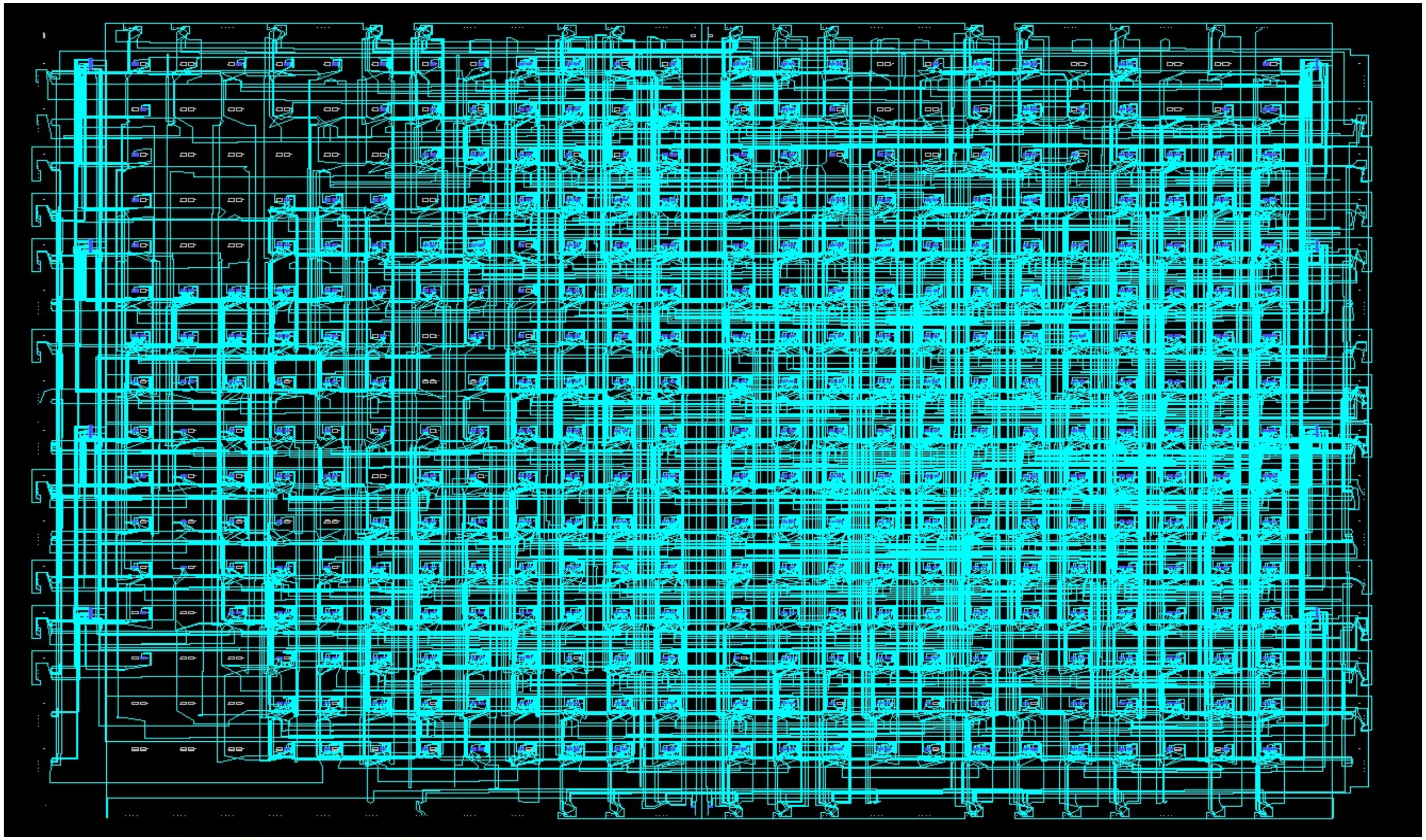
Placement and Routing

A Lecture in CE Freshman Seminar Series:
Puzzling Problems in Computer Engineering



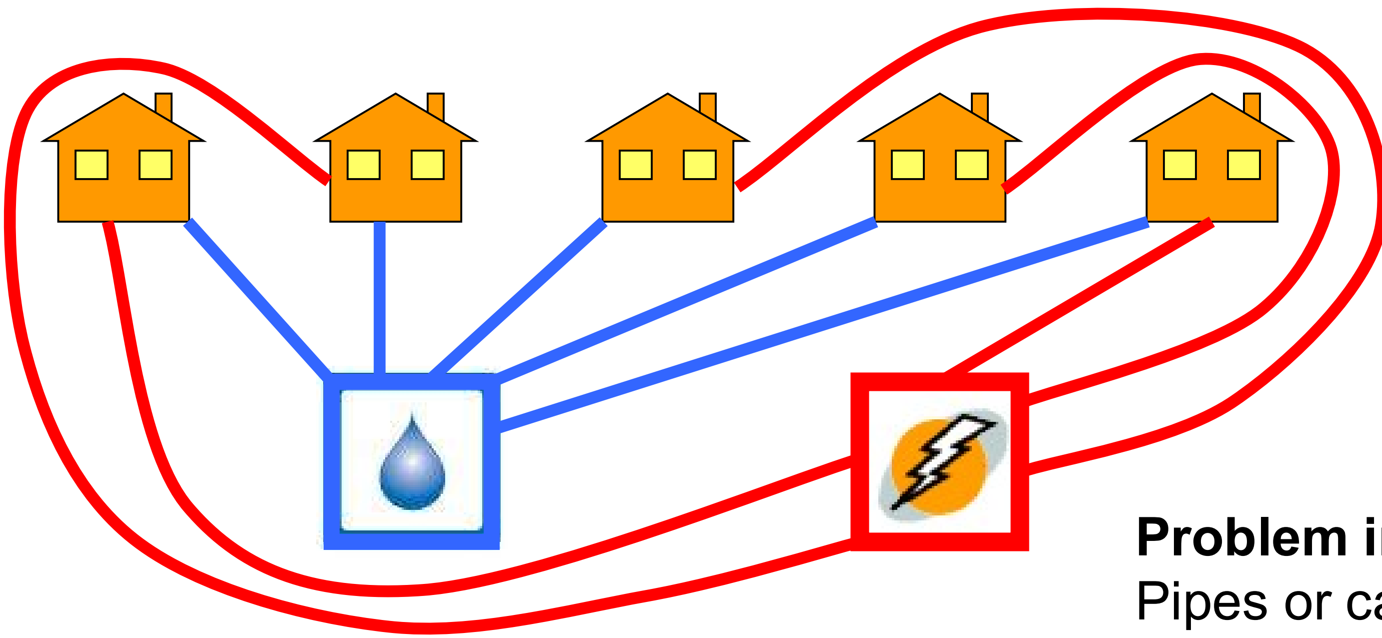
About This Presentation

This presentation belongs to the lecture series entitled "Ten Puzzling Problems in Computer Engineering," devised for a ten-week, one-unit, freshman seminar course by Behrooz Parhami, Professor of Computer Engineering at University of California, Santa Barbara. The material can be used freely in teaching and other educational settings. Unauthorized uses, including any use for financial gain, are prohibited. © Behrooz Parhami

Edition	Released	Revised	Revised	Revised	Revised
First	Apr. 2007	Apr. 2008	Apr. 2009	Apr. 2010	Mar. 2011
		Apr. 2012	Apr. 2015	Apr. 2016	Mar. 2020

Houses and Utilities: Warm-up Version

There are *n* houses on one side of a street and 2 utility companies on the other. Connect each utility facility to every house via lines of any desired shape such that the lines do not intersect.

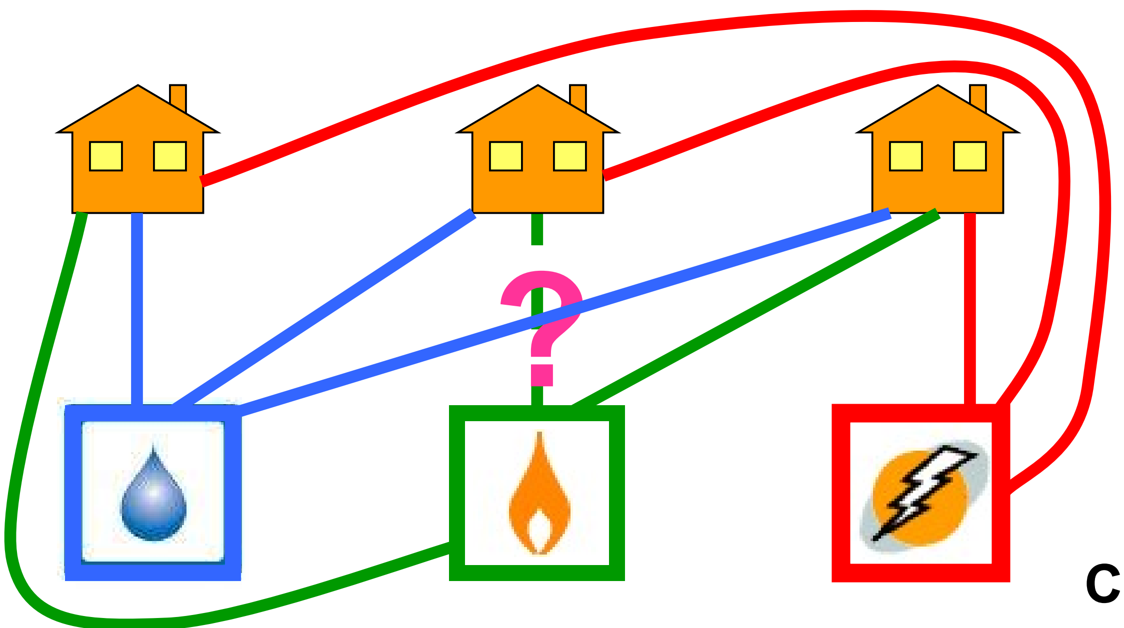


The scheme above works for 2 utilities, with any number *n* of houses

Problem interpretation:
Pipes or cables must be laid in separate trenches (at the same depth)

Houses and Utilities: Classic Version

There are 3 houses on one side of a street and 3 utility companies on the other. Connect each utility facility to every house via lines of any desired shape such that the lines do not intersect.



Answer: A solution is impossible (unless you are allowed to cut through a house), but why?

Challenge: Given *h* houses and *u* utilities, when does the puzzle have a solution?

