Elective II: VLSI Design

Code: CISM 402

Pritha Banerjee

Courtsey for slides: Debasis Mitra, NIT Durgapur

Design and Fabrication of devices

- Books:
  - Chapter 2,3 of Naveed A. Sherwani, *Algorithms for VLSI Physical Design Automation*, Kluwer Academic Publishers

The underlying technology

- Chip has several layers of different materials on silicon wafer
  - Shape, size and location of materials in each layer is given by mask
  - Layers are ordered
  - Devices formed by overlapping different materials in different layers
  - Design rules required to ensure proper fabrication
- Transistors : basic building blocks of chip
  - nMOS: n-type Metal Oxide Semiconductor Field effect Transistor (nMOSFET)
  - pMOS: p-type Metal Oxide Semiconductor
  - CMOS: Complementary Metal Oxide Semiconductor

MOS Transistors

Silicon –a semiconductor, forms the starting material

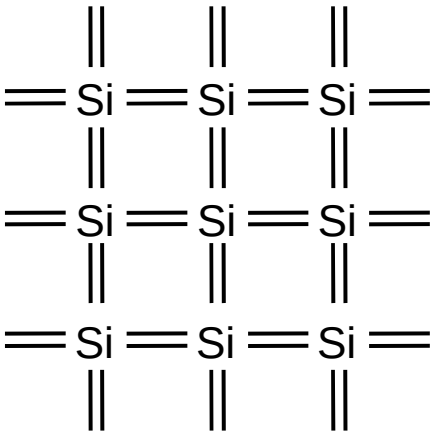
MOS (Metal Oxide Silicon) structure created by superimposing several layers of conducting, insulating, and transistor forming materials

n-type/nMOS: Electrons carry charge
p-type/pMOS: Holes carry charge

- CMOS transistors
- Building logic gates from transistors
- Transistor layout and fabrication

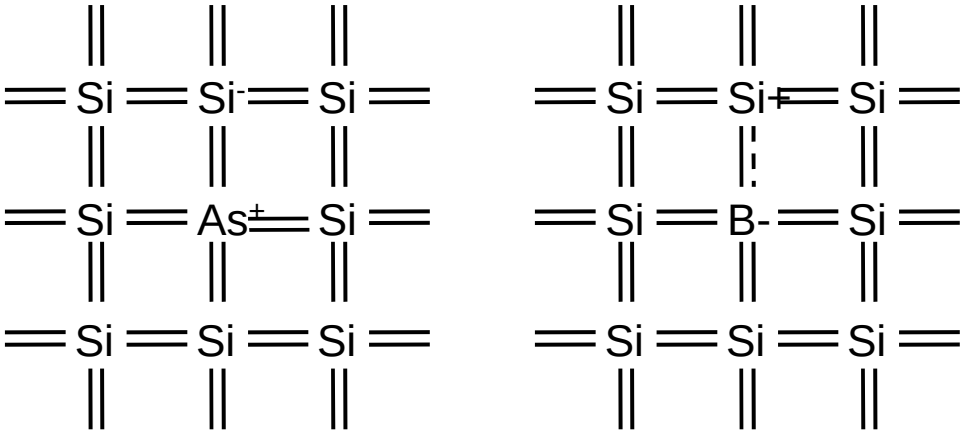
Silicon Lattice

- Transistors are built on a silicon substrate
- Silicon is a Group IV material
- Forms crystal lattice with bonds to four neighbors



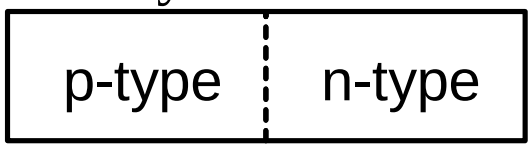
Dopants

- Silicon is a semiconductor
- Pure silicon has no free carriers and conducts poorly
- Adding dopants increases the conductivity
- Group V: extra electron (n-type);Arsenic;
- Group III: missing electron , called hole (p-type); Boron

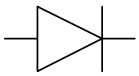


P-n junction

- A junction between p-type and n-type semiconductor forms a diode.
- Current flows only in one direction
- Voltage at p-type > volt. At n-type ; current flows; else very little current flows

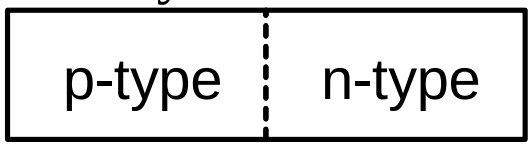


anode cathode

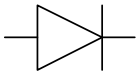


P-n junction

- A junction between p-type and n-type semiconductor forms a diode.
- Current flows only in one direction
- Voltage at p-type > volt. At n-type ; current flows; else very little current flows

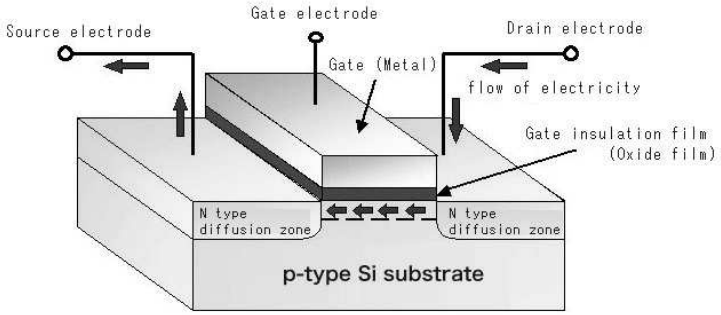


anode cathode



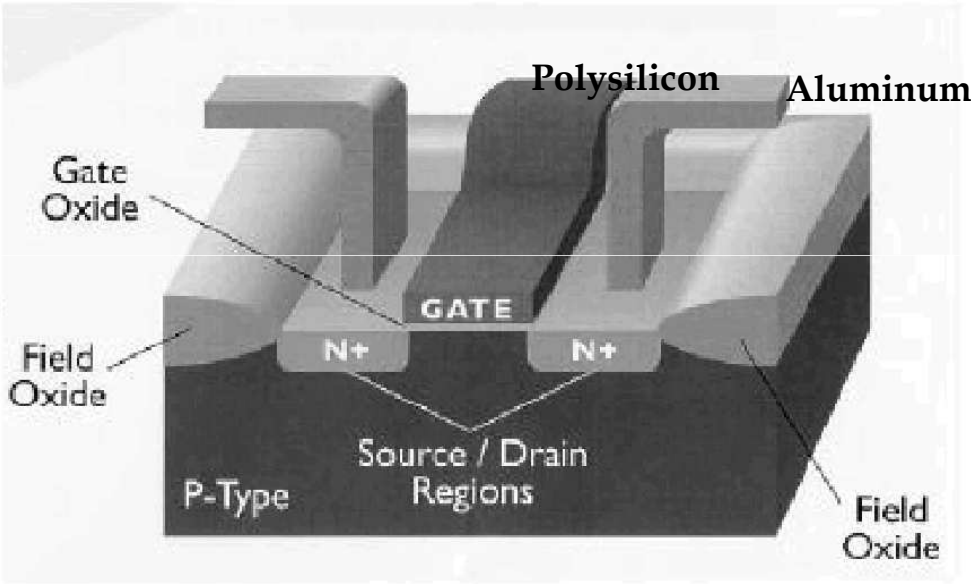
An n-MOS Transistor

- Four terminals: gate, source, drain, body
- Gate – oxide – body stack looks like a capacitor
  - Gate and body are conductors
  - SiO_2 (oxide) is a very good insulator
  - Gate is polycrystalline silicon
  - Body : grounded
  - P-n junction: no current Flow
  - Gate <= 0, drain and source disconnected
  - Gate <=1, current flow from drain to source

