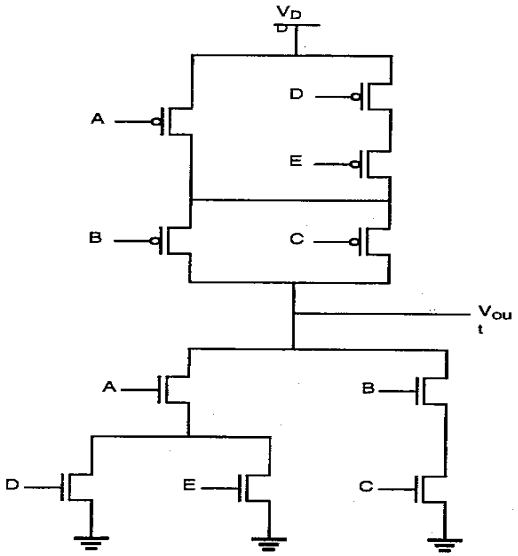
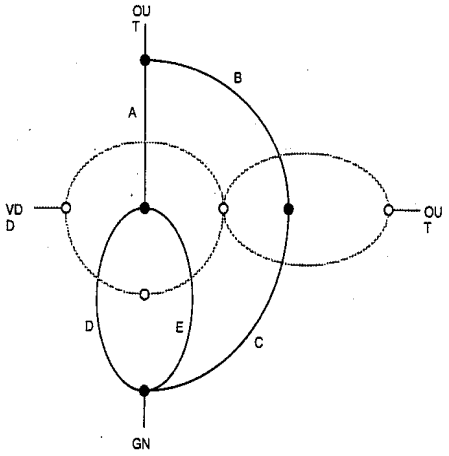
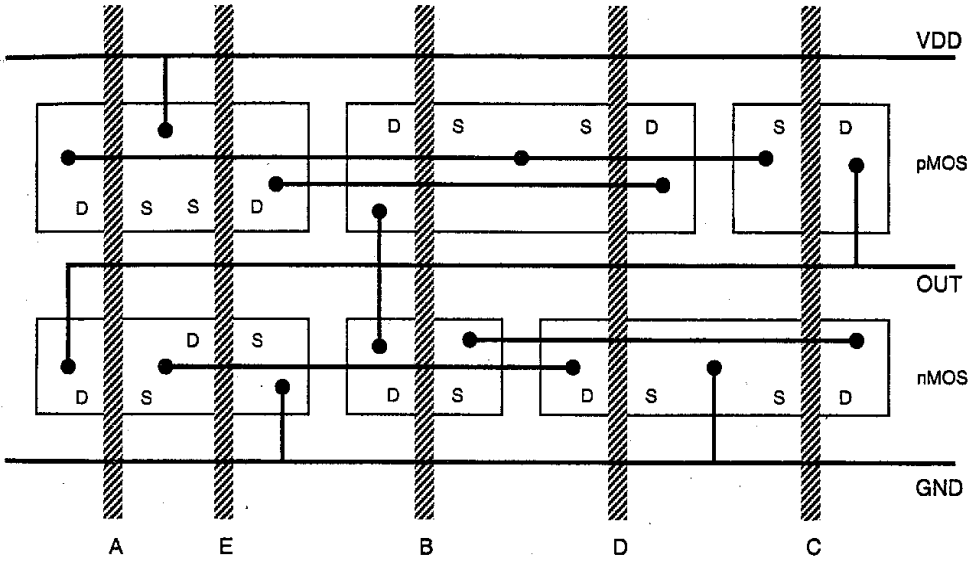
Layout Technique using Euler Graph Method

