IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 43: INTERCONNECT MODELING (PART 1)

PROF. INDRANIL SENGUPTA

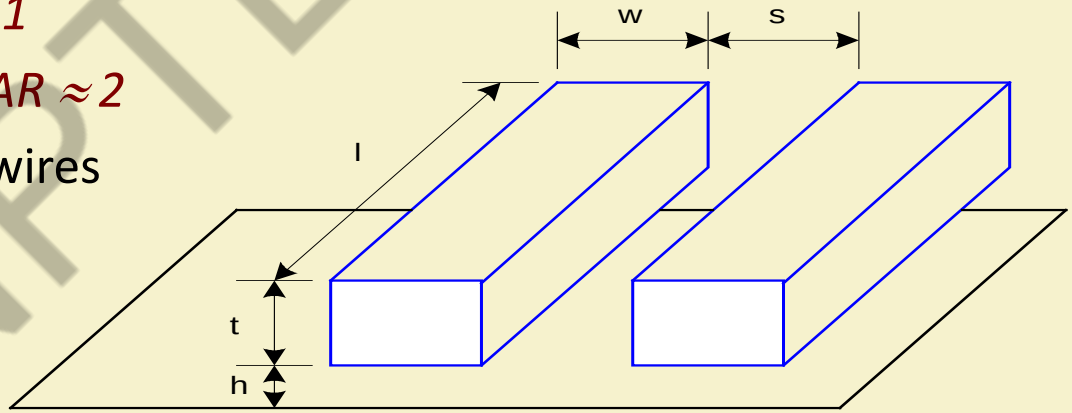
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Introduction

- Chips are mostly made of wires called *interconnect*.
  - In stick diagram, wires have set sizes.
  - Transistors are little things under the wires.
  - Many layers of wires exist (poly-silicon, diffusion, Metal1, Metal2, etc.).
- Wires are as important as transistors, as they determine:
  - Speed
  - Power
  - Noise
- Alternating layers run orthogonally.

Wire Geometry

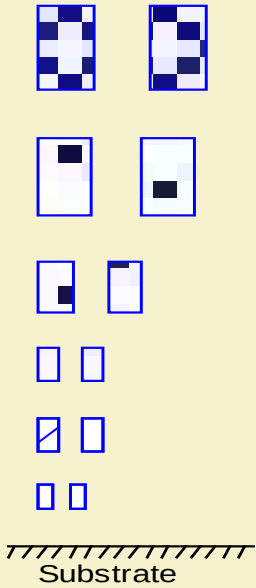
- Pitch = *w + s*
- Aspect ratio: *AR = t/w*
  - Old processes had *AR << 1*
  - Modern processes have *AR ≈ 2*
    - Pack in many skinny wires



Layer Stack

- Modern processes use 6-10+ metal layers
- Example:
  Intel 180 nm process
- M1: thin, narrow
  - High density cells
- M2-M4: thicker
  - For longer wires
- M5-M6: thickest
  - For V_{DD}, GND, clk

Layer	T (nm)	W (nm)	S (nm)	AR
6	1720	860	860	2.0
	1000			
5	1600	800	800	2.0
	1000			
4	1080	540	540	2.0
	700			
3	700	320	320	2.2
	700			
2	700	320	320	2.2
	700			
1	480	250	250	1.9
	800			

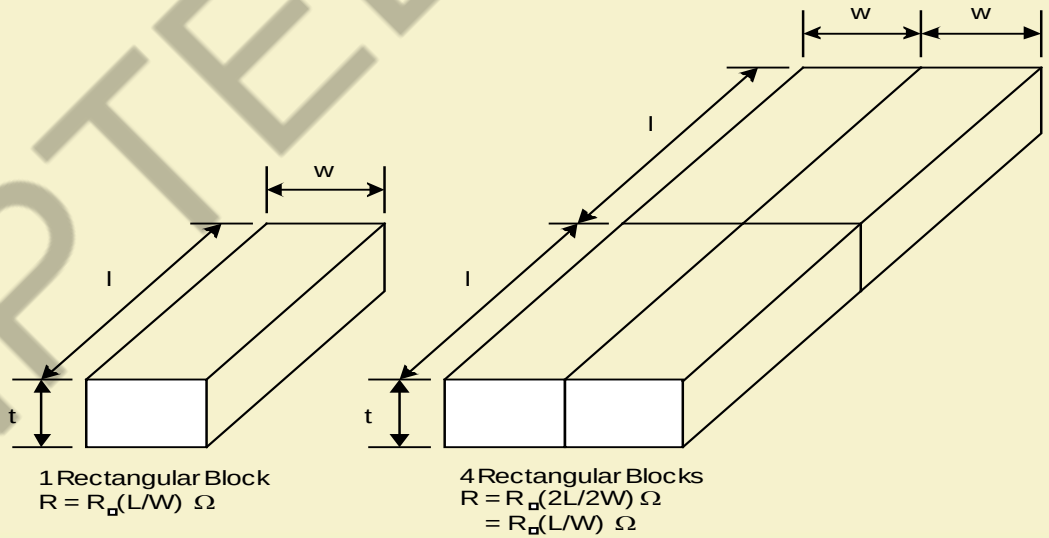


Wire Resistance

- ρ = *resistivity*

$$R = \frac{\rho}{t}\frac{l}{w} = R_{\square}\frac{l}{w}$$

- $R_{\square}$ = *sheet resistance* (Ω/□)
  - □ is a dimensionless unit(!)
- Count number of squares
  - $R = R_{\square}$ * (# of squares)



Choice of Metals

- Until 180 nm generation, most wires were Al.
- Modern processes often use Cu.
  - Cu atoms diffuse into silicon and damage FETs.
  - Must be surrounded by a diffusion barrier.