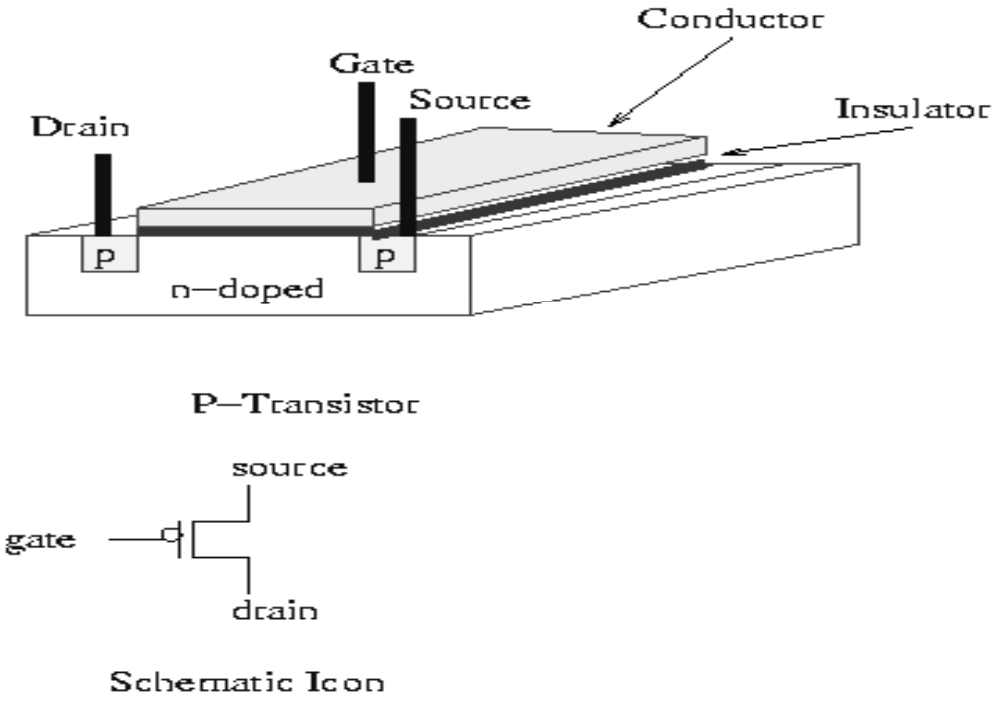


Construction of MOSFET

An n-MOS Transistor: 3D Perspective

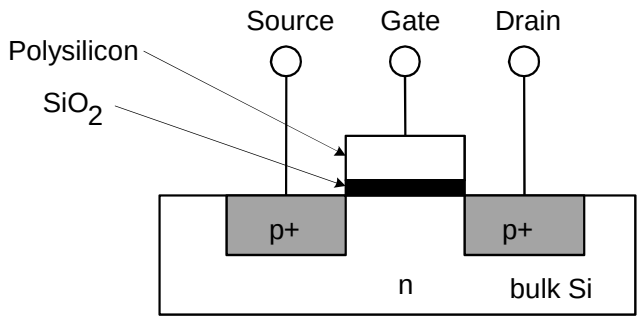


An p-MOS Transistor



An p-MOS Transistor

- Similar, but doping and voltages reversed
  - Body tied to high voltage (V_{DD})
  - Gate low: transistor ON
  - Gate high: transistor OFF
  - Bubble indicates inverted behavior



Working of MOS Transistors

N-type: If gate is ‘0’the source is disconnected from the drain and if gate is ‘1’, source is connected to the drain

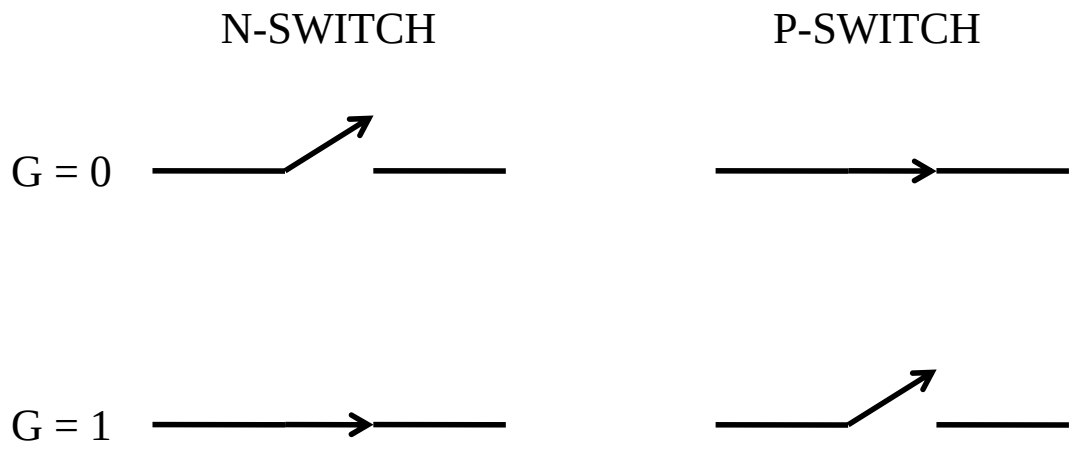
P-type: If gate is ‘0’the source is connected to the drain and if gate is ‘1’, source is disconnected from the drain

Working of MOS Transistors

N-type: If gate is ‘0’the source is disconnected from the drain and if gate is ‘1’, source is connected to the drain

P-type: If gate is ‘0’the source is connected to the drain and if gate is ‘1’, source is disconnected from the drain

MOS Transistors as Switches



More on MOS Switches

N-SWITCH:

perfect if ‘0’ is to be passed but
imperfect when ‘1’ is to be passed

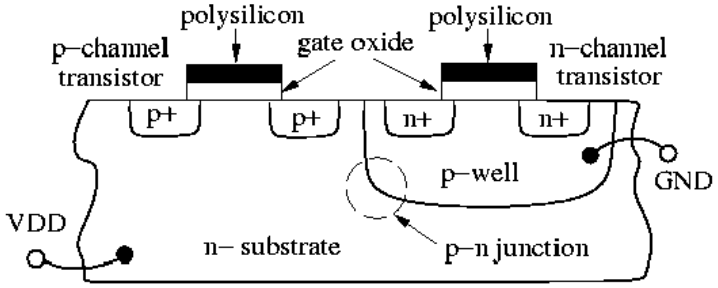
P-SWITCH:

perfect if ‘1’ is to be passed but
imperfect when ‘0’ is to be passed

N and P switches are ON/OFF on complementary values of gate signal, need both types to get perfect 0 and 1

