## 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...

CALTECH CS137 Fall2005 -- DeHon

31

## 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.

CALTECH CS137 Fall2005 -- DeHon

32

## 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

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

CALTECH CS137 Fall2005 -- DeHon

33

## 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

CALTECH CS137 Fall2005 -- DeHon

34

## 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
    - g:1—3
    - b:2—4
    - d:2—6
    - f:3—7
    - e:4—7
    - c:5—6
  - End 0
  - Add a:1—5
  - End 5

CALTECH CS137 Fall2005 -- DeHon

35

## 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
    - b:2—4
    - d:2—6
    - f:3—7
    - e:4—7
    - c:5—6
  - End 0
  - g:1—3
  - End 3
  - e: 4—7
  - End 7

CALTECH CS137 Fall2005 -- DeHon

36

## Example: Left-Edge

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

CALTECH CS137 Fall2005 -- DeHon

37

## Example: Left-Edge

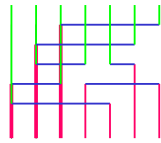
- Top: a b g b c d f
- Bottom: g d f e a c e
  - Track 0: a:1—5
- Sort Left Edge:
  - Track 1: g:1—3, 4:e—7
  - Track 2: b:2—4, c:5—6
  - d:2—6
  - Track 3: d:2—6
  - f:3—7
  - Track 4: f:3—7

CALTECH CS137 Fall2005 -- DeHon

38

## Example: Left-Edge

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



CALTECH CS137 Fall2005 -- DeHon

39

## Constrained Left-Edge

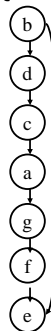
1. Construct VCG
2. Sort nets on leftmost end position
3. Start new track; end=0
4. While there are nets that have
  - ✓ No descendents in VCG
  - ✓ And left edge > end
  - 1. Place net on track and update end
  - 2. Delete net from list, VCG
5. If there are still nets left to route, return to 2

CALTECH CS137 Fall2005 -- DeHon

40

## Example: Constrained 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
- Vertical Constraints
  - a→g
  - b→d
  - g→f
  - b→e
  - c→a
  - d→c
  - f→e

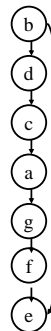


CALTECH CS137 Fall2005 -- DeHon

41

## Example: ...

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

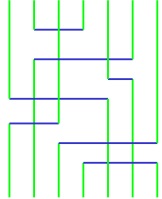


CALTECH CS137 Fall2005 -- DeHon

42

## 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

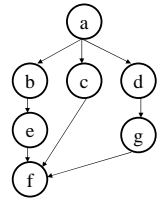


CALTECH CS137 Fall2005 -- DeHon

43

## Example 2: ...

- Top: a a a b e d g c
- Bottom: b c d e f g f f
- Sort Left Edge:
  - b:1-4
  - a:1-3
  - c:2-8
  - d:3-6
  - e:4-5
  - f:5-8
  - c:6-7



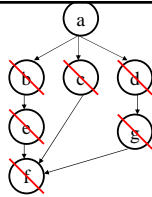
CALTECH CS137 Fall2005 -- DeHon

44

## Example 2: ...

- Top: a a a b e d g c
- Bottom: b c d e f g f f
- Sort Left Edge:
  - ~~- b:1-4~~
  - a:1-3
  - ~~- c:2-8~~
  - ~~- d:3-6~~
  - ~~- e:4-5~~
  - ~~- f:5-8~~
  - ~~- g:6-7~~

Track 0: f  
Track 1: c  
Track 2: e, g  
Track 3: b  
Track 4: d  
Track 5: a

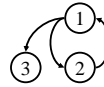


CALTECH CS137 Fall2005 -- DeHon

45

## VCG Cycles

- Top: 1 1 2
- Bottom: 2 3 1
- VCG:

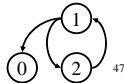


CALTECH CS137 Fall2005 -- DeHon

46

## 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

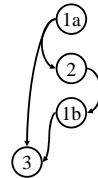
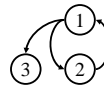