Metal	Bulk resistivity (μΩ*cm)
Silver (Ag)	1.6
Copper (Cu)	1.7
Gold (Au)	2.2
Aluminum (Al)	2.8
Tungsten (W)	5.3
Molybdenum (Mo)	5.3

Typical Sheet Resistance in 180 nm process

Layer	Sheet Resistance (Ω/□)
Diffusion (silicided)	3-10
Diffusion (no silicide)	50-200
Polysilicon (silicided)	3-10
Polysilicon (no silicide)	50-400
Metal1	0.08
Metal2	0.05
Metal3	0.05
Metal4	0.03
Metal5	0.02
Metal6	0.02

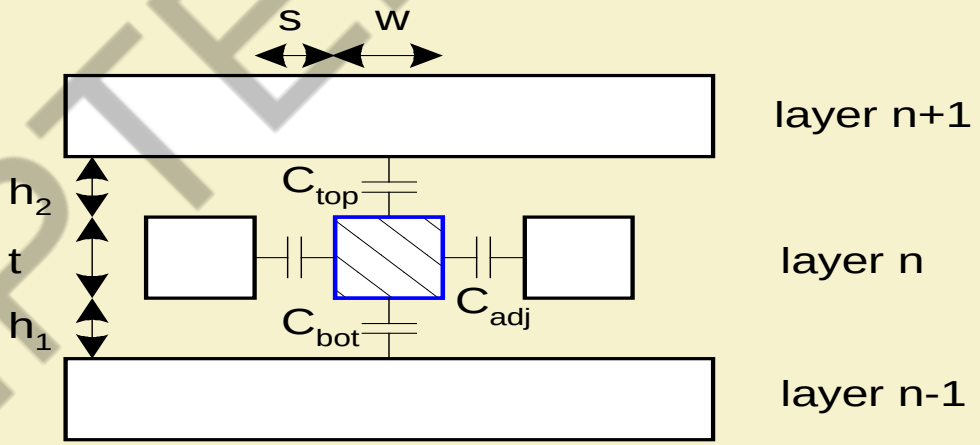
Contacts Resistance

- Contacts and vias also have 2-20 Ω resistance.
- Use many contacts for lowering R.
  - Many small contacts for current crowding around periphery.

Wire Capacitance

- Wire has capacitance per unit length.
  - To neighbors
  - To layers above and below

$$C_{total} = C_{top} + C_{bot} + 2C_{adj}$$

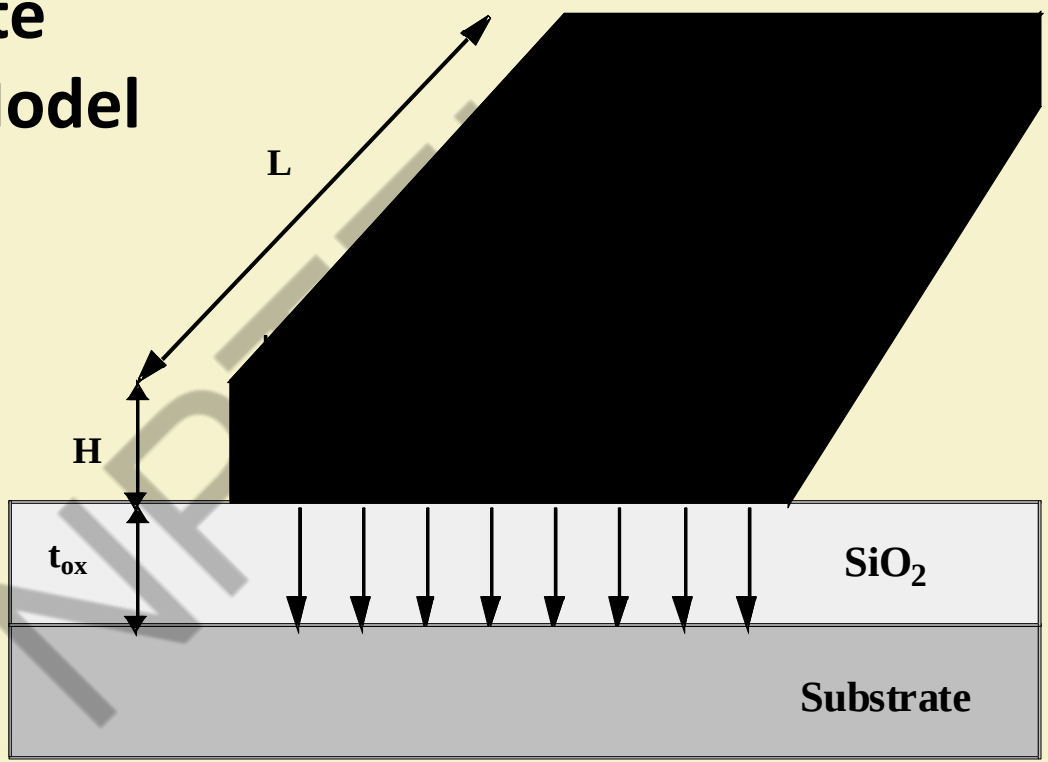


Capacitance Trends

- Parallel plate equation: $C = \varepsilon A/d$
  - Wires are not parallel plates, but obey trends.
  - Increasing area (W, t) increases capacitance.
  - Increasing distance (s, h) decreases capacitance.
- Dielectric constant $\varepsilon = k\ \varepsilon_0$

  where $\varepsilon_0 = 8.85 \times 10^{-14}$ F/cm, k = 3.9 for SiO_2
- Processes are starting to use low-k dielectrics.
  - $k \approx 3$ (or less) as dielectrics use air pockets.