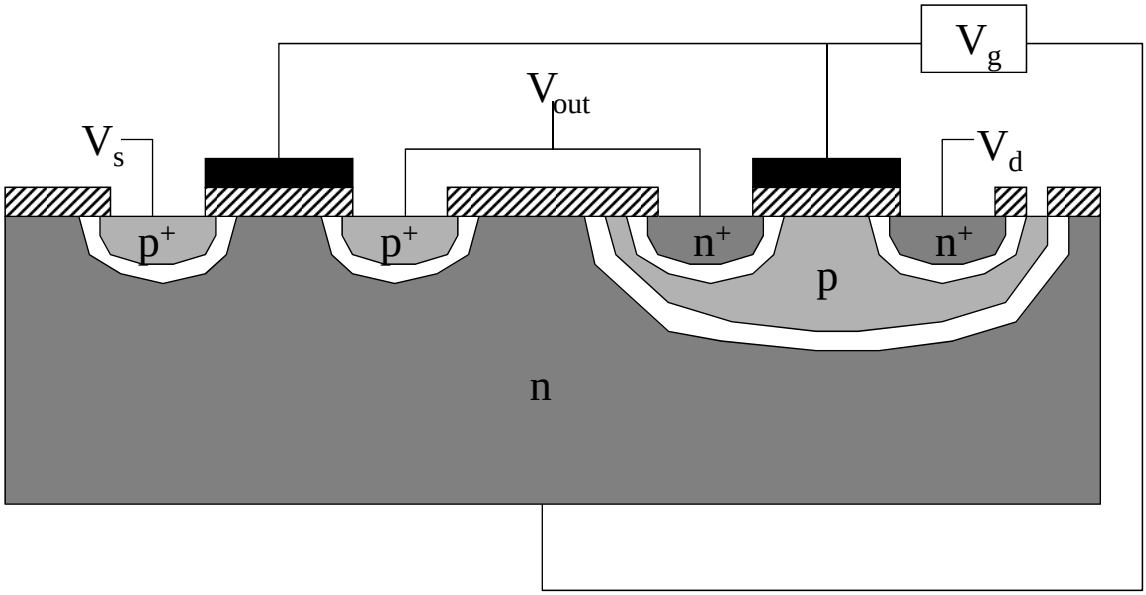
Complementary MOS (CMOS)

- The most popular VLSI technology (v.s. BiCMOS, nMOS).
- CMOS uses both *n*-channel and *p*-channel transistors.
- Advantages: lower power dissipation, higher regularity, more reliable performance, higher noise margin, larger fanout, etc.
- Each type of transistor must sit in a material of the complementary type (the reverse-biased diodes prevent unwanted current flow).

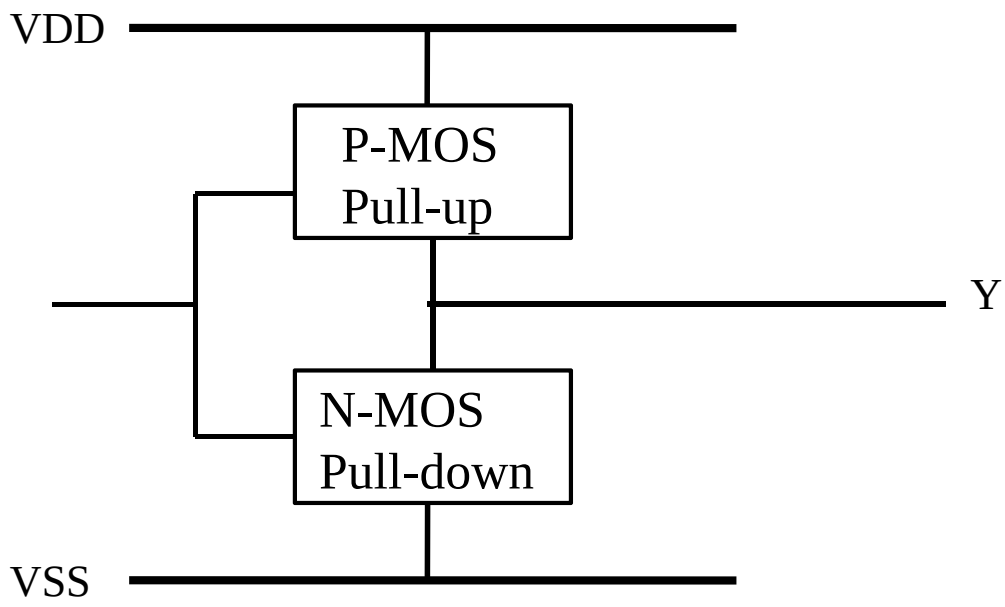


p–well CMOS structure

Complementary MOS (CMOS)



CMOS Technology

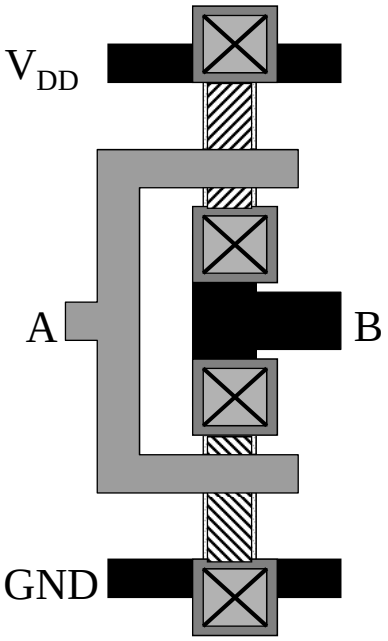


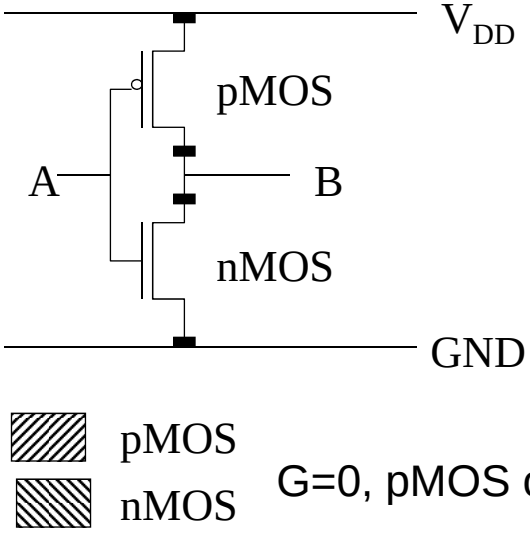
Why CMOS Technology

For all input combinations, there is always a path either from ‘VDD’ to the output or from the output to ‘VSS’

There is never a direct path between ‘VDD’ and ‘VSS’–this is the basis for the low static power dissipation for CMOS

CMOS Inverter





pMOS

nMOS

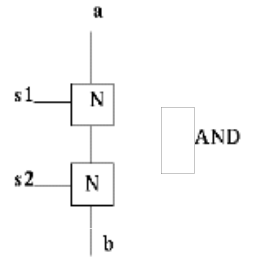
G=0, pMOS on

G =1, nMOS on

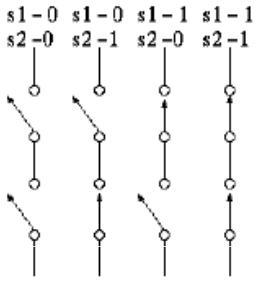
CMOS Transistor layout

Series and Parallel connections

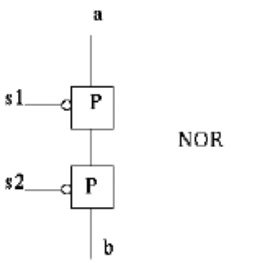
- nMOS: 1 = ON
- pMOS: 0 = ON
- *Series*: both must be ON
- *Parallel*: either can be ON