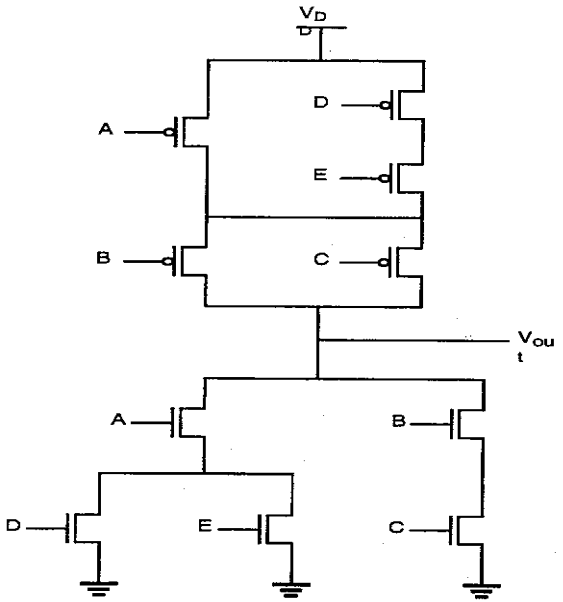




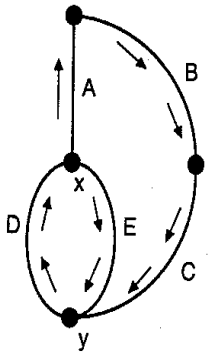


- Euler Graph Technique can be used to determine if any complex CMOS gate can be physically laid out in an optimum fashion
  - Start with either NMOS or PMOS tree (NMOS for this example) and connect lines for transistor segments, labeling devices, with vertex points as circuit nodes.
  - Next place a new vertex within each confined area on the pull-down graph and connect neighboring vertices with new lines, making sure to cross each edge of the pull-down tree only once.
  - The new graph represents the pull-up tree and is the dual of the pull-down tree.
- The stick diagram at the left (done with arbitrary gate ordering) gives a very non-optimum layout for the CMOS gate above.

Layout with Optimum Gate Ordering



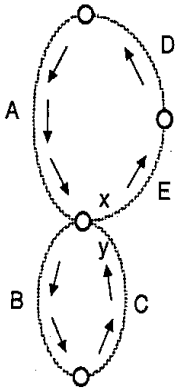
- By using the Euler path approach to re-order the polysilicon lines of the previous chart, we can obtain an optimum layout.
- Find a Euler path in both the pull-down tree graph and the pull-up tree graph with identical ordering of the inputs.
  - Euler path: traverses each branch of the graph exactly once!
- By reordering the input gates as E-D-A-B-C, we can obtain an optimum layout of the given CMOS gate with single actives for both NMOS and PMOS devices (below).



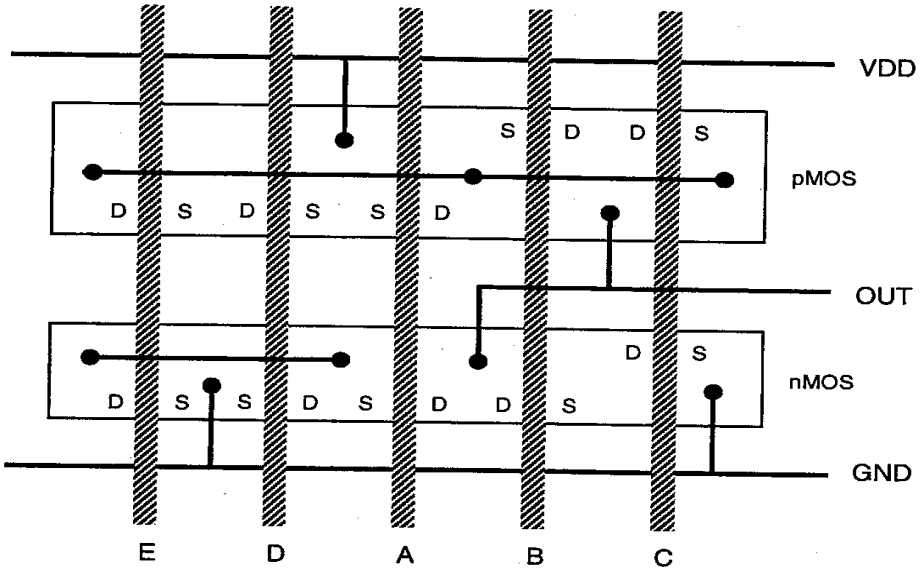
nMOS network

Common Euler path :