History and Equivalent Puzzles

“Houses and utilities” has a long history and has appeared in many different forms over the years

Even though many authors characterize the puzzle as “ancient,” the first published version dates back to 1917

A less pleasant, pre-gas/electricity variant:

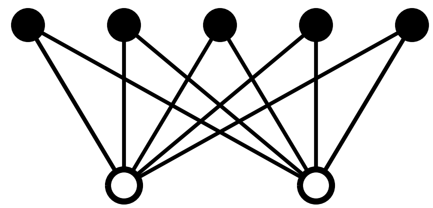
Three people live in adjacent houses next to three wells. Because wells may run dry on occasion, each person needs paths to all wells. After a while, the residents develop strong dislikes for each other and try to construct their paths so that they never have to meet . . .

A violent version:

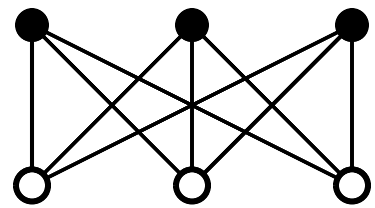
There are three families. Any member of one family will try to kill members of the other families if their paths cross. However, the well, the market, and the church are, by tradition, neutral places . . .

Simplifying the Representation

Complete bipartite graphs:



$K_{2,n}$



$K_{3,3}$

Graphs with white nodes and black nodes in which every white node is connected to every black node, and vice versa

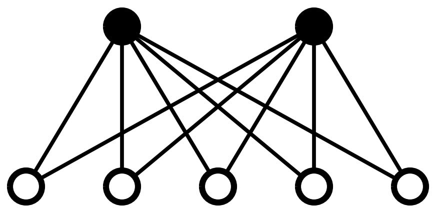
A graph is planar if it can be drawn so that no two edges intersect

Warm-up puzzle: Is $K_{2,n}$ planar for any n? Answer: Yes

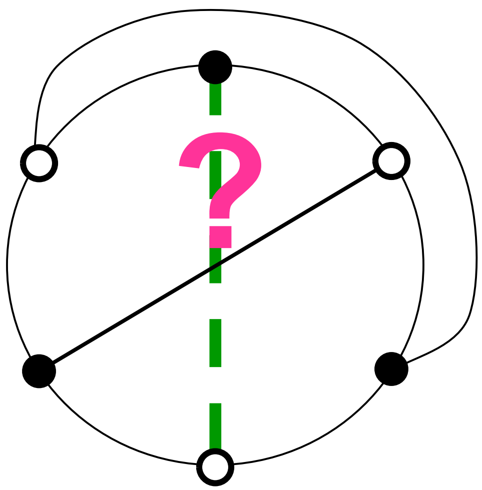
Classic puzzle: Is $K_{3,3}$ planar? Answer: No

Variations on the Puzzle

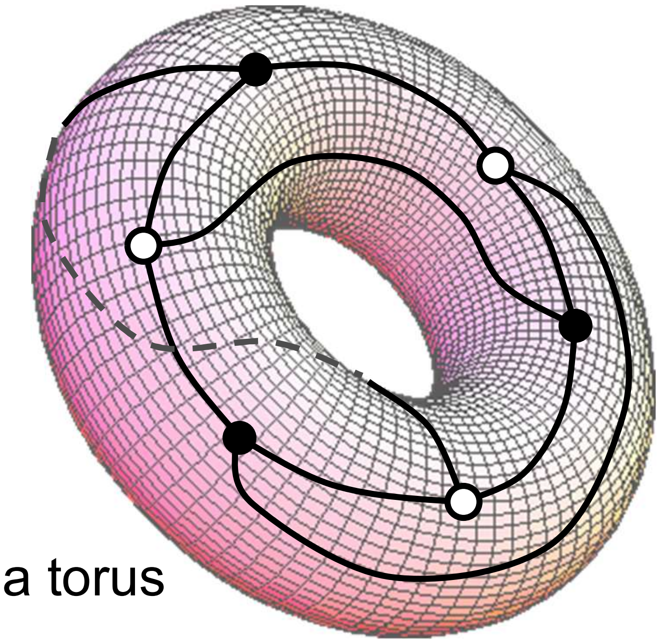
Two houses and *n* utilities



$K_{n,2} = K_{2,n}$

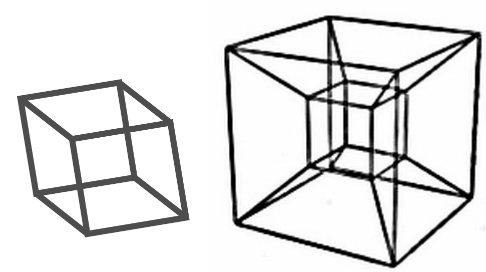


A different drawing of $K_{3,3}$



$K_{3,3}$ on a torus

Challenge questions:
Is the 3D cube graph planar?
What about the 4D cube graph?



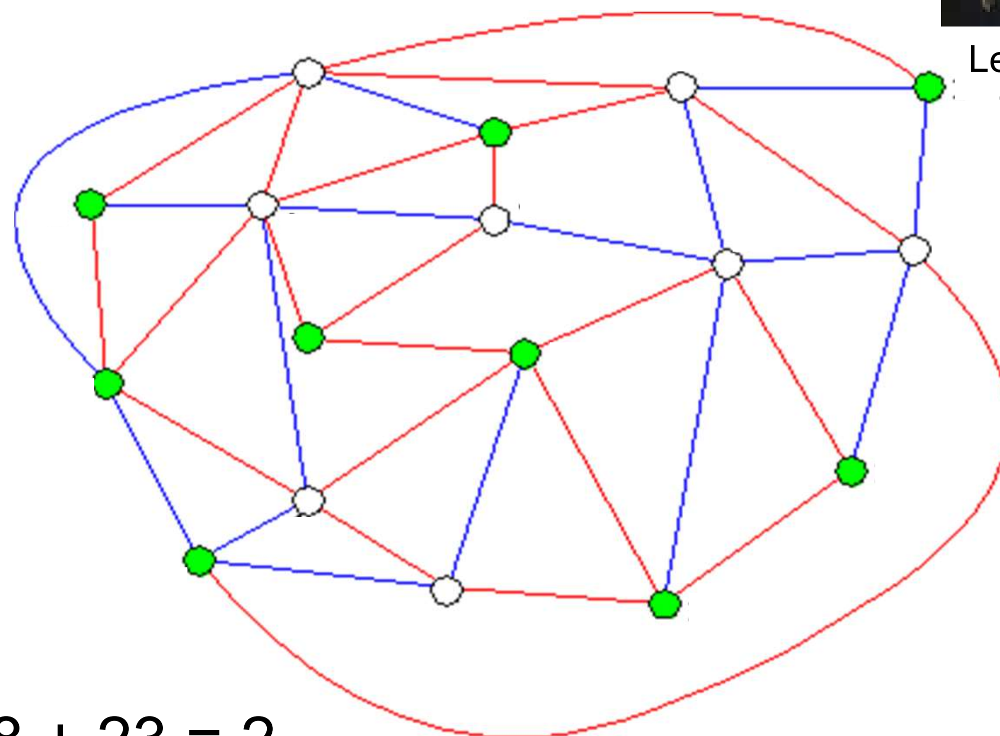
Euler's Formula for Planar Graphs

Leonard Euler
1707-1783

v Number of vertices or nodes
e Number of edges or links
f Number of faces

$v - e + f = 2$

Note that the area outside of the graph counts as a face



Vertex and edge colors do not matter

$v = 17$

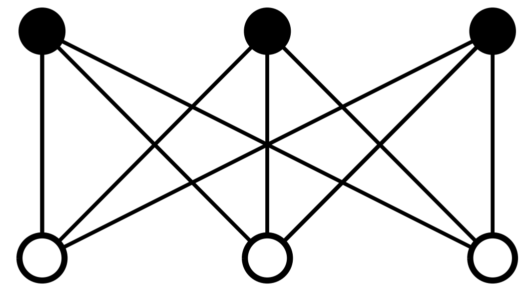
$e = 38$

$f = 23$

$v - e + f = 17 - 38 + 23 = 2$

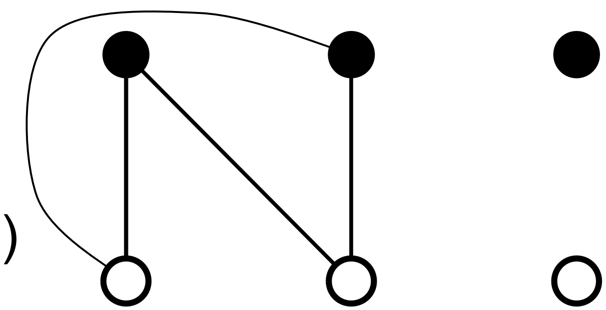
Q1: See if you can provide a proof of Euler's formula $v - e + f = 2$ for planar graphs. It's okay to use Internet sources.

Euler's Formula Tells Us that $K_{3,3}$ Isn't Planar



v Number of vertices or nodes
e Number of edges
f Number of faces

$$v - e + f = 2$$



In a planar bipartite graph, each face has at least 4 sides (edges)