Parallel Plate Capacitance Model



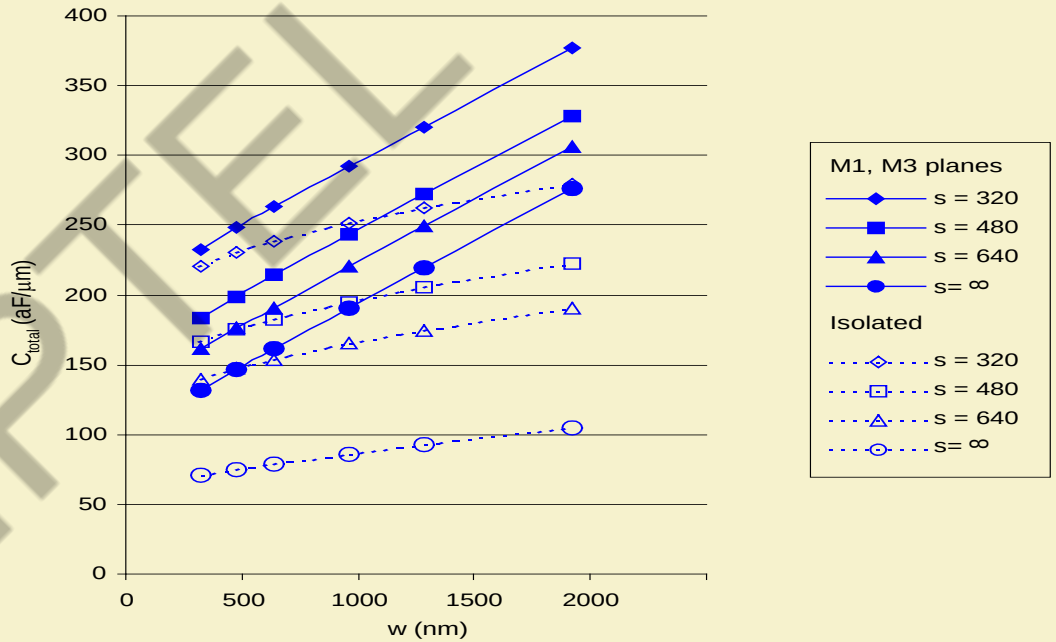
Typical Wiring Capacitance Values

Interconnect Layer	fF/μm^2
Polysilicon to Substrate	0.058 ± 0.004
Metal1 to Substrate	0.031 ± 0.001
Metal2 to Substrate	0.015 ± 0.001
Metal3 to Substrate	0.010 ± 0.001
N+ Diffusion to Substrate	0.36 ± 0.02
P+ Diffusion to Substrate	0.46 ± 0.06

For 1 μ CMOS

M2 Capacitance Data

- Typical wires have ~ 0.2 fF/μm
  - Compare to 2 fF/μm for gate capacitance



END OF LECTURE 43

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 44: INTERCONNECT MODELING (PART 2)

PROF. INDRANIL SENGUPTA

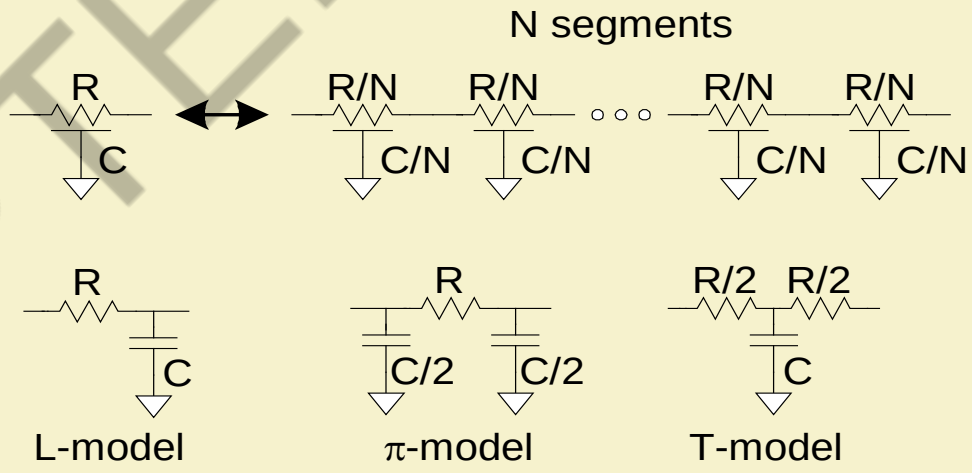
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Diffusion & Polysilicon

- Diffusion capacitance is very high (about 2 fF/μm).
  - Comparable to gate capacitance.
  - Diffusion also has high resistance.
  - Avoid using diffusion *runners* for wires!
- Polysilicon has lower C but high R.
  - Use for transistor gates.
  - Occasionally for very short wires between gates.

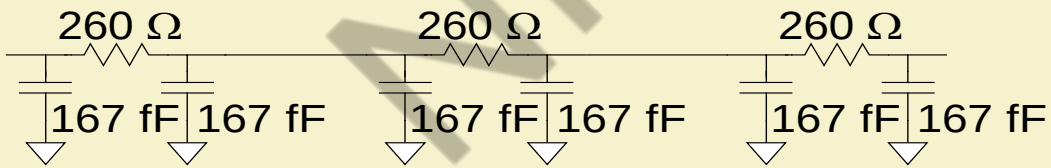
Lumped Element Models

- Wires are a distributed system
  - Approximate with lumped element models.
- 3-segment π-model is accurate to 3% in simulation.
- L-model needs 100 segments for same accuracy!
- Use single segment π-model for Elmore delay.



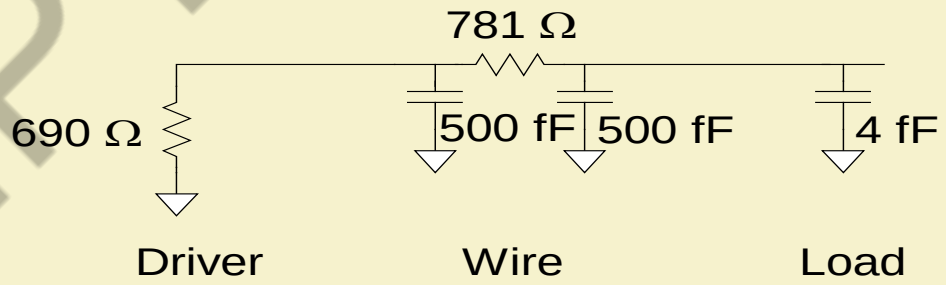
Example

- Metal2 wire in 180 nm process
  - 5 mm long
  - 0.32 μm wide
- Construct a 3-segment π-model
  - $R_{\square}$ = 0.05 Ω/□ => R = 781 Ω
  - $C_{permicron}$ = 0.2 fF/μm => C = 1 pF



Wire RC Delay

- Estimate the delay of a 10x inverter driving a 2x inverter at the end of the 5mm wire from the previous example.
  - R = 2.5 kΩ*μm for gates
  - Unit inverter: 0.36 μm nMOS, 0.72 μm pMOS
  - t_{pd} = 1.1 ns

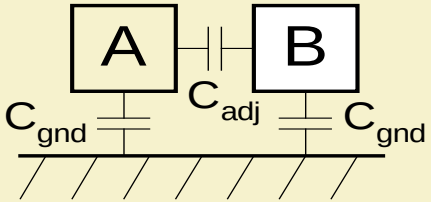


Crosstalk

- A capacitor does not like to change its voltage instantaneously.
- A wire has high capacitance to its neighbor.
  - When the neighbor switches from 1→0 or 0→1, the wire tends to switch too.
  - Called capacitive *coupling* or *crosstalk*.
- Crosstalk effects:
  - Noise on non-switching wires.
  - Increased delay on switching wires.