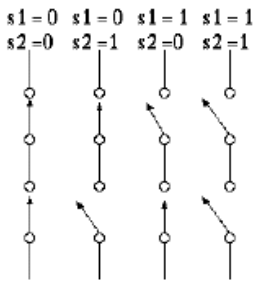
AND



s2 \ s1	0	1
0	off	off
1	off	on



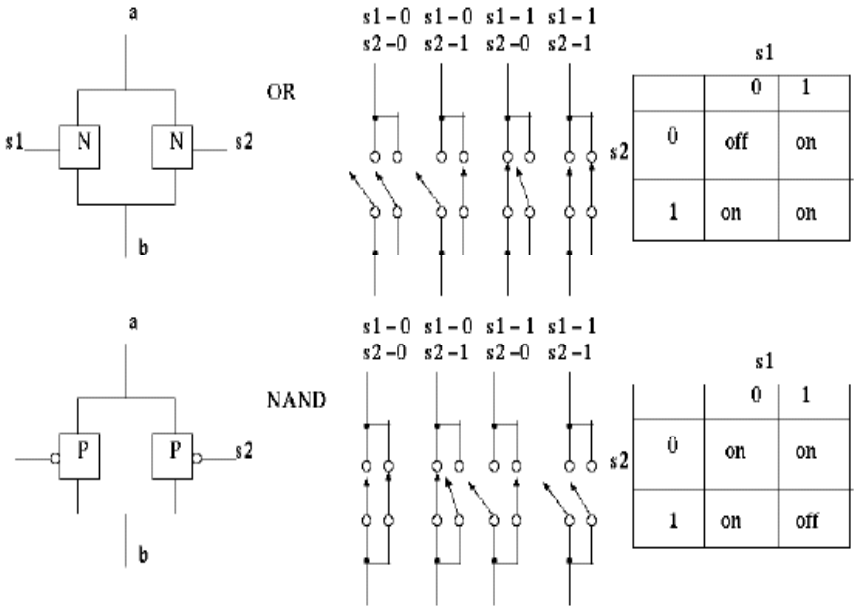
NOR



s2 \ s1	0	1
0	on	off
1	off	off

Series and Parallel connections

- nMOS: 1 = ON
- pMOS: 0 = ON
- *Series*: both must be ON
- *Parallel*: either can be ON



Logic Design in CMOS

Fundamentals:

Series: combination is ON when both are ON

Parallel: combination is ON when either is ON

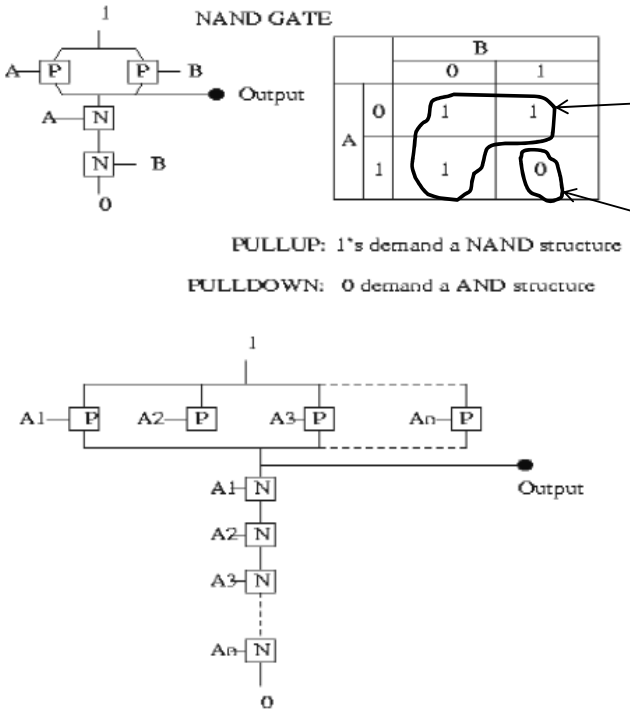
P switches in Parallel : A' + B' + C'+ ….. [or (A.B.C…..)']
P switches in Series : A'.B'.C'…. [or (A+B+C+…)']

N switches in Parallel : A + B + C + …..
N switches in Series : A.B.C….

Build NAND using CMOS

NAND gate

2-input NAND gate

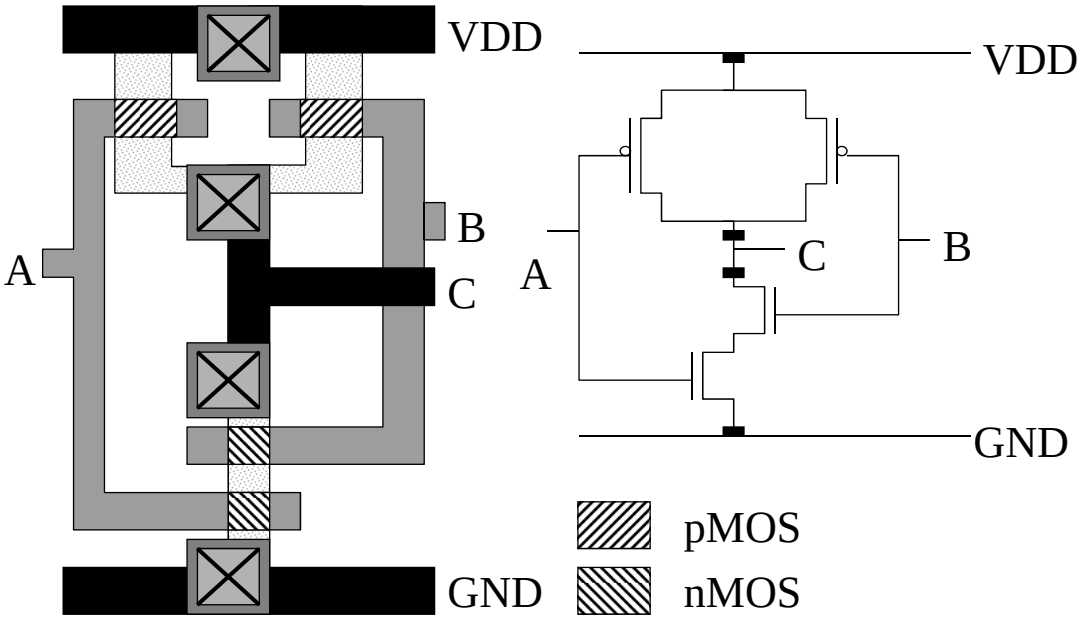


Pull-up implements Original function

Pull-down implements complement of the function

n-input NAND gate

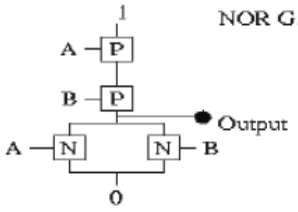
CMOS NAND Gate



Buid NOR using CMOS

NOR gate

2-input NOR gate



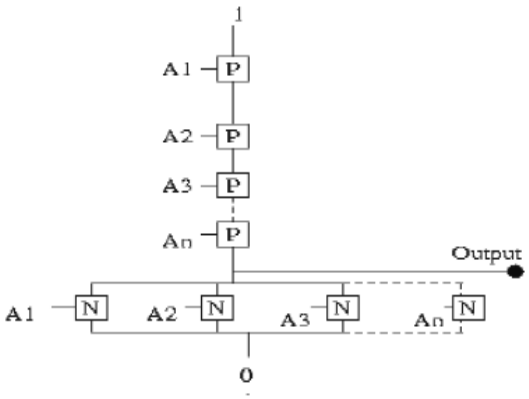
NOR GATE

		B	
		0	1
A	0	1	0
	1	0	0

PULLUP: 1 demands a NOR structure

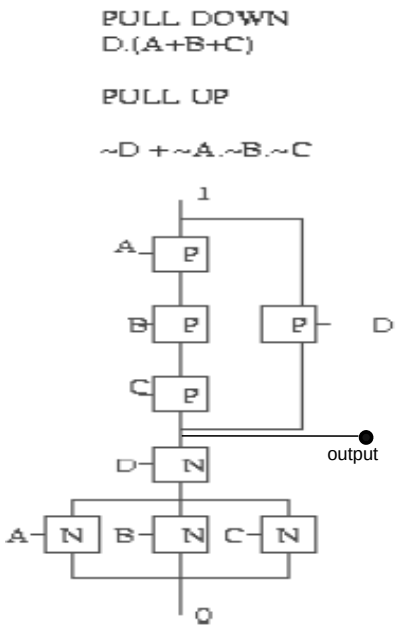
PULLDOWN: 0's demand a OR structure

n-input NOR gate



Logic Design in CMOS

General Boolean function: ((A+B+C).D)'