$v = 6$
$e = 9$
$f = ?$

$$6 - 9 + f = 2$$

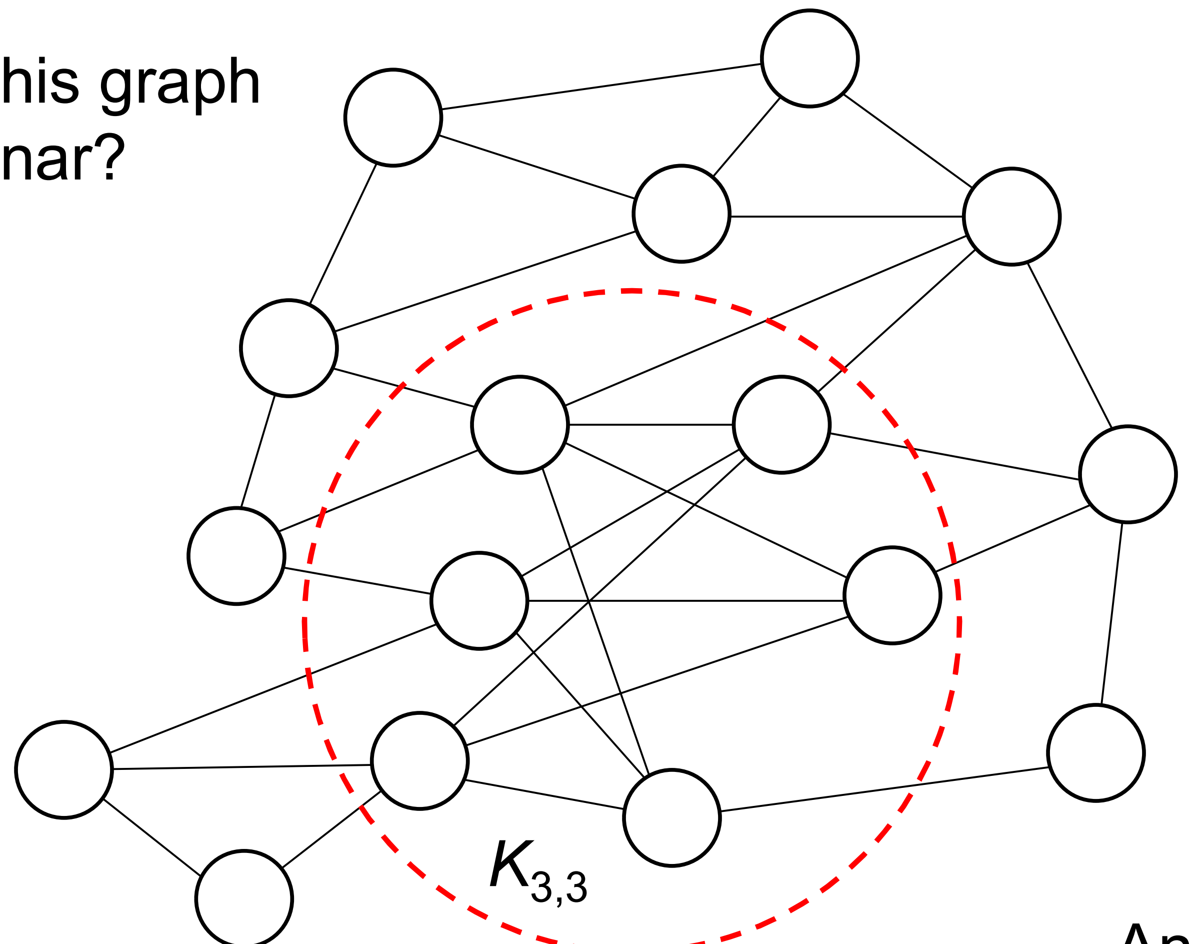
$$f = 5$$

Therefore, to form 5 faces, we need at least $5 \times 4 / 2 = 10$ edges

Division by 2 is due to each edge being part of two different faces

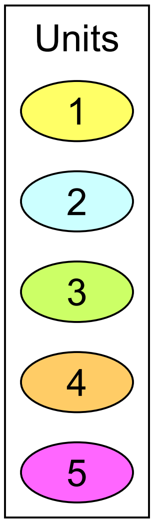
No Graph that Contains $K_{3,3}$ Is Planar

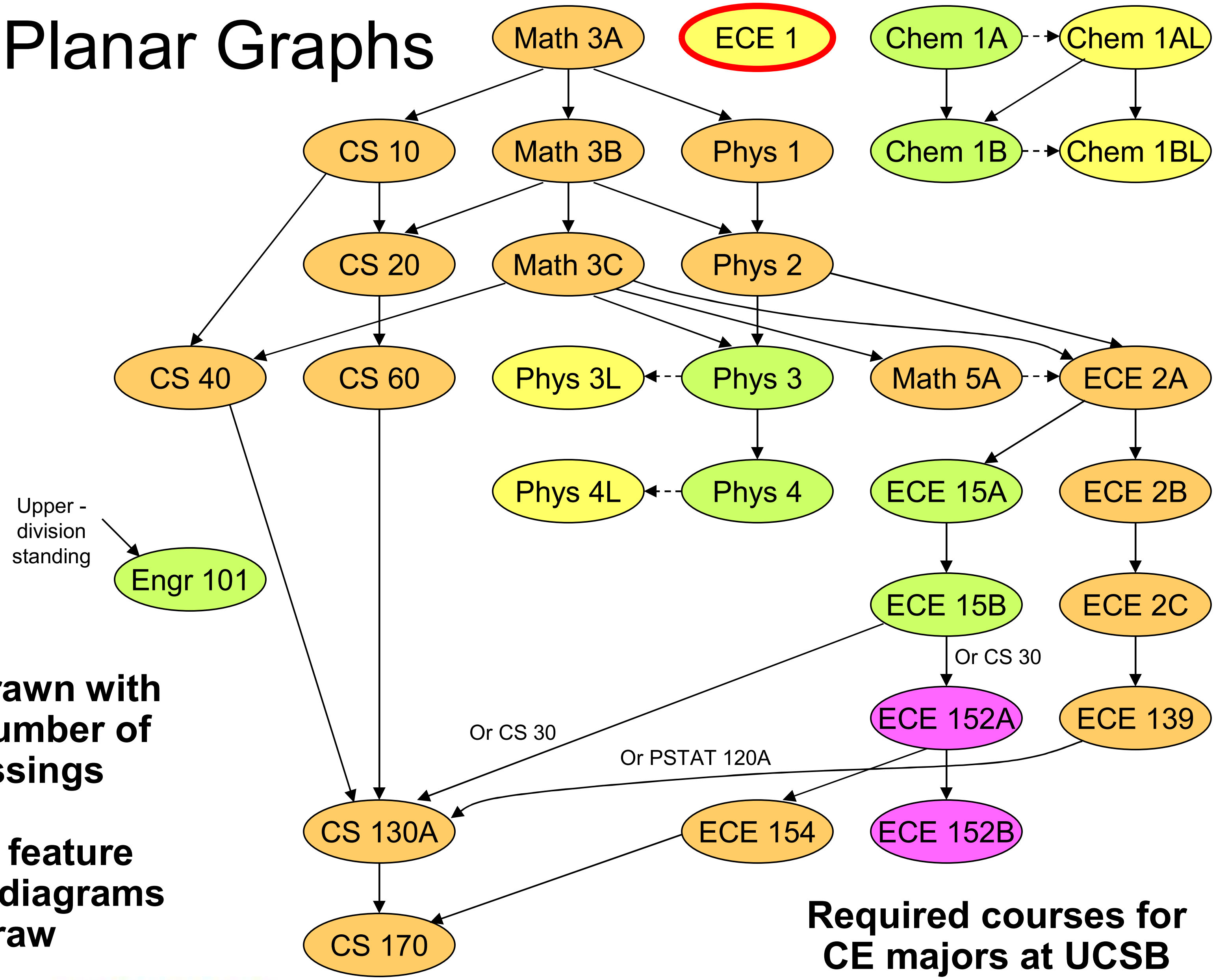
Is this graph planar?



Answer: No

Nearly Planar Graphs





Can be drawn with a small number of edge crossings

Desirable feature for many diagrams that we draw

Required courses for CE majors at UCSB

Rectilinear Paths on a Grid

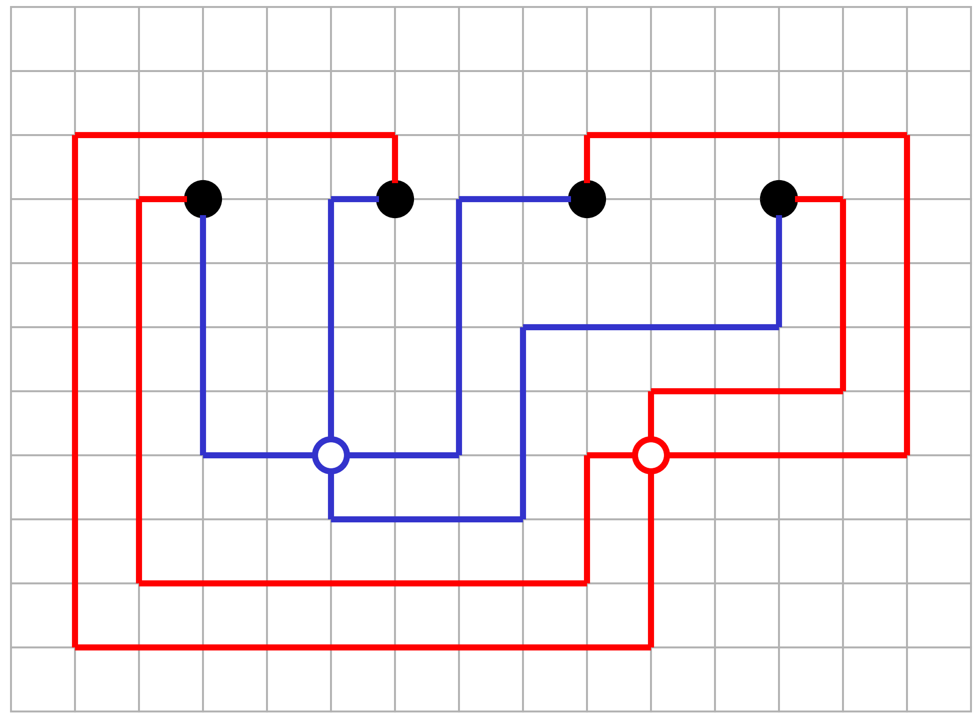
Solve the puzzle with 2 utilities and 4 houses using rectilinear grid paths.

Why rectilinear paths:

Trenches should not be too close to each other

Straight-line trenches with right-angle turns are easier to dig; also easier to locate later