Crosstalk Delay

- Assume layers above and below on average are quiet.
  - Second terminal of capacitor can be ignored.
  - Model as $C_{gnd} = C_{top} + C_{bot}$
- Effective C_{adj} depends on behavior of neighbors.
  - *Miller effect*

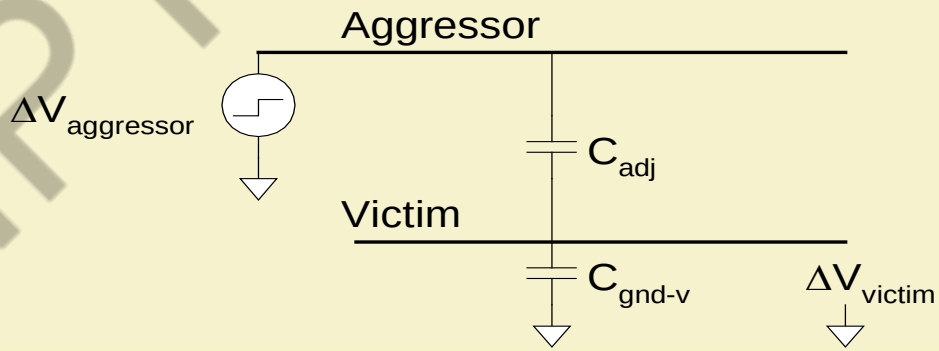


B	ΔV	$C_{eff(A)}$	MCF
Constant	V_{DD}	$C_{gnd} + C_{adj}$	1
Switching with A	0	C_{gnd}	0
Switching opposite A	$2V_{DD}$	$C_{gnd} + 2\,C_{adj}$	2

Crosstalk Noise

- Crosstalk causes noise on non-switching wires.
- If victim is floating:
  - Model as capacitive voltage divider.

$$\Delta V_{victim} = \frac{C_{adj}}{C_{gnd-v} + C_{adj}} \Delta V_{aggressor}$$

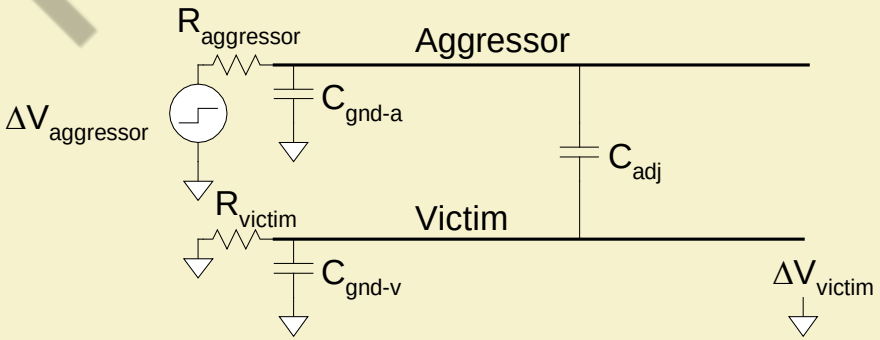


Driven Victims

- Usually victim is driven by a gate that fights noise.
  - Noise depends on relative resistances.
  - Victim driver is in linear region, aggressor in saturation.
  - If sizes are same, $R_{aggressor}$ = 2-4 x R_{victim}

$$\Delta V_{victim} = \frac{C_{adj}}{C_{gnd-v} + C_{adj}} \frac{1}{1+k} \Delta V_{aggressor}$$

$$k = \frac{\tau_{aggressor}}{\tau_{victim}} = \frac{R_{aggressor}\left(C_{gnd-a} + C_{adj}\right)}{R_{victim}\left(C_{gnd-v} + C_{adj}\right)}$$