CALTECH CS137 Fall2005 -- DeHon

47

## Dogleg Cycle Elimination

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

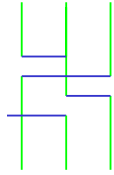
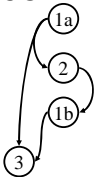


CALTECH CS137 Fall2005 -- DeHon

48

## Dogleg Cycle Elimination

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



CALTECH CS137 Fall2005 -- DeHon

49

## Dogleg Algorithm

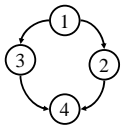
1. Break net into segments at pin positions
2. Build VCG based on segments
3. Run constrained on segments rather than full wires

CALTECH CS137 Fall2005 -- DeHon

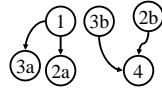
50

## Dogleg Example (no cycle)

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



1 1 2a/2b - 2b 3b  
2a 3a - 3a/b 4 4

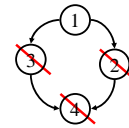


CALTECH CS137 Fall2005 -- DeHon

51

## No Dogleg

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

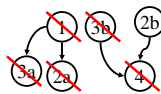
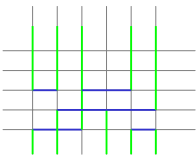


CALTECH CS137 Fall2005 -- DeHon

52

## With Dogleg

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

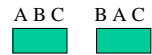


CALTECH CS137 Fall2005 -- DeHon

53

## Other Freedoms

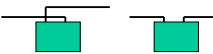
- Swap equivalent pins
  - E.g. nand inputs equivalent
- Mirror cells
  - if allowed electrically
- Choose among cell instances
  - Permute pins

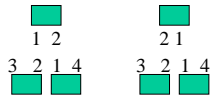


CALTECH CS137 Fall2005 -- DeHon

54

## Exploit Freedom To

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



CALTECH CS137 Fall2005 -- DeHon

55

## Over The Cell

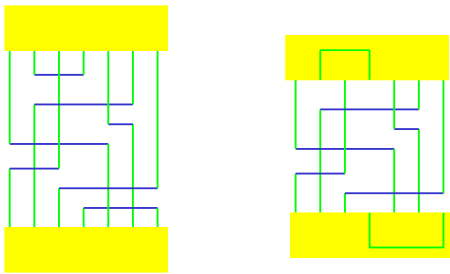
- Limit cell to lower metal
  - Maybe only up to M1
- Can route over with higher metal

CALTECH CS137 Fall2005 -- DeHon

56

## Example: OTC

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

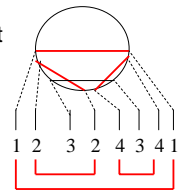


CALTECH CS137 Fall2005 -- DeHon

57

## 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



CALTECH CS137 Fall2005 -- DeHon

58

## Multilayer

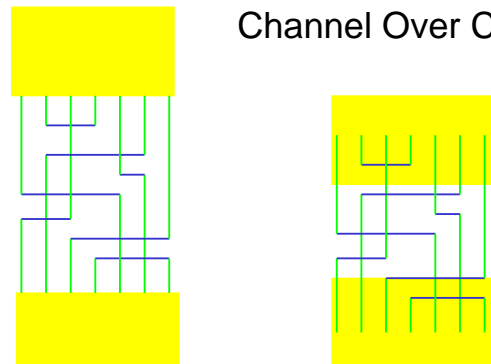
- With 3 layer
  - Can run channel over cells
  - Put Terminals in center of cell



CALTECH CS137 Fall2005 -- DeHon

59

## Channel Over Cell



CALTECH CS137 Fall2005 -- DeHon

60

## 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

CALTECH CS137 Fall2005 -- DeHon

61

## Summary

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

CALTECH CS137 Fall2005 -- DeHon

62

## Admin

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

CALTECH CS137 Fall2005 -- DeHon

63

## 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

CALTECH CS137 Fall2005 -- DeHon

64