Trenches must be dug along existing streets

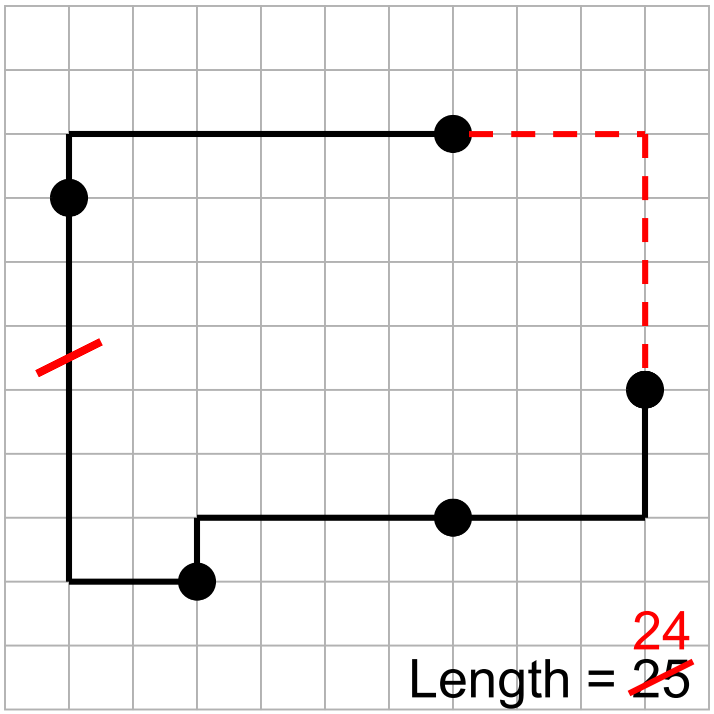


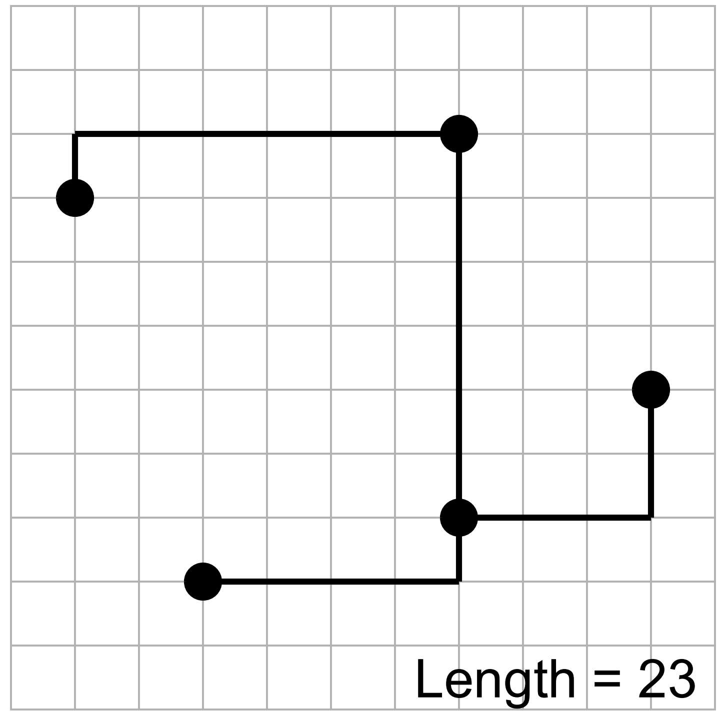
Challenge: Solve the puzzle above with paths that have the minimum possible total length. Now try to solve the puzzle with paths from one utility to all four houses having exactly the same length.

Q2: Argue for or against the claim that if one can solve the houses and utilities puzzle with arbitrary paths, then there is a solution with rectilinear paths.

Spanning Trees

A spanning tree connects a set of nodes in a way that there are no loops (if you remove any tree edge, then nodes are no longer connected)

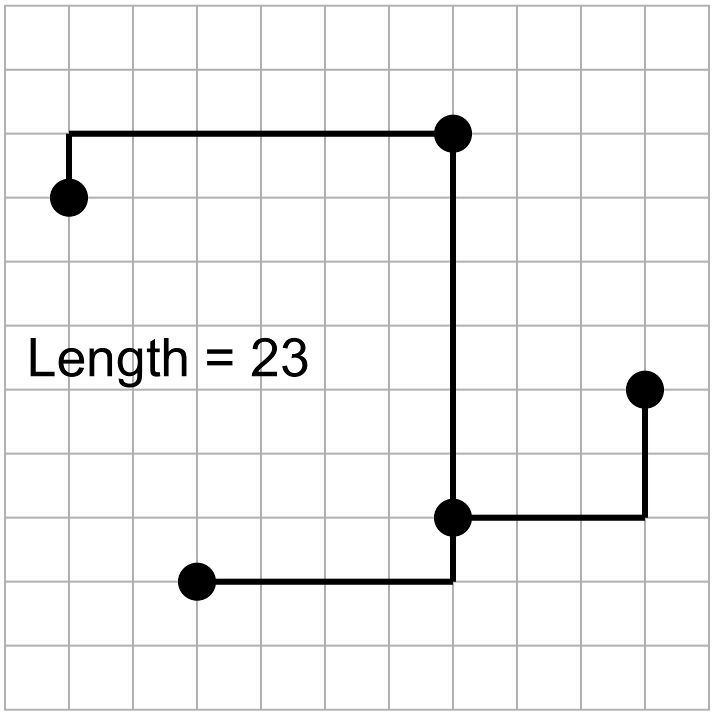




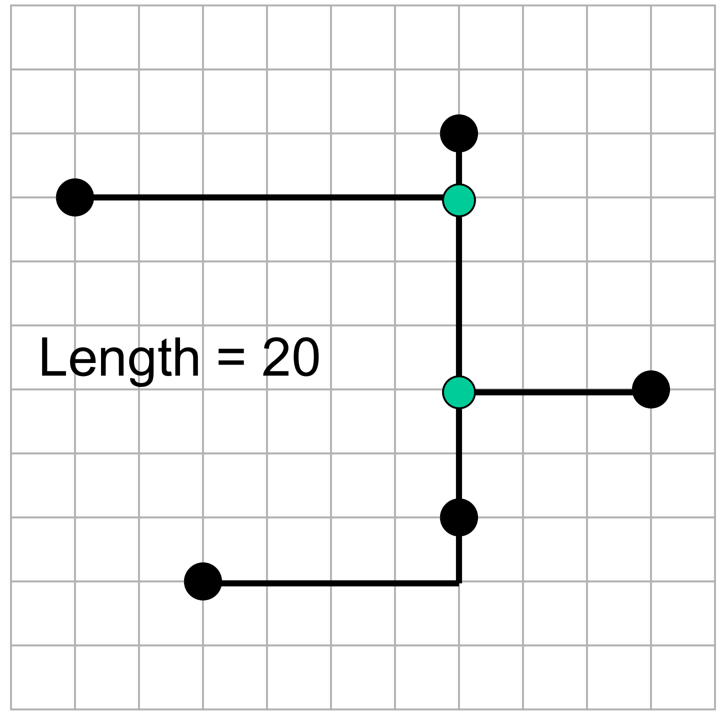
Greedy algorithm for building a minimal spanning tree: Begin by connecting the closest pair of nodes. Then, at each step, connect the partial tree to the node closest to it (closest to one of its nodes)

Steiner Trees

Given *n* grid points, connect them to each other via a rectilinear network such that the total wire length is minimized.



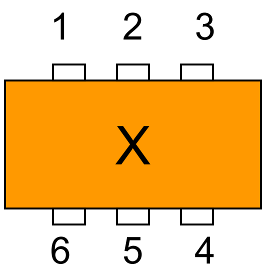
Spanning tree



Steiner tree

Q3: Is the Steiner tree above the best for connecting the five nodes?