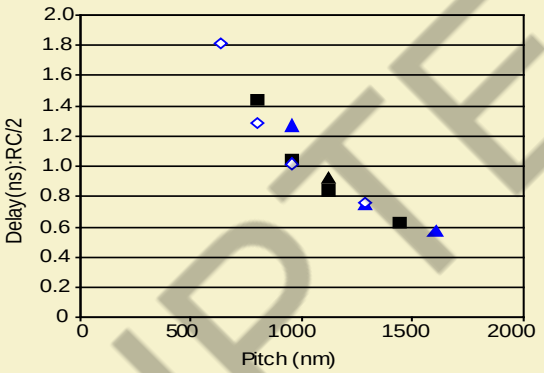


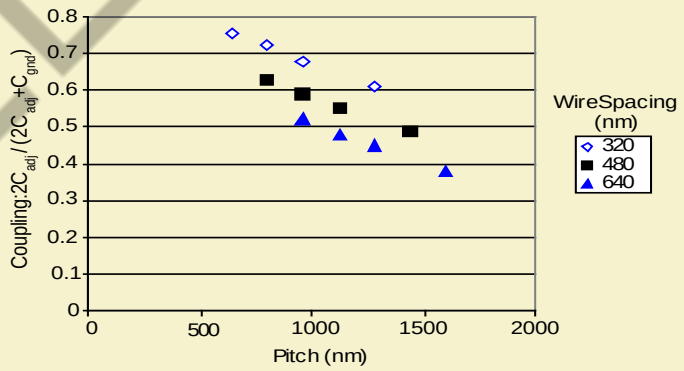
Noise Implications

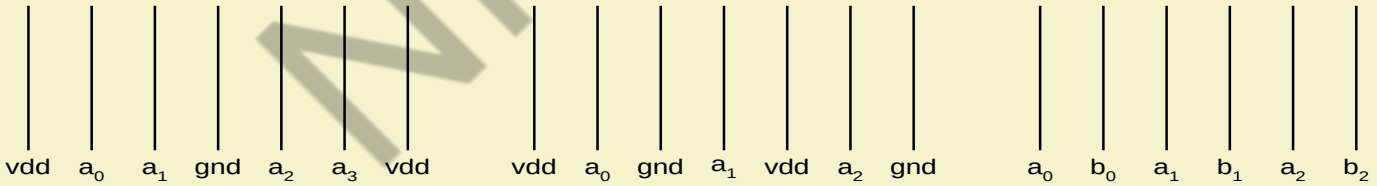
- If the noise is less than the noise margin, nothing happens.
- Static CMOS logic will eventually settle to correct output even if disturbed by large noise spikes.
  - But glitches cause extra delay.
  - Also cause extra power from false transitions.
- Dynamic logic never recovers from glitches.
- Memories and other sensitive circuits also can produce the wrong answer.

Wire Engineering

- Goal: achieve delay, area, power goals with acceptable noise
- Degrees of freedom:
  - Width
  - Spacing
  - Layer
  - Shielding

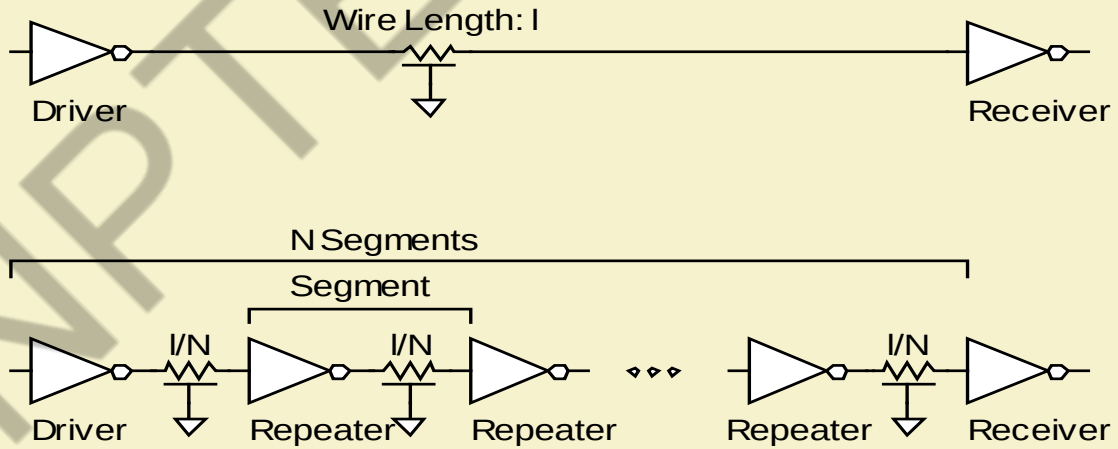






Repeaters

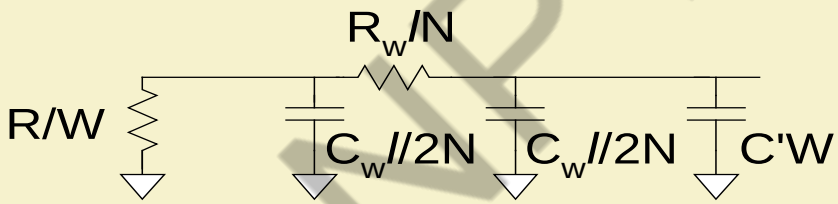
- R and C are proportional to *l*.
- RC delay is proportional to l^2.
  - Unacceptably great for long wires.
- Break long wires into N shorter segments.
  - Drive each one with an inverter or buffer.



Repeater Design

- How many repeaters should we use?
- How large should each one be?
- Equivalent Circuit
  - Wire length l/N
    - Wire Capaitance = C_w* l/N, Resistance = R_w* l/N
  - Inverter width W (nMOS = W, pMOS = 2W)
    - Gate Capacitance = C'*W, Resistance = R/W

- Equivalent Circuit
  - Wire length *l*
    - Wire Capacitance = C_w* l, Resistance = R_w* l
  - Inverter width W (nMOS = W, pMOS = 2W)
    - Gate Capacitance = C’ *W, Resistance = R/W



END OF LECTURE 44

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 45: DESIGN RULE CHECK

PROF. INDRANIL SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Introduction

- In VLSI design, as processes become more complex, it becomes very difficult for the designer to understand the intricacies of the fabrication process and identify errors (if any) in the photo masks.
- ***Solution***: Define a set of design rules.
  - Act as interface between the circuit designer and the process engineer during the manufacturing phase.
  - Design rules can be either conservative or aggressive, depending upon whether yield or performance is the main goal.

- Basically, the design rules specify certain geometric constraints on the layout artwork.
  - Ensures that the patterns on the processed wafers will preserve the topology and geometry of the designs.
  - Minimum-width and minimum-spacing constraints.
  - Between objects on the same or different layers.