AB / CD	00	01	11	10
00	1	1	1	1
01	1	0	0	0
11	0	0	0	0
10	1	1	1	1

NOTE: Karnaugh map can be used to minimize the given function

Logic Design in CMOS

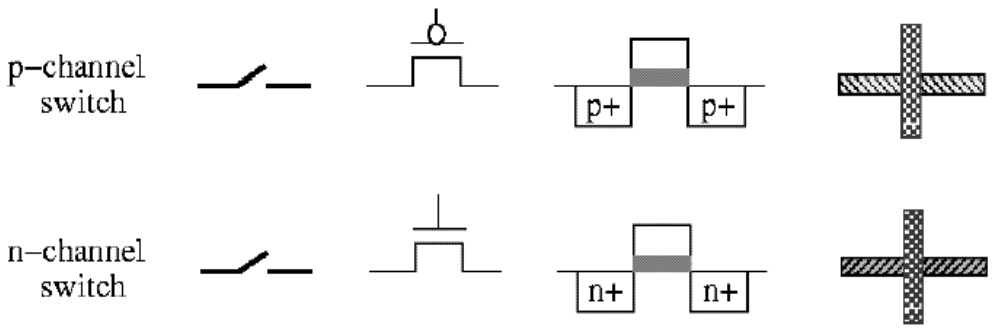
Assignment 1: Design the following Boolean functions in CMOS.

((A.B) + (C.D))'

(((A.B) + C).D)'

Stick Diagram

- Intermediate representation between the transistor level and the mask (layout) level.
- Gives topological information (identifies different layers and their relationship)
- Assumes that wires have no width.
- Possible to translate stick diagram automatically to layout with correct **design rules**.

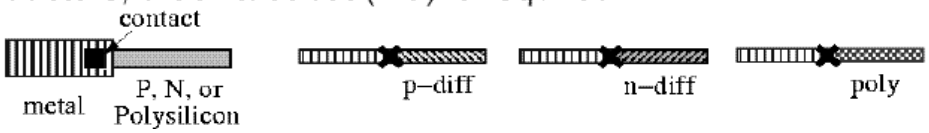


Stick Diagram

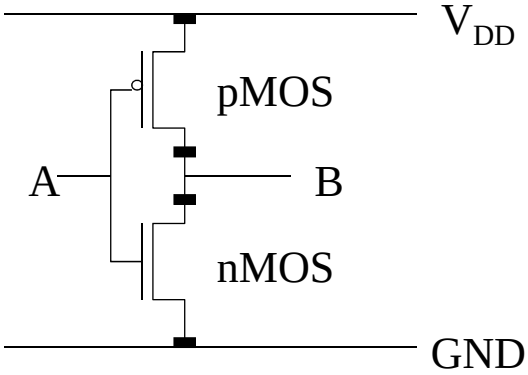
- When the same material (on the same layer) touch or cross, they are connected and belong to the same electrical node.

- When **polysilicon** crosses N or P **diffusion**, an N or P transistor is formed.
  - Polysilicon is drawn on top of diffusion.
  - Diffusion must be drawn connecting the source and the drain.
  - Gate is automatically self-aligned during fabrication.

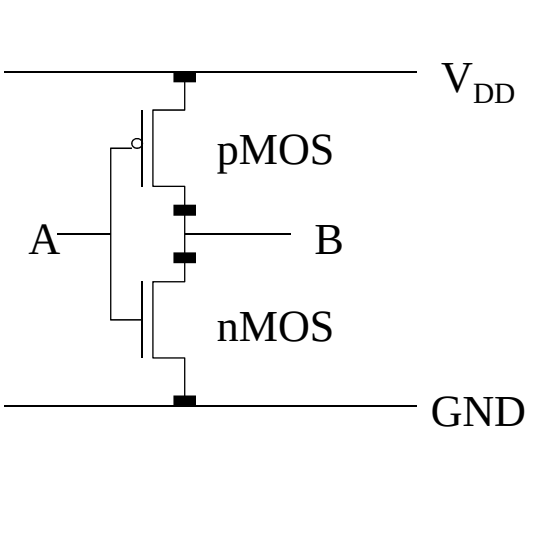
- When a metal line needs to be connected to one of the other three conductors, a **contact** cut (**via**) is required.