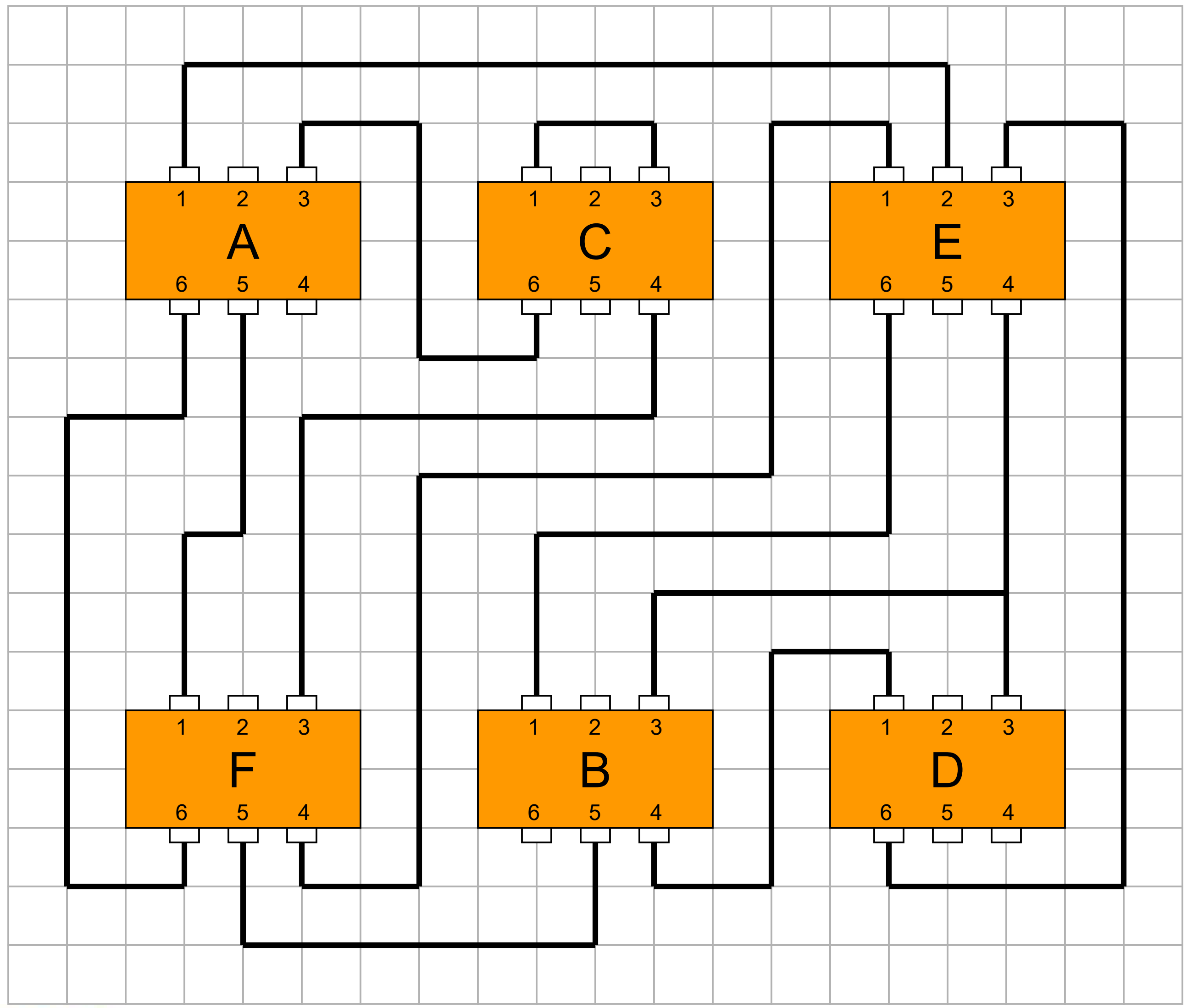
Placement and Routing



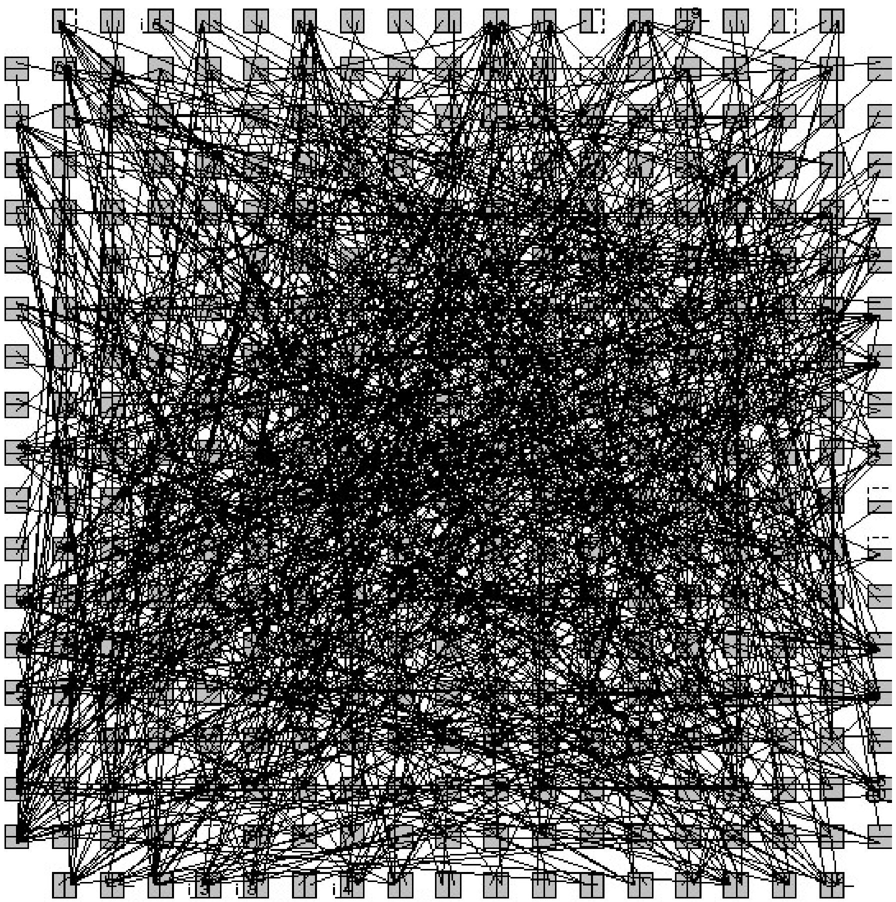
Components:
A,B,C,D,E,F

Net List:

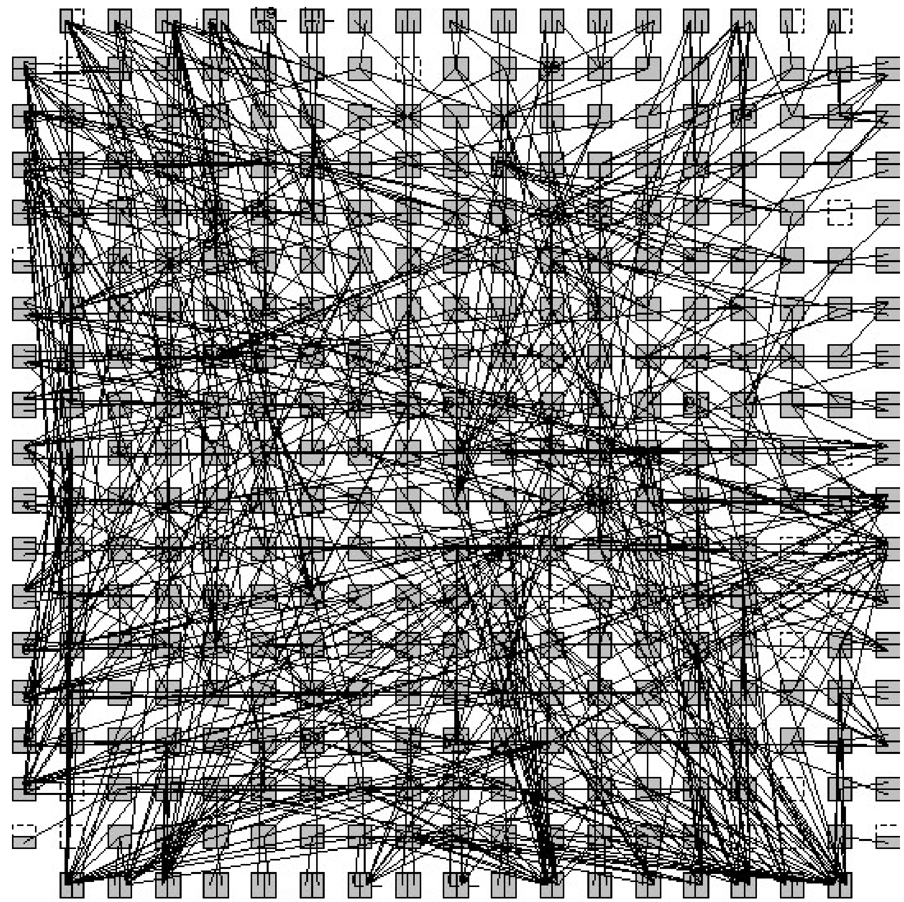
A1, E2
A3, C6
A5, F1
A6, F6
B1, E6
B3, D3, E4
B4, D1
B5, F5
C1, C3
C4, F3
D6, E3
E1, F4



The Importance of Placement



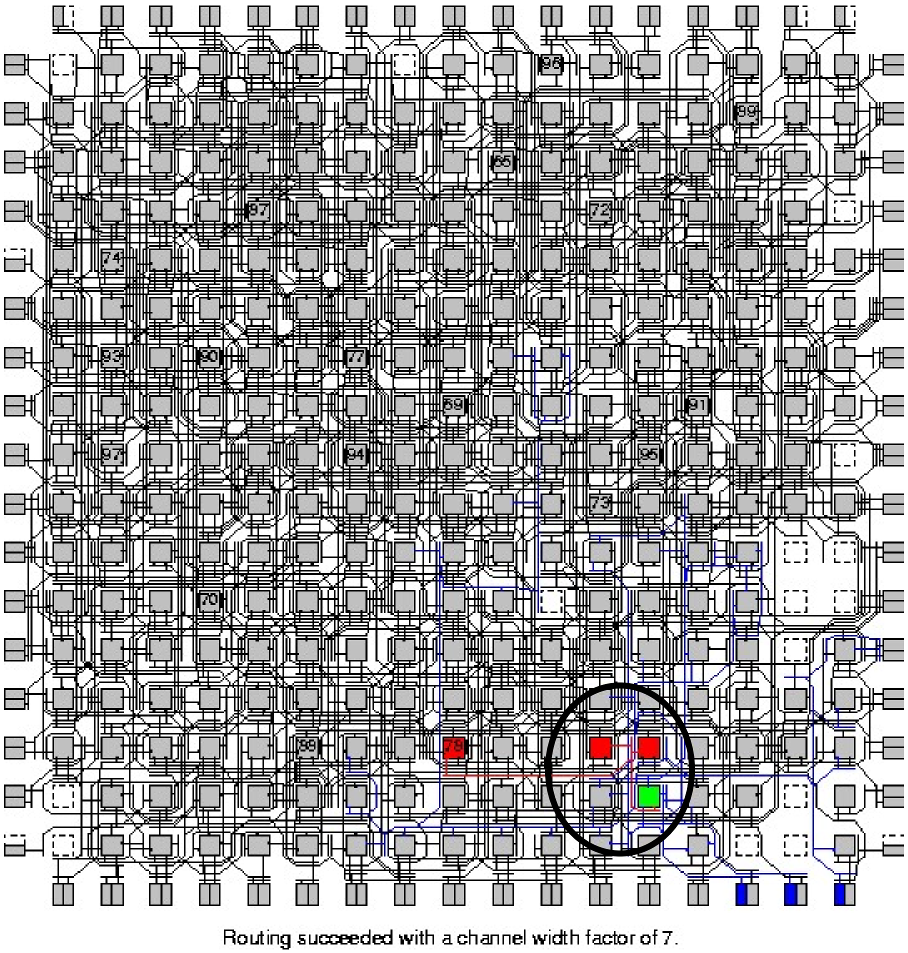
Random



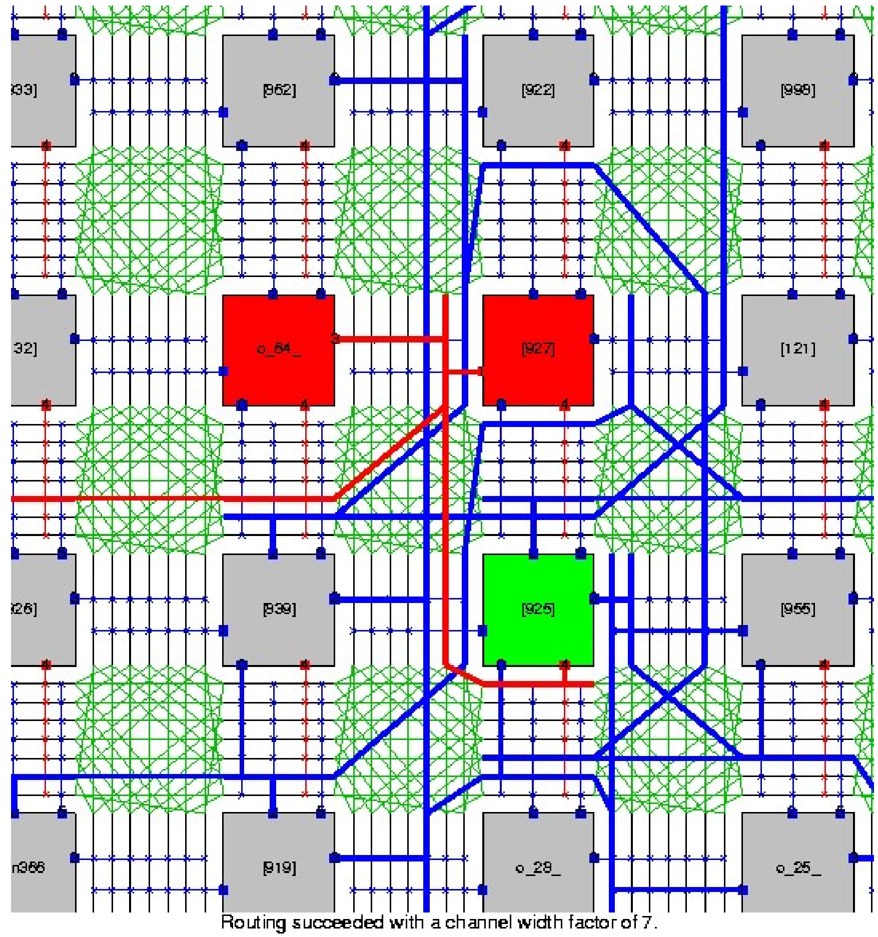
Final

Source: http://www.eecg.toronto.edu/~vaughn/vpr/e64.html

Routing after Placement



Channel routing (width = 7)