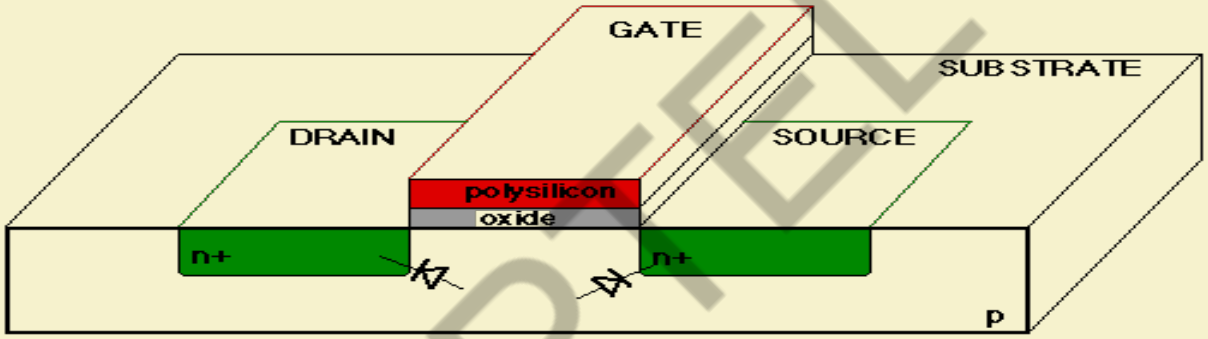
Scalable Design Rules

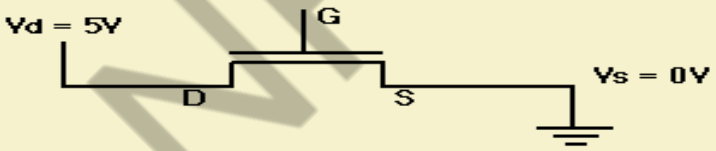
- All rules are defined in terms of a single parameter λ.
  - Also called lambda-based design rules.
- The rules are so chosen that a design can be easily ported over a series of fabrication processes, as technology scales with time.
- For deep submicron designs, some companies also use absolute design rules as scaling always does not work.

How are Layouts Represented?

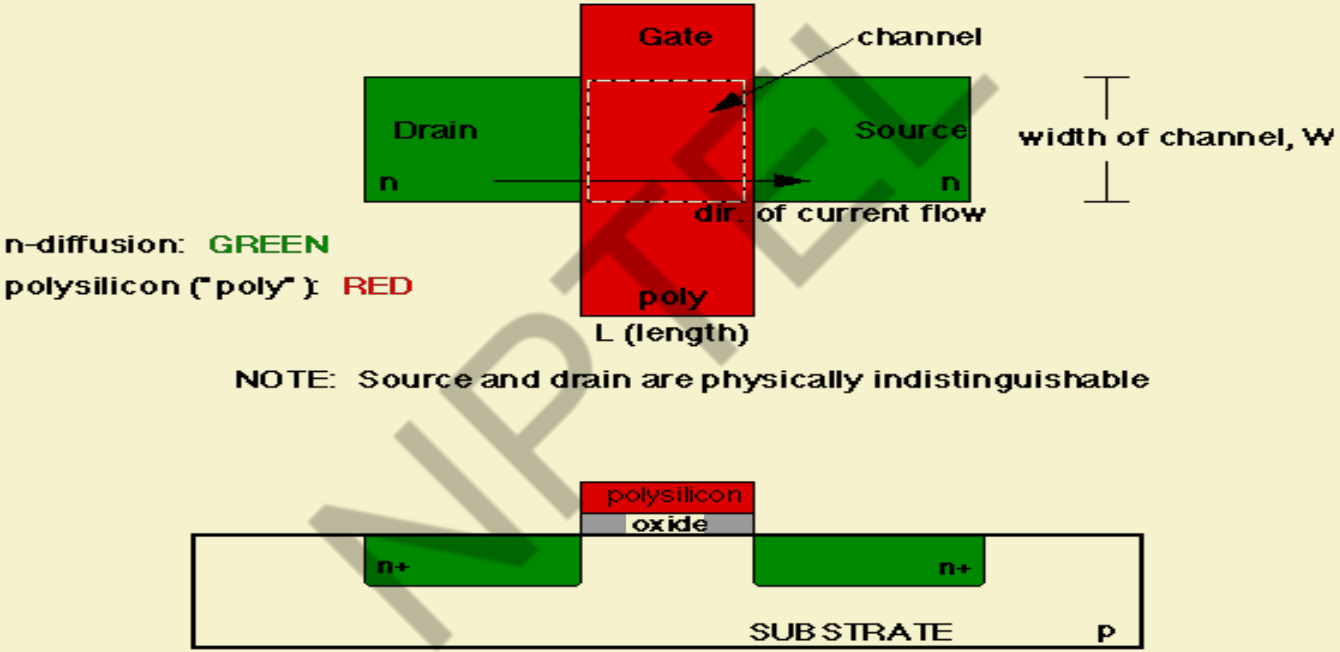
- Symbolic layout
  - Schematic drawing showing rectangles corresponding to objects on various layers.
  - Use color coding convention → Every layer is represented by a well-defined color.
- Stick diagrams
  - A compact representation, where every object is shown as a line.
  - Design rules determine the minimum width and spacing between the lines.