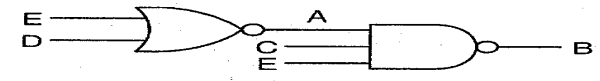
E - D - A - B - C

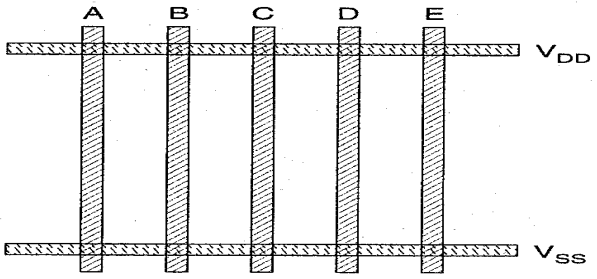


pMOS network

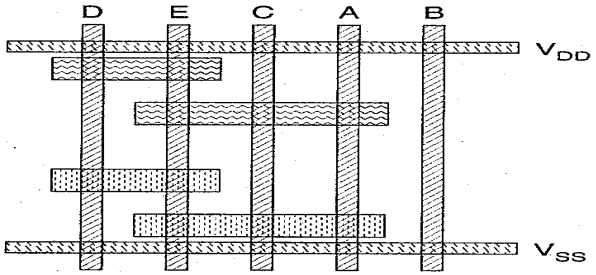


Automated Approach to CMOS Gate Layout

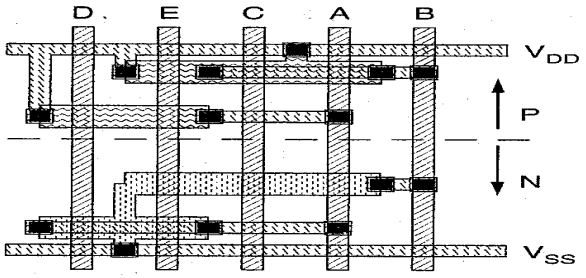




(a) Route Power, Ground, Gate and Output Signals



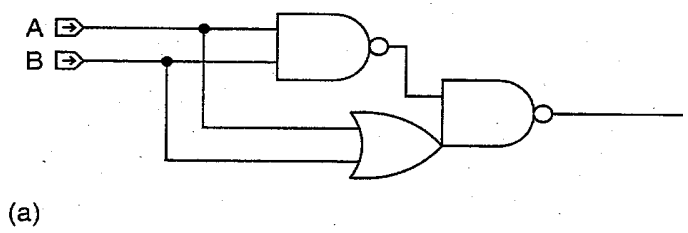
(b) Order Gate and Outputs to optimize horizontal transistor connectivity



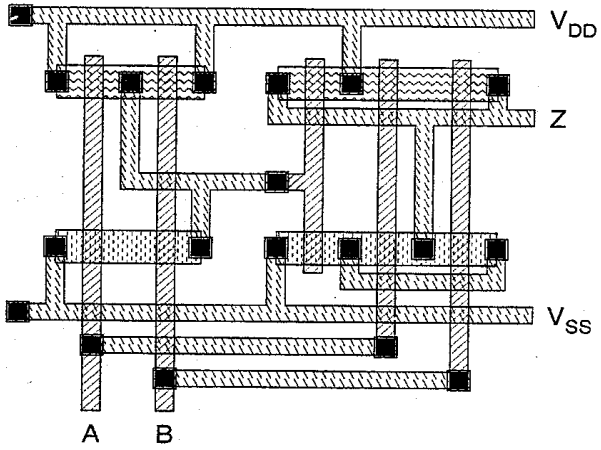
(c) Rearrange vertical strip ordering to optimize power routing internal gate connection and output connections

- Place inputs as vertical poly stripes
- Place Vdd and Vss as horizontal stripes
- Group transistors within stripes to allow maximum source/drain connection
- Allow poly columns to interchange in necessary to improve stripe wireability
- Place device groups in rows
- Wire up the circuit by using vertical diffusions for connections and manhattan metal routing (both horizontal and vertical)

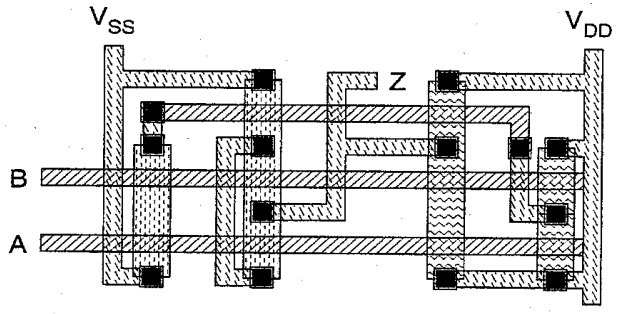
Complementary CMOS XNOR Gate Layouts



(a)