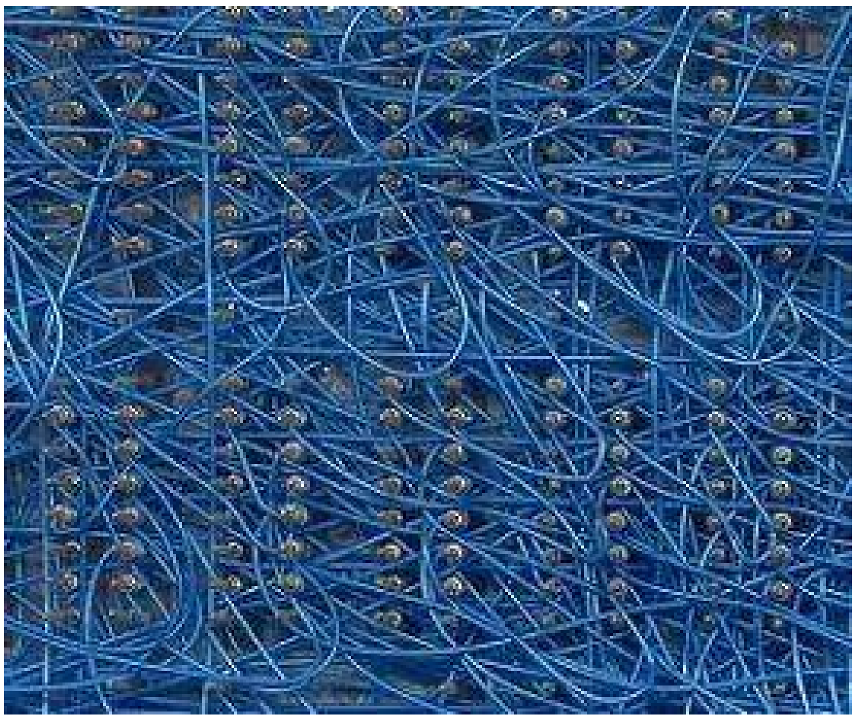


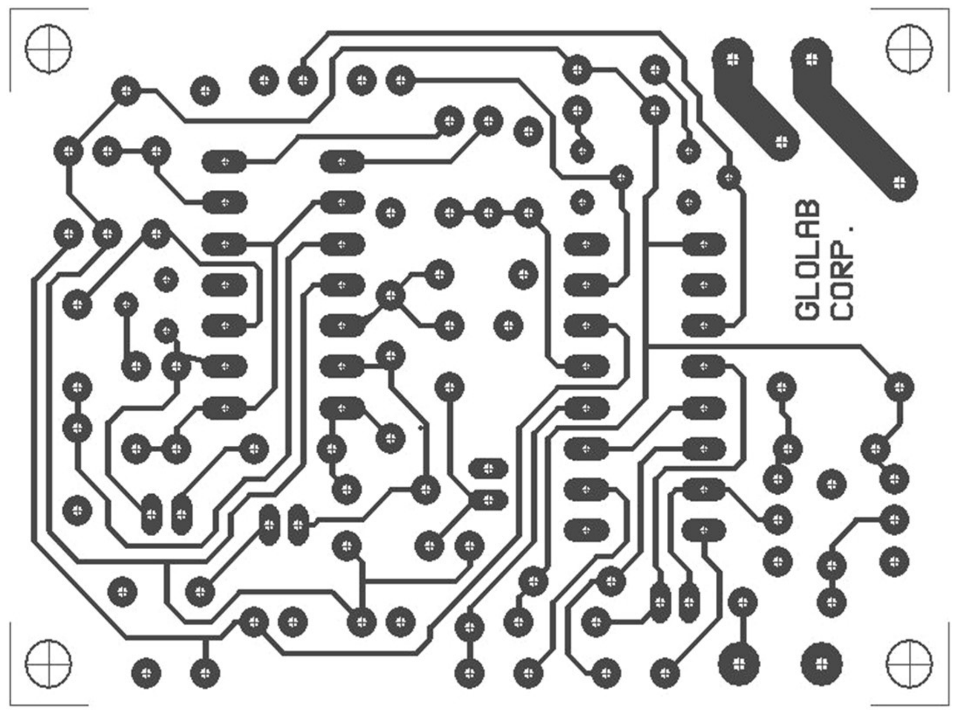
FPGA routing details

Wire-Wrapping vs Printed Circuits

In laboratory prototypes, we use wire-wrapping (using ordinary wires) to connect components, as we develop and test our design

Once the design has been finalized, the connections will be printed on a circuit board to make them both less cluttered and more reliable

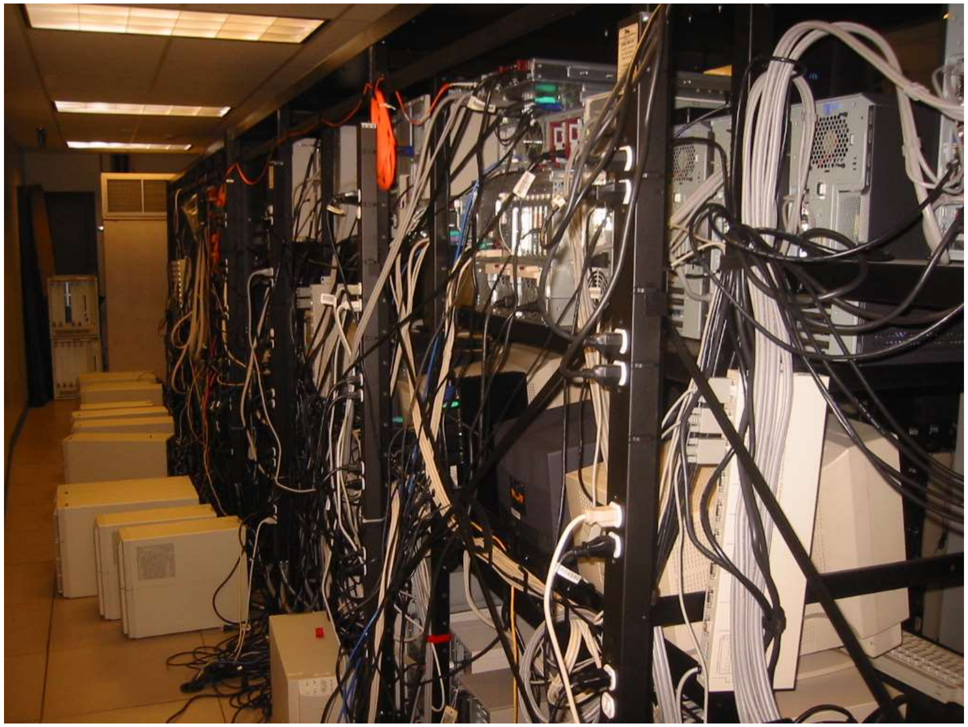




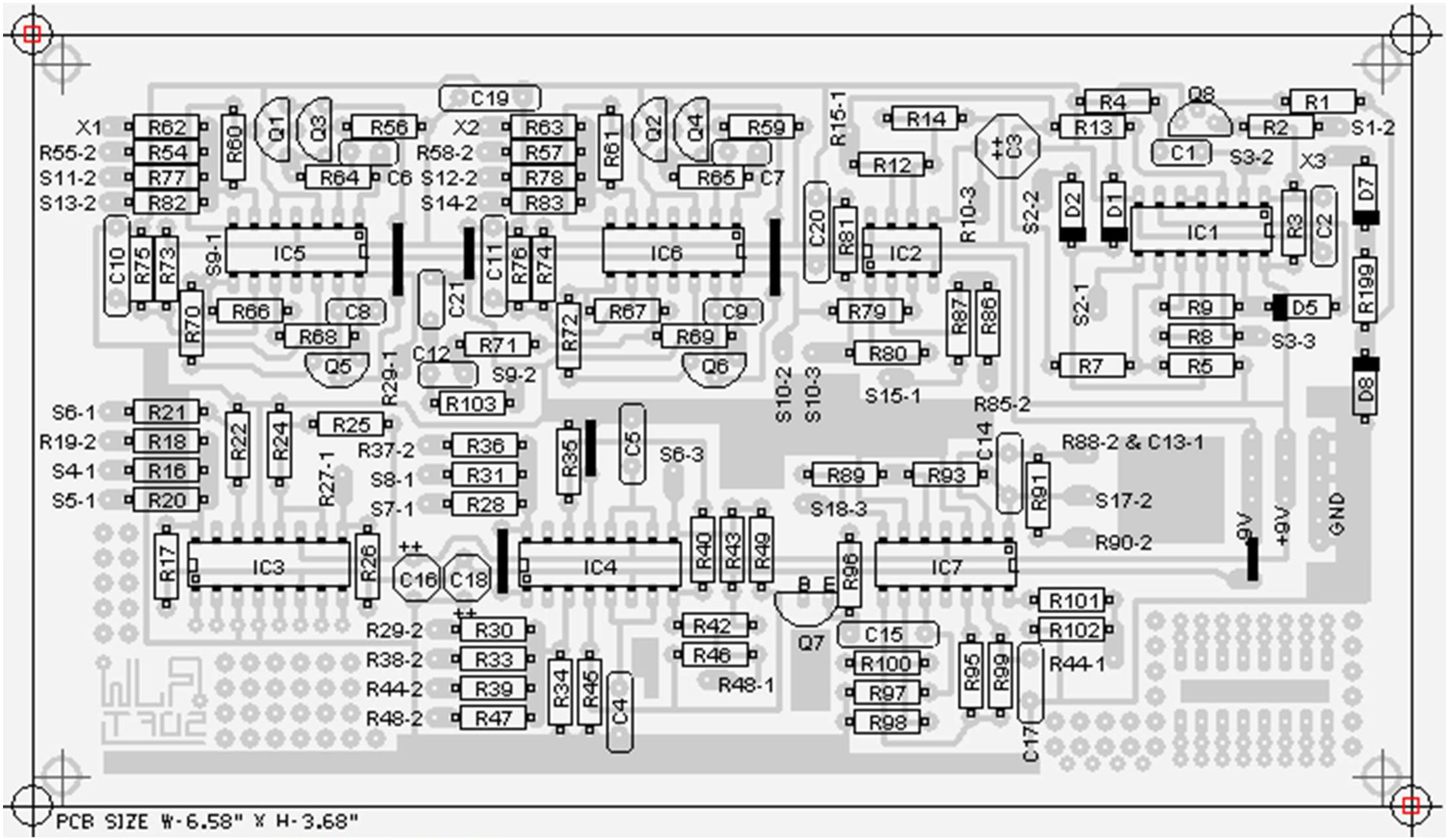
Backplane Wiring

Backplane wires located behind computer cabinets presents the same problems as wiring on a printed circuit board