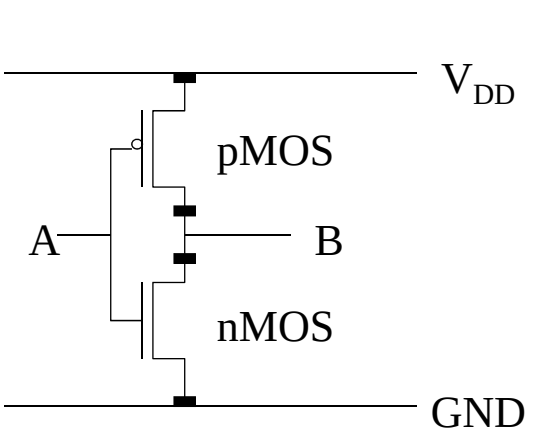
CMOS Inverter Stick Diagrams

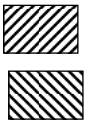


CMOS Inverter Stick Diagrams

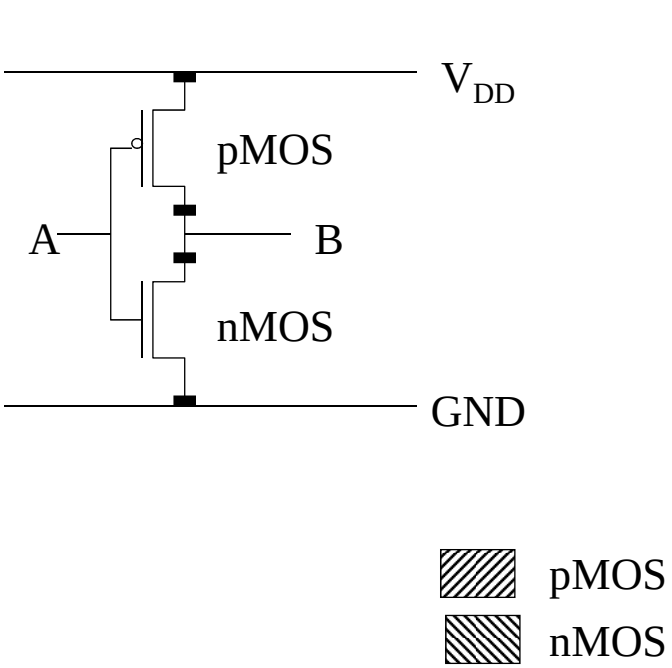


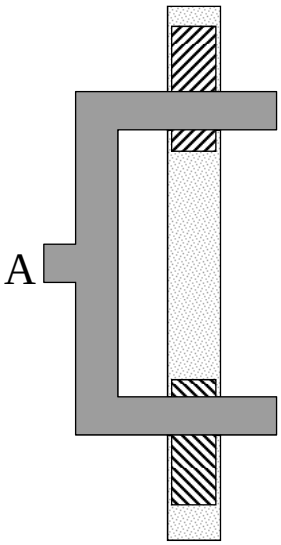
CMOS Inverter Stick Diagrams





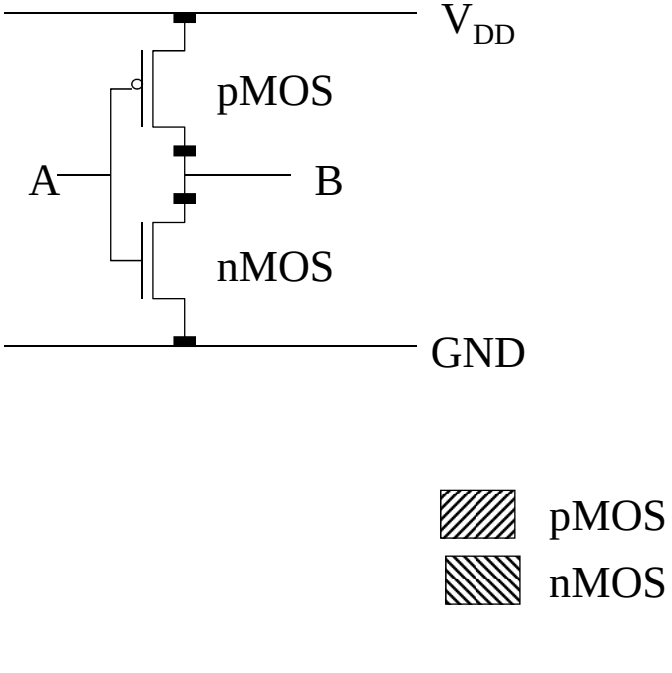
CMOS Inverter Stick Diagrams

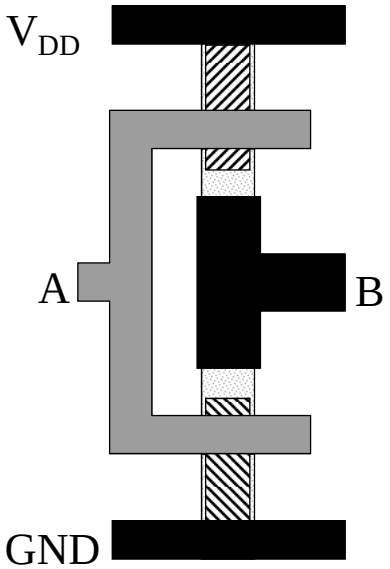




pMOS
nMOS

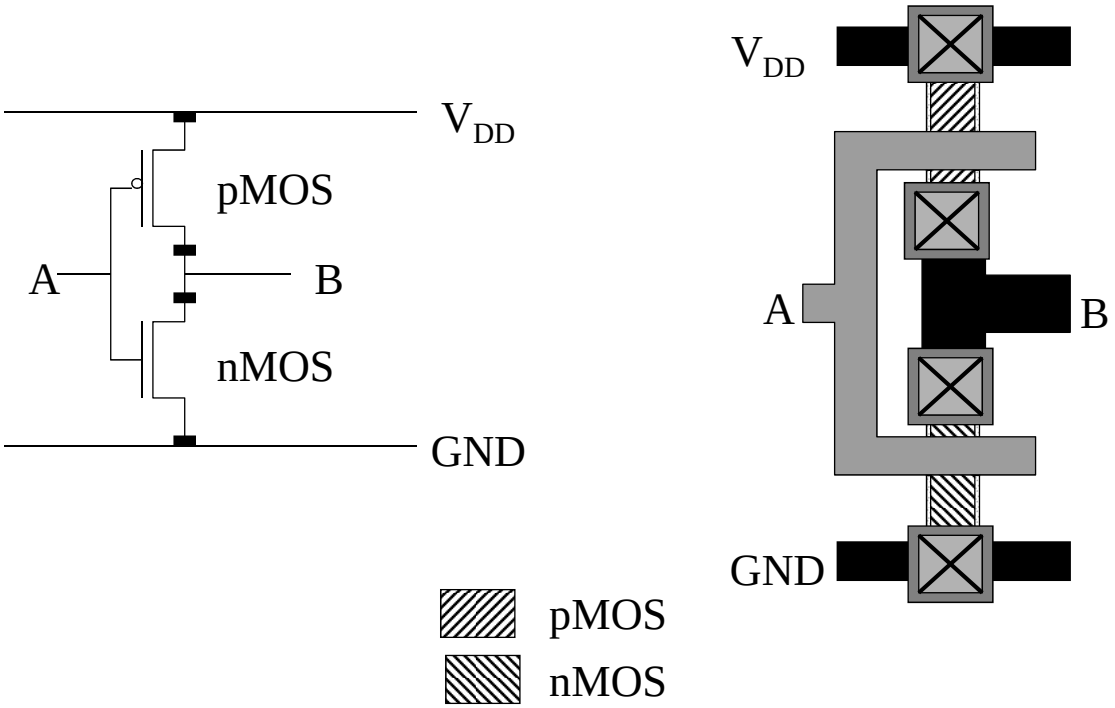
CMOS Inverter Stick Diagrams





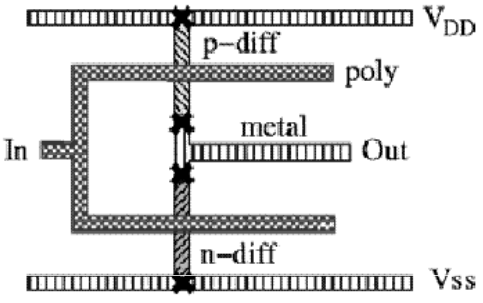
pMOS
nMOS

CMOS Inverter Stick Diagrams

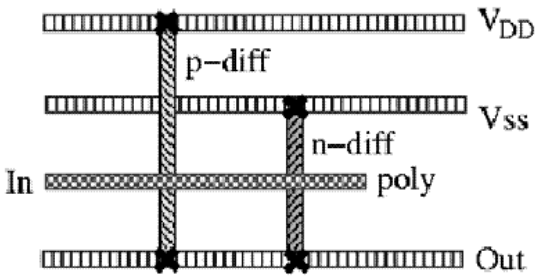


CMOS Inverter Stick Diagrams

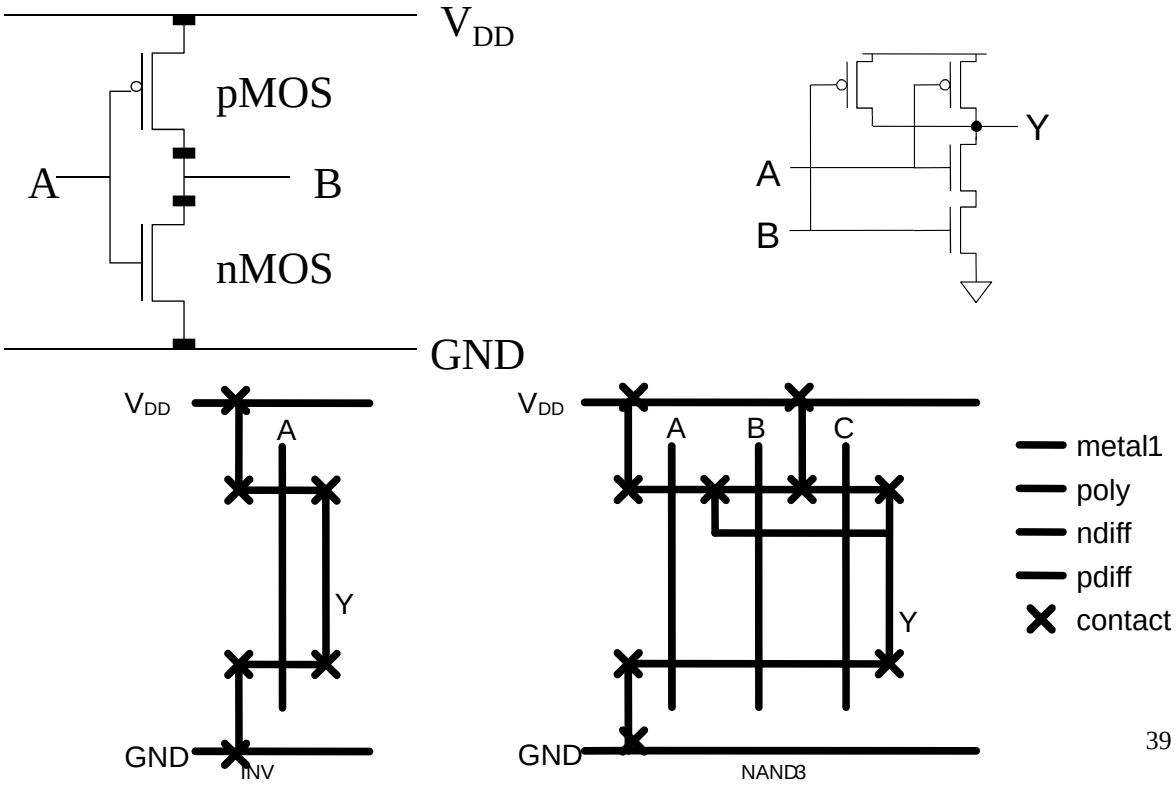
- Basic layout



- More area efficient layout



CMOS Inverter Stick Diagrams



Design Rules

- Interface between designer and process engineer
- Guidelines for constructing process masks
- Unit dimension: Minimum line width
  - –scalable design rules: lambda parameter
  - –absolute dimensions (micron rules)

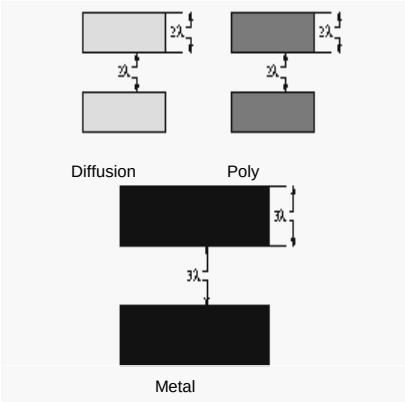
Design Rules

The constraints imposed on the geometry of a circuit layout to guarantee that the circuit can be fabricated with an acceptable yield

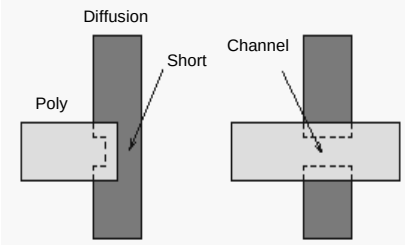
Diffusion Region Width	2λ
Polysilicon Region Width	2λ
Diffusion-Diffusion Spacing	3λ
Poly-Poly Spacing	2λ
Polysilicon Gate Extension	2λ
Contact Extension	λ
Metal Width	3λ

λ is the maximum misalignment of a feature from its intended position in the wafer due to over-etching, over or underexposure, distortion, ...

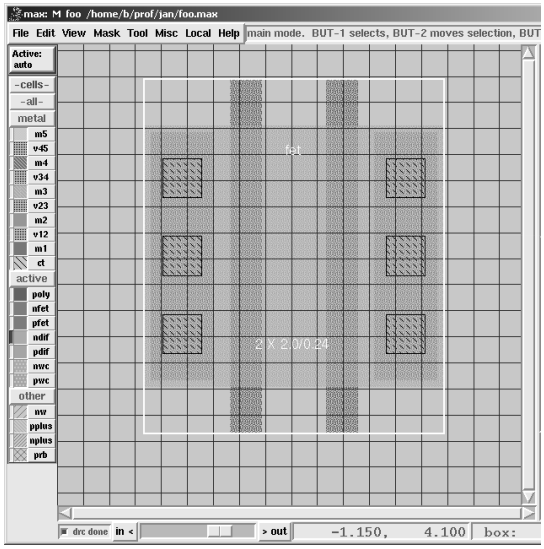
Size and Separation Rules