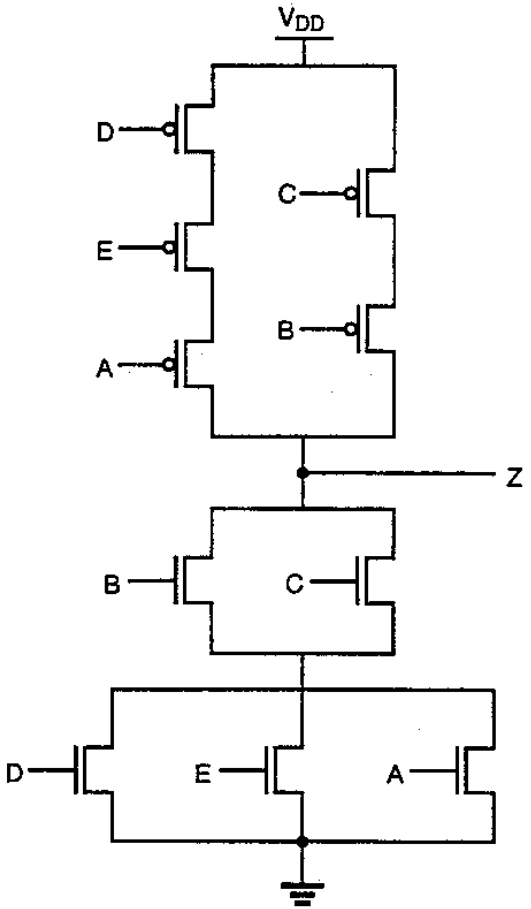
(b)



(c)

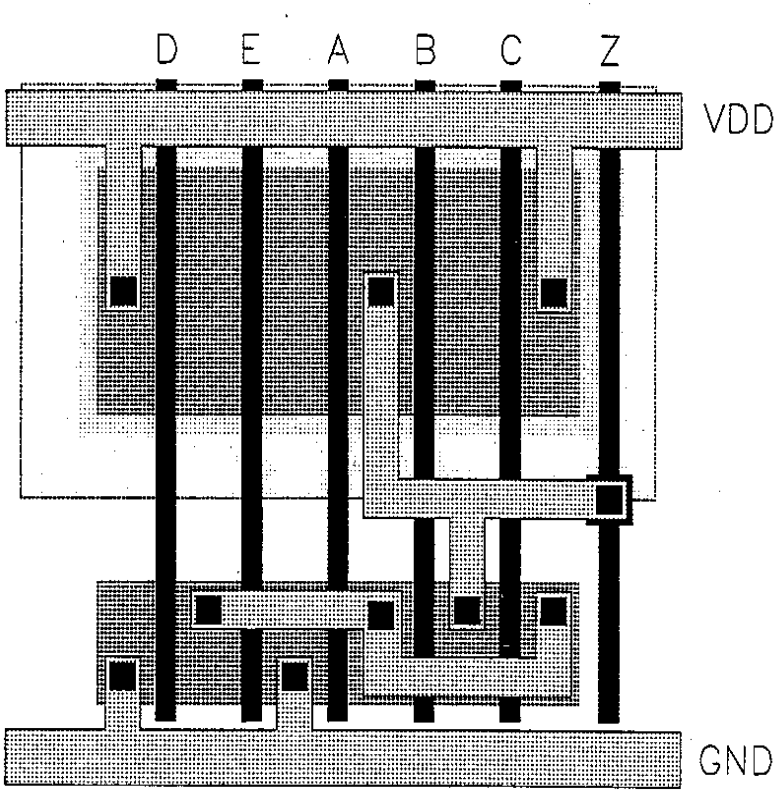
- XNOR is an example of a complementary CMOS circuit where a single input is applied to the gates of multiple transistors in the N (and P) tree:
  - Separate sections and stack transistors for each section over identical gate inputs
  - XNOR implementation in (b) shows separate sections with X = (AB)' and Z = ((A + B) X)' = XNOR (A,B)
    - Uses single row of N (and P) transistors with a break between the active regions
  - Alternate layout in (c) uses vertical device regions perhaps making it a bit more compact

Euler Method Example: OAI Circuit Schematic



- Use the Euler Method to layout the OAI circuit at the left.
  - Can the circuit be laid out with optimum layout area?
    - Single poly stripes
    - Single active shapes
- Method:
  - Find NMOS network graph
  - Find PMOS network graph
  - Find a traverse of both graphs with a common Euler path
- Answer on next chart!