Judicious placement of cabinets helps. Also, wires can be made neater and more tractable by using rectilinear paths and grouping cables

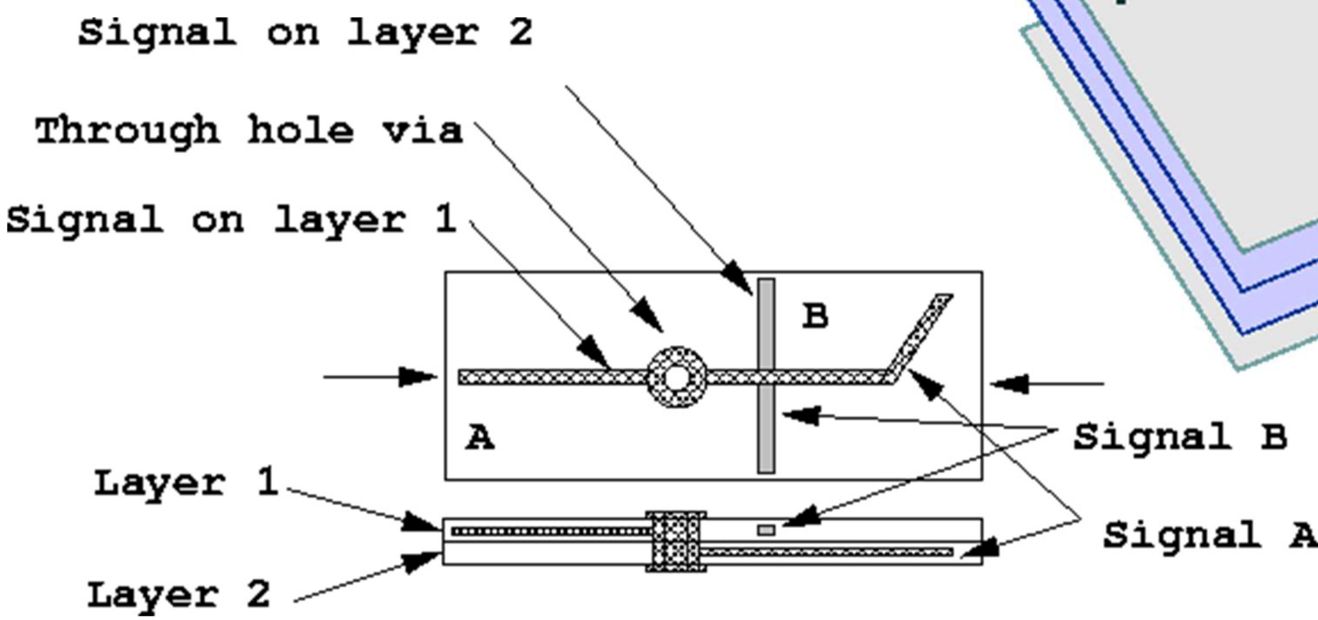


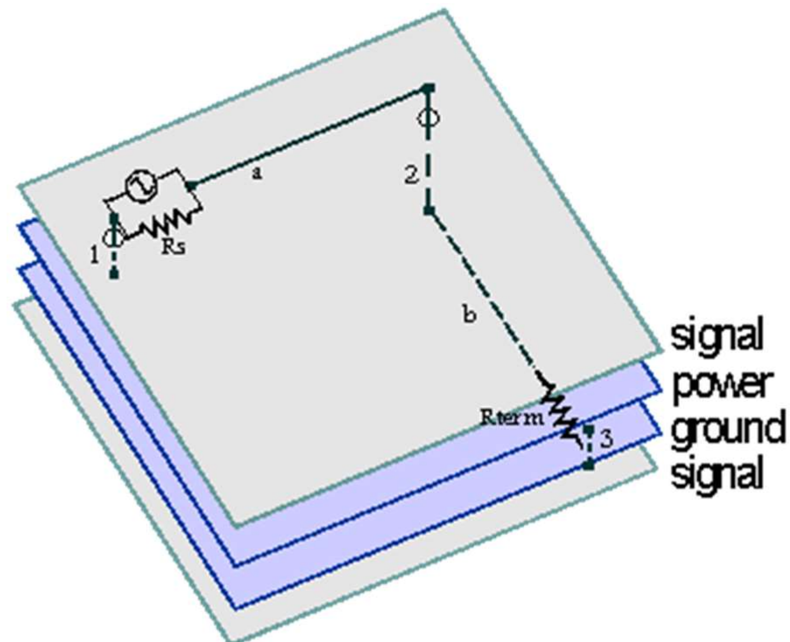
Single-Layer Routing on a PC Board



Multilayer Routing on PC Boards

Wires can cross each other if they are located at different levels
Through holes or “vias” can connect wires that are on different levels





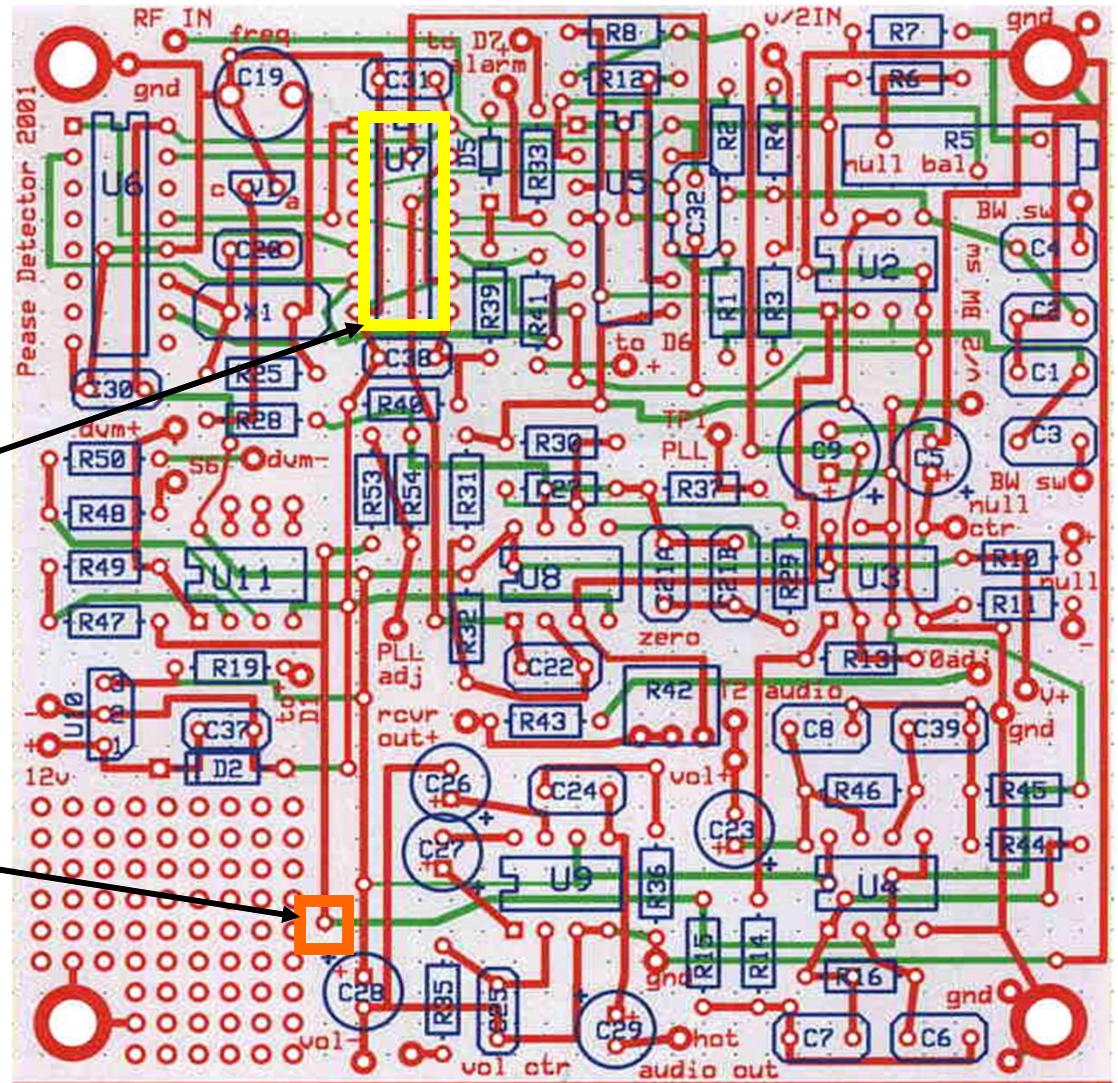
Example of 2-Layer Routing on a PC Board

Wires shown in red are mostly vertical

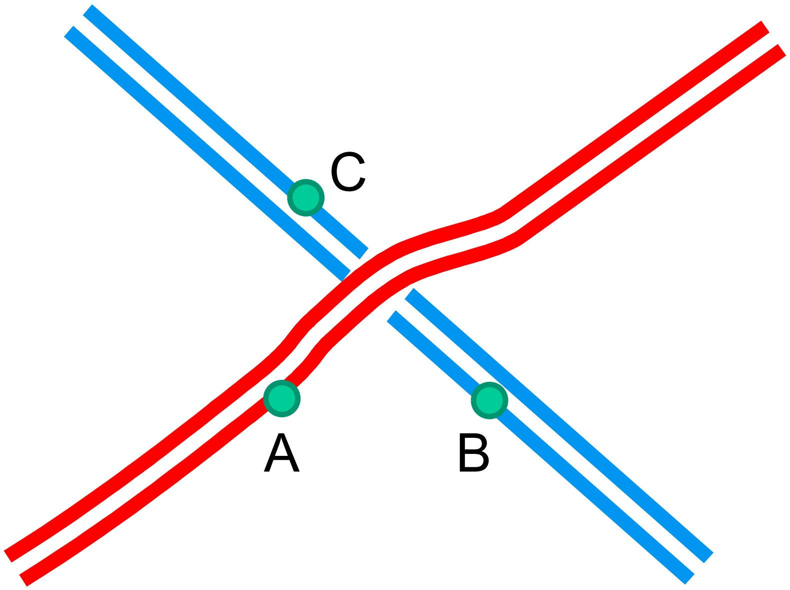
Wires shown in green are mostly horizontal