Incorrectly and Correctly Formed Channels



Layout Editor



Magic

http://opencircuitdesign.com/magic/

http://vlsi.cornell.edu/magic/

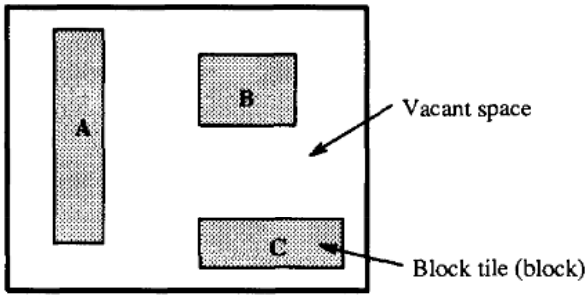
http://esaki.ee.boun.edu.tr/courses/ee537/magic-tut.pdf

Layout Editor

• Allows designer to create and edit VLSI layout – semiautomatic features

• What data structure to use?

• How to represent a layout ?



- Each feature is a rectangular region
- partition the region into a collection of tiles
- elements of a layout : block tiles/blocks
  - p-diff, n-diff, poly, metal ….
- area with no block- vacant space, vacant tiles

Operations on Layout Editor

Basic operations to be performed on layout

• Point finding : given coordinate of point p(x,y), find if it lies in a block, if yes, identify block

•Neighbour finding: find all blocks touching a given block

•Block visibility: find all bocks visible from block B in x and y direction

• Area searching: Given a block with fixed area, upper-left corner, width, height, find if it intersects with other block

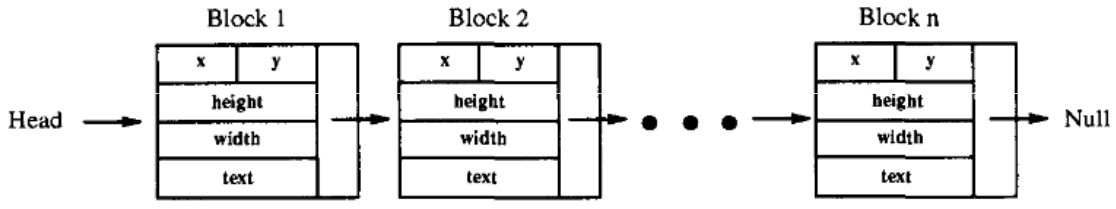
•Directed area enumeration: Given above for a block B, visit each block intersected by B in sorted order

Operations on Layout Editor

Block insertion: Insert a block in the layout without any overlap
Block Deletion: Delete a block from layout
Plowing: Given an area A & direction d, remove all blocks Bi from A by shifting each Bi in direction of d, while preserving order of blocks
Compaction: compress entire layout
Channel generation: determine vacant space in the layout and partition into tiles

How to store Blocks?