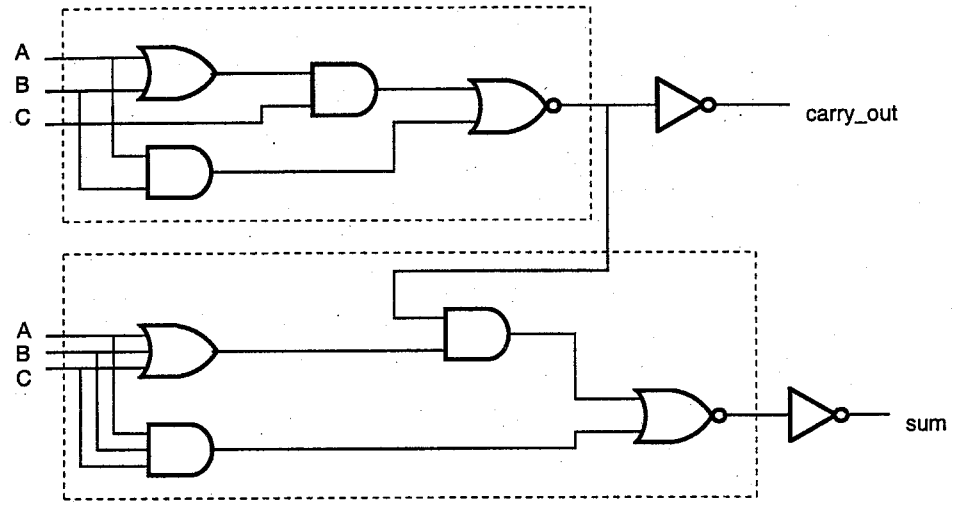
Example Layout: OAI Circuit of Chart 19a

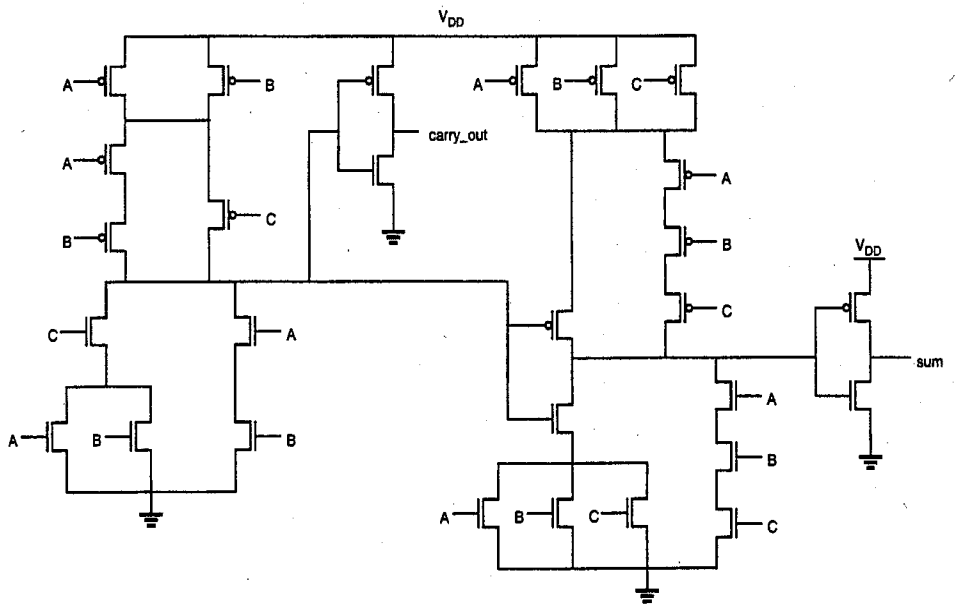


- The layout at the left is an optimum layout of the OAI circuit of chart 19a
  - Single poly vertical inputs
  - Unbroken single active regions for both N and P transistors

- Problem: Find an equivalent inverter circuit for the layout at left assuming the following
  - $W/L)_p = 15$ for all PMOS transistors
  - $W/L)_N = 10$ for all NMOS transistors

CMOS 1-Bit Full Adder Circuit





- 1-Bit Full Adder logic function:

  Sum = A XOR B XOR C

  = ABC + AB'C' + A'BC' + A'B'C

  Carry_out = AB + AC + BC

  - Exercise: Show that the sum function can be written as shown at left

  Sum = ABC + (A + B + C) · carry_out'

- This alternate representation of the sum function allows the 1-bit full adder to be implemented in complex CMOS with 28 transistors, as shown at left below.
  - Carry_out' internal node is used as an input to the adder complex CMOS gate
  - Exercise: Show that the two P-trees in the complex CMOS gates of the carry_out and sum are optimizations of the proper dual derivations from the two N-tree networks.

CMOS Full Adder Layout (Complex Logic)

- Mask layout of 1-bit full adder circuit is shown below
  - A layout designed with Euler method shows that the carry_out inverter requires separate active shapes, but all other N (and P) transistors were laid out in a single active region
  - Layout below is non-optimized for performance
    - All transistors are seen to be minimum W/L
- Design of n-bit full adder:
  - A carry ripple adder design uses the carry_out of stage k as the carry_in for stage k+1
  - Typically the layout is modified from that shown below in order to use larger transistors for the carry_out CMOS gate in order to improve the performance of the ripple bit adder
    - See Fig. 7.30 in Kang and Leblebici