n-channel MOS Transistor

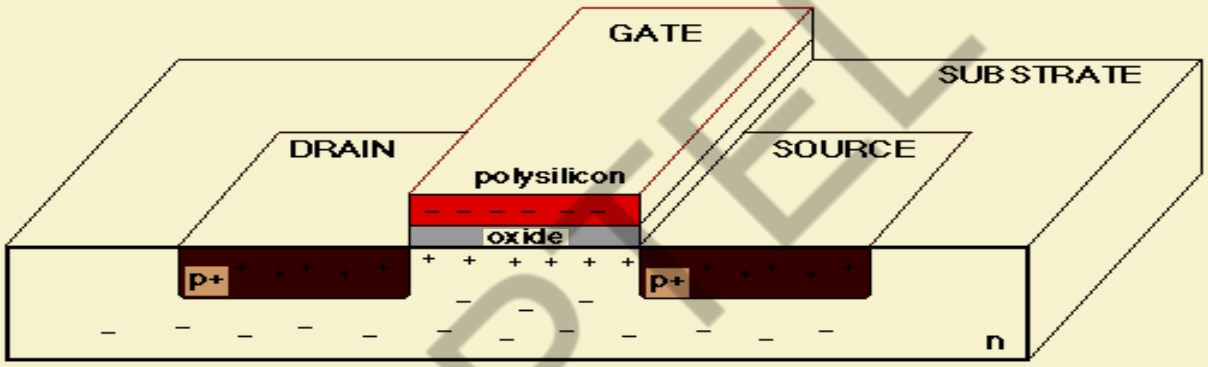


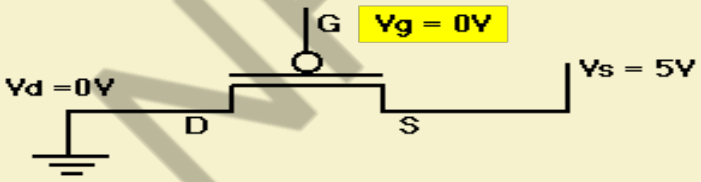


n-channel Transistor Layout

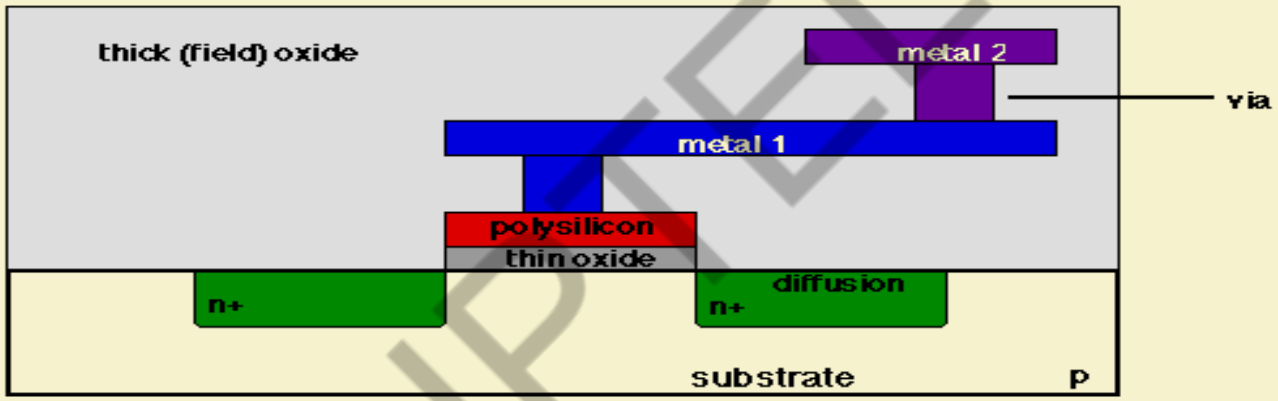


p-channel MOS Transistor





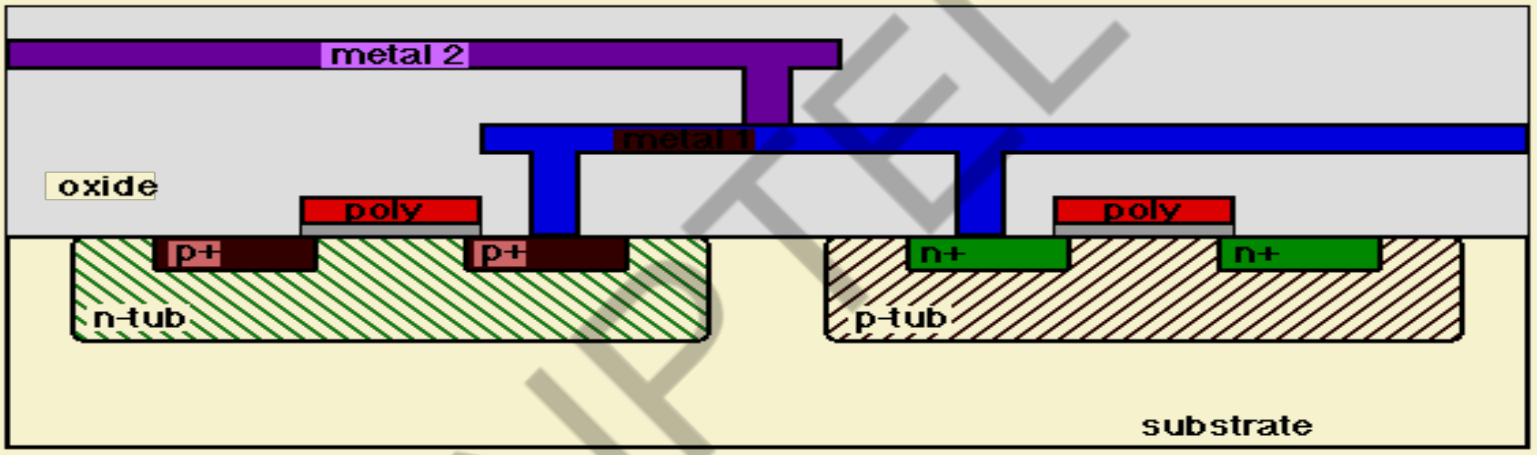
Fabrication Layers



Color Convention

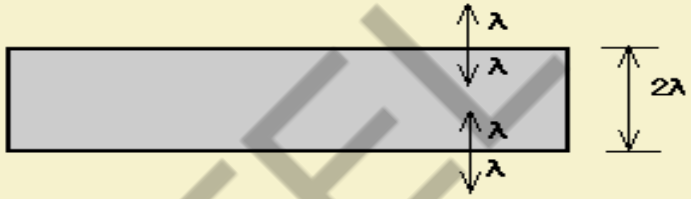
- Poly-silicon **RED**
- N-diffusion **GREEN**
- P-diffusion **BROWN**
- Metal-1 **BLUE**
- Metal-2 **PURPLE**
- Via’s **BLACK**