Example component

Example via



Freeway Interchange

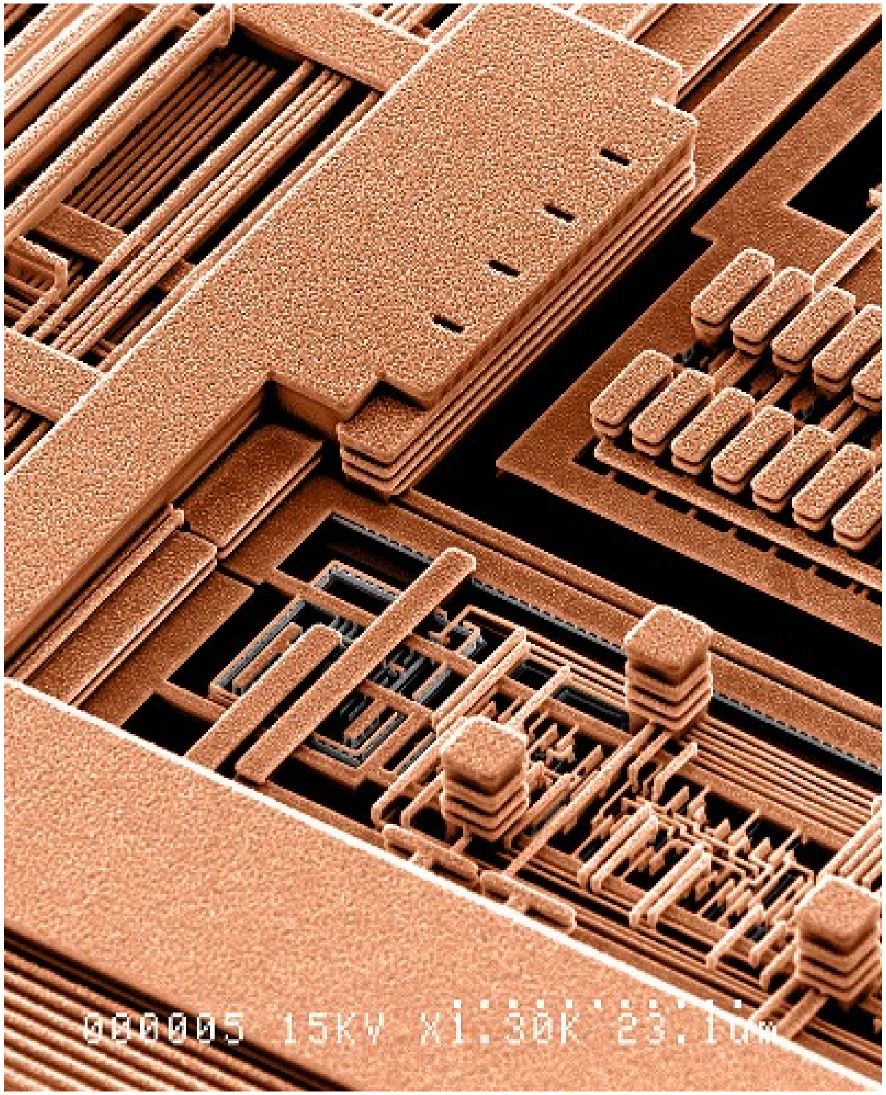


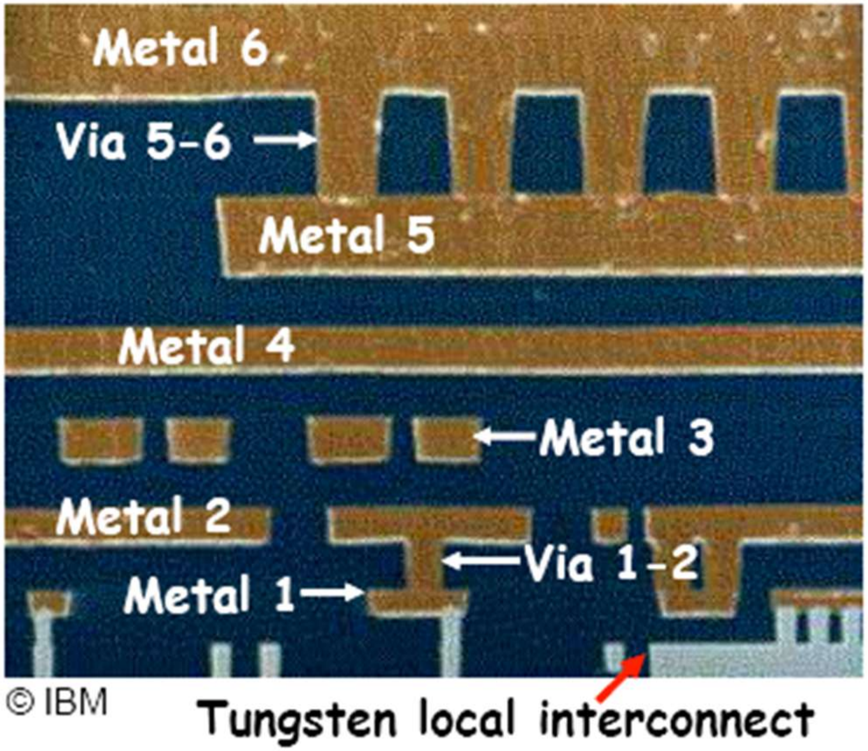
Nonintersection requirement leads to cloverleaf interchange

Put in a path from A to B and from A to C

Multilayer Crisscrossing Freeways

Multilayer Wiring in Integrated Circuits





© IBM

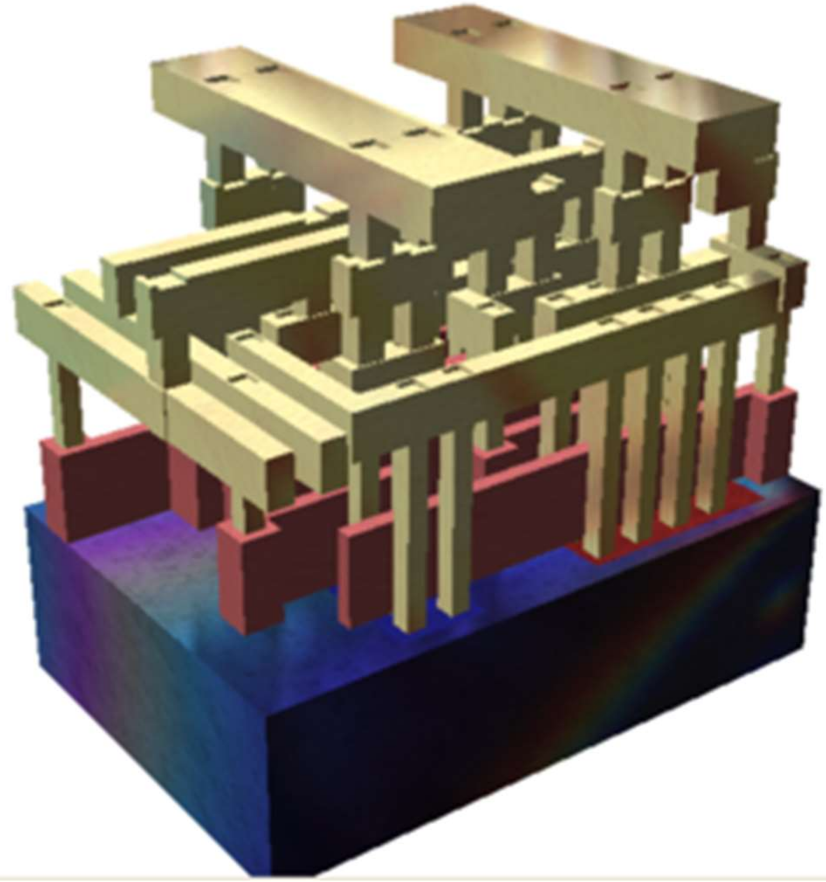
IBM CMOS7 Process

6 layers: copper wiring

1 layer: tungsten local interconnects

Visualizing Multilayer Wiring on a Chip

Photomicrograph of actual connections

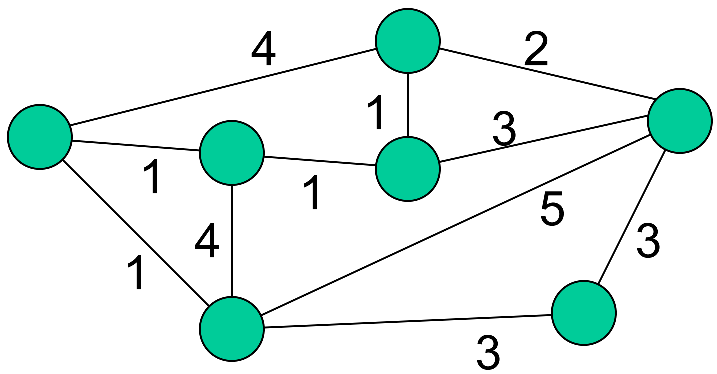


Drawing, with the vertical dimension exaggerated

The ability to connect many millions of transistors together, in a way that does not hamper signal propagation speed, is a main challenge today

Related Puzzling Problems to Think About

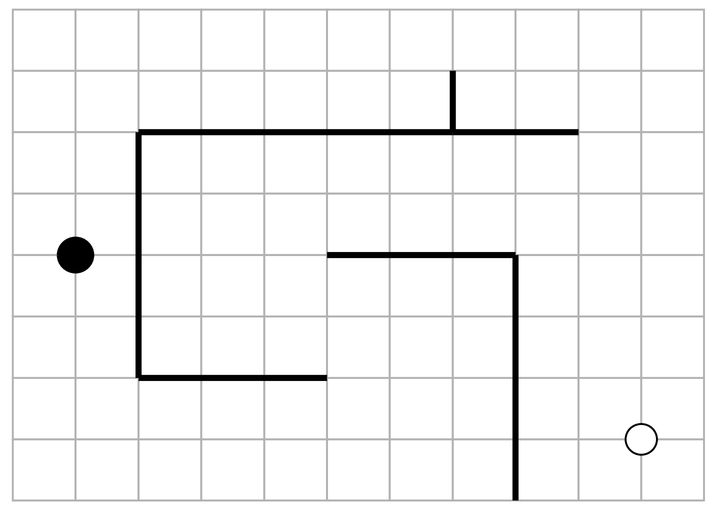
Resource placement: Place *n* fire stations in a city to minimize the worst-case response time. Alternatively, given a desired worst-case response time, what is the minimum number of fire stations needed?



View the city as a number of intersections, connected by streets (often a planar graph); numbers indicate travel times for fire trucks

Moving in a room with obstacles: Robot (black circle, with non-zero size) must move to the location of the white circle via a rectilinear path

Also, routing of wires when there are some restrictions (e.g., placed components or existing wires)



Q4: Draw a shortest rectilinear path from the black circle to the white one.