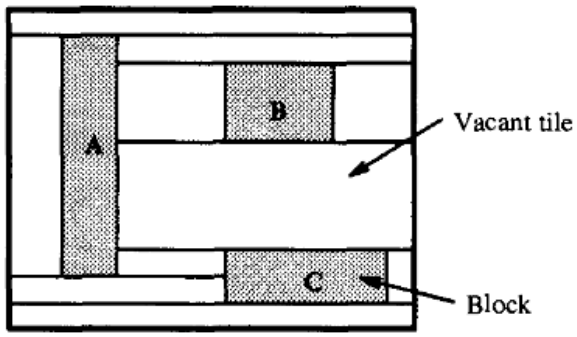
- Linked list L



- Find neighbour of B in L
  - Check each side of B if it shares the side with all the remaining blocks
  - O(n) complexity

How to store vacant spaces?

- Linked list L structure does not store vacant spaces of layout
- How to store vacant spaces?
- Convert vacant spaces into vacant tiles by extending upper & lower boundaries of each block horizontally
- Vacant tiles can also be stored as blocks in the linked list



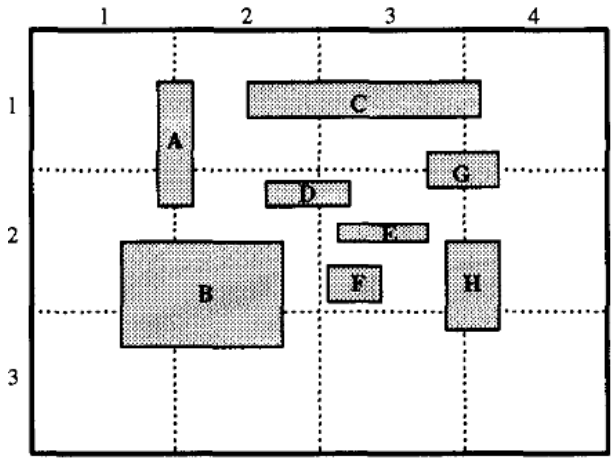
Data structures for layout

- Linked list of blocks
- Bin Based method
- Neighbour pointers
- Corner stiching

Bin –based Method

- Does not store vacant tiles
- A grid imposed on layout; creates bins
- Each bin maintains a linked list of blocks that intersect with this bin
- Space complexity : O(bn), b: # of bins, n: # blocks
- Find neighours of block B
- For all bins that contain

B, has to be searched for



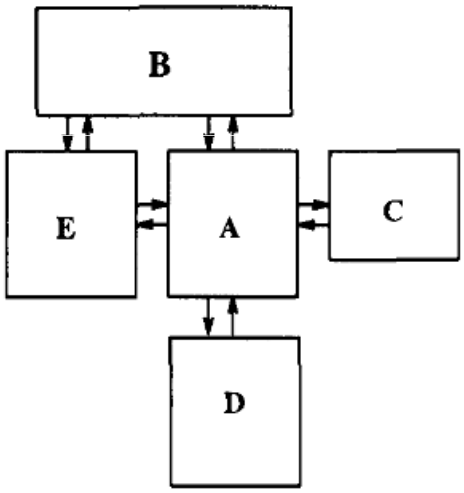
Neighbour-pointers

- Represents blocks by left-upper corner, length, width and pointers to all its neighbouring blocks
- Space complexity : $O(n^2)$
- Difficult to maintain
- Block insertion/deletion

take O(n) time

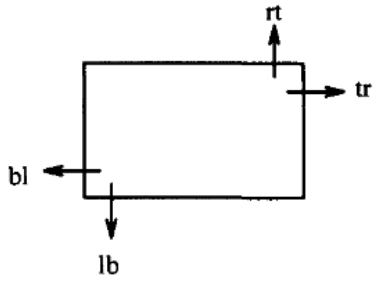
- Does not represent

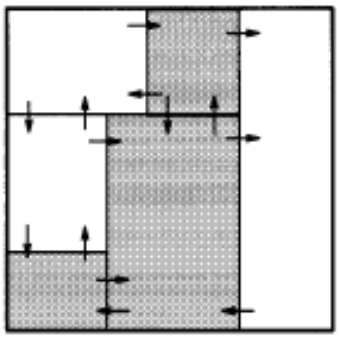
vacant tiles



Corner Stitch

- Best representation so far
- Both block & vacant tiles are stored
- Neighbour information is stored but pointer update easier than neighbour pointer data structure
- Most of the operations are O(n)
- Tiles are linked by set of pointers called corner stitches





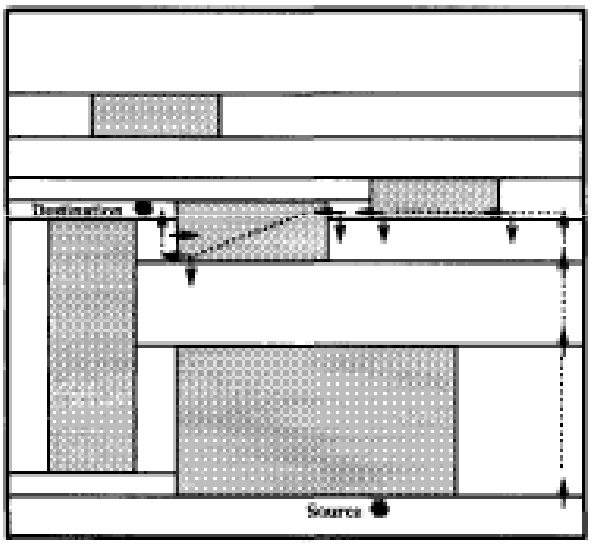
Operations using Corner Stitch

- Point finding
- Neighbour finding
- Area Search

Point finding using Corner Stitch

- Given a point p2, find the path through the corner stitches from current point p1 to p2, traversing minimum number of tiles.
- Move up/down (rt,lb) until we find tile whose vertical range contains the destination point
- Move left/right (tr,bl) until we find tile whose horizontal range contains the destination point
- Repeat steps 1& 2 whenever search goes out of the horizontal range of tile containing destination point

Point finding using Corner Stitch



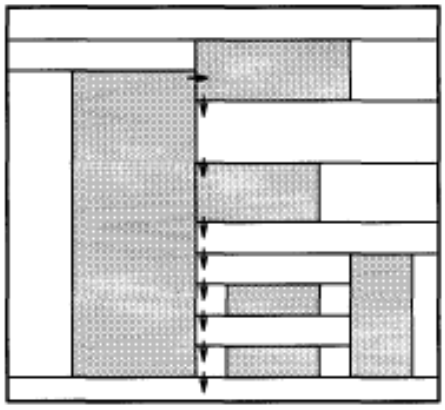
Point-find(B, x,y)

B: current block
x,y: the desired point

B is stored as :
Coordinate of left-bottom corner
Width, height

Neighbour finding using Corner Stitch

- Find all tiles that touch a given side of a given tile
- From tr ptr of current tile, start traversing using lb ptr downward until a tile is found which dose not lie in vertical range of current tile



Area search using Corner Stitch

- Given an area, report if there exists any block in the area
- Find tile where upper-left cornet of given area is located.(point finding)
- If tile is vacant tile then if its right edge is within given area, the adjacent tile must be block
- If tile is a block, search is complete. If no block is found, go to next tile touching right edge of given area by traversing lb ptr down and then traversing right using tr ptr
- Repeat steps2 and 3 until area has been searched or block has been found

Area search using Corner Stitch