CMOS Fabrication



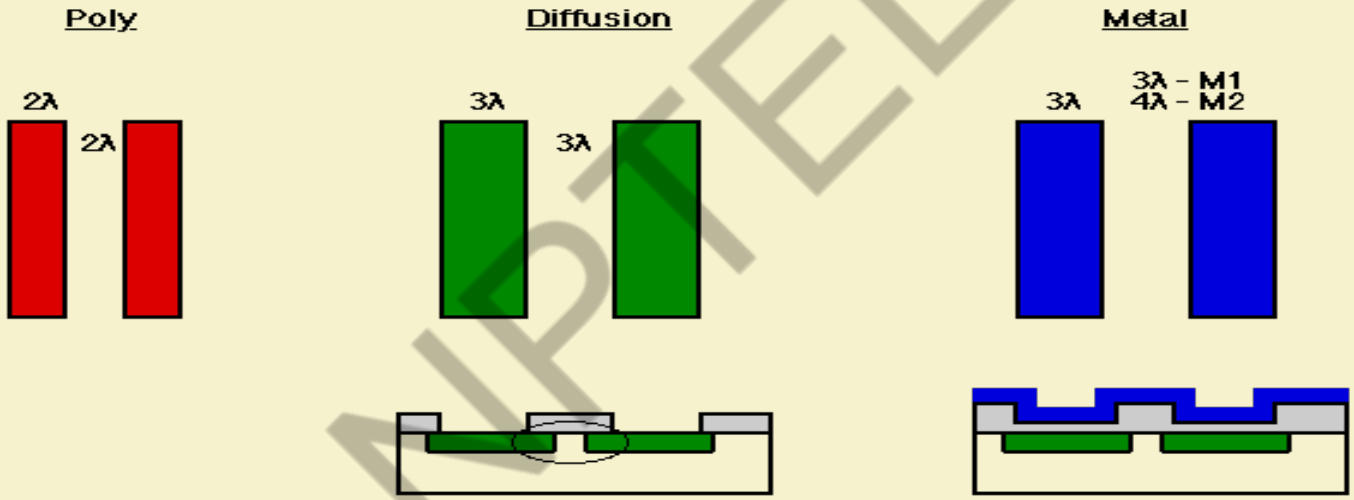
General Design Rules

1. Feature Width



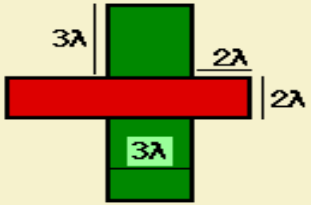
2. Feature Spacing

Width/Spacing Rules (MOSIS)

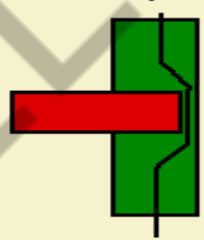


Poly-Diffusion Interaction

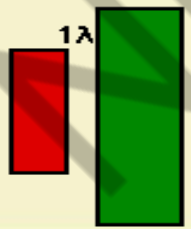
Transistors



Catastrophic error



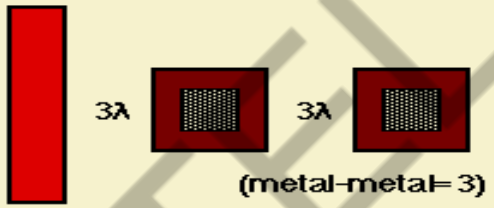
Unrelated Poly & Diffusion



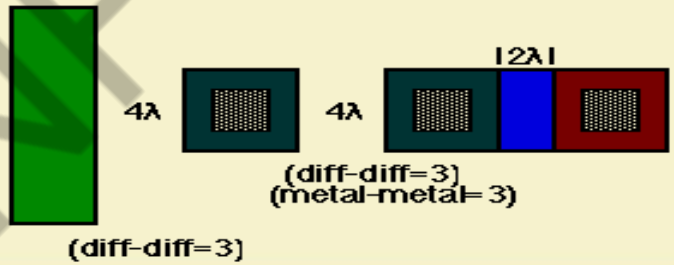
Thinner diffusion, but still working.

Contact Spacing

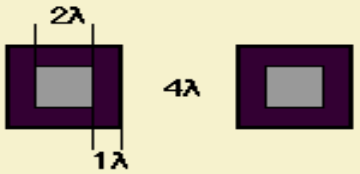
Poly contact



Diffusion contact

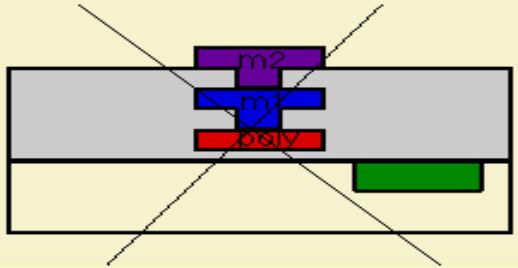


M2 Contact (Via)



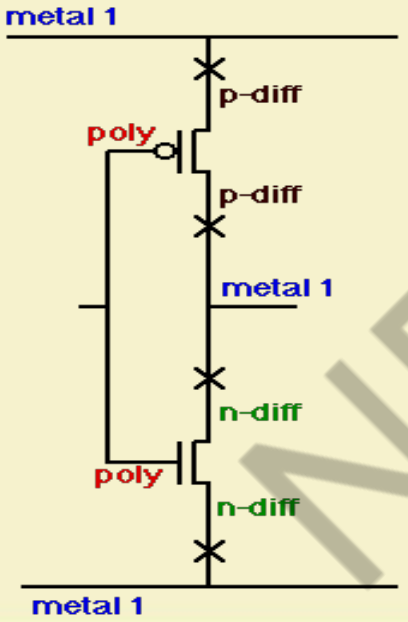
NOTE: M2 can connect only to M1

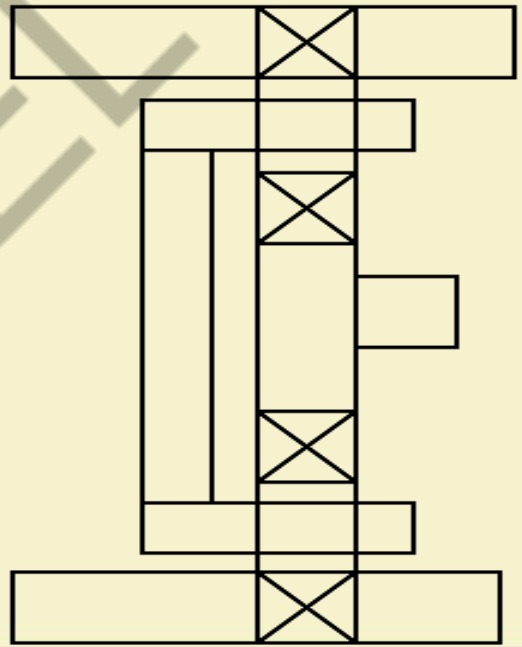
- No via- contact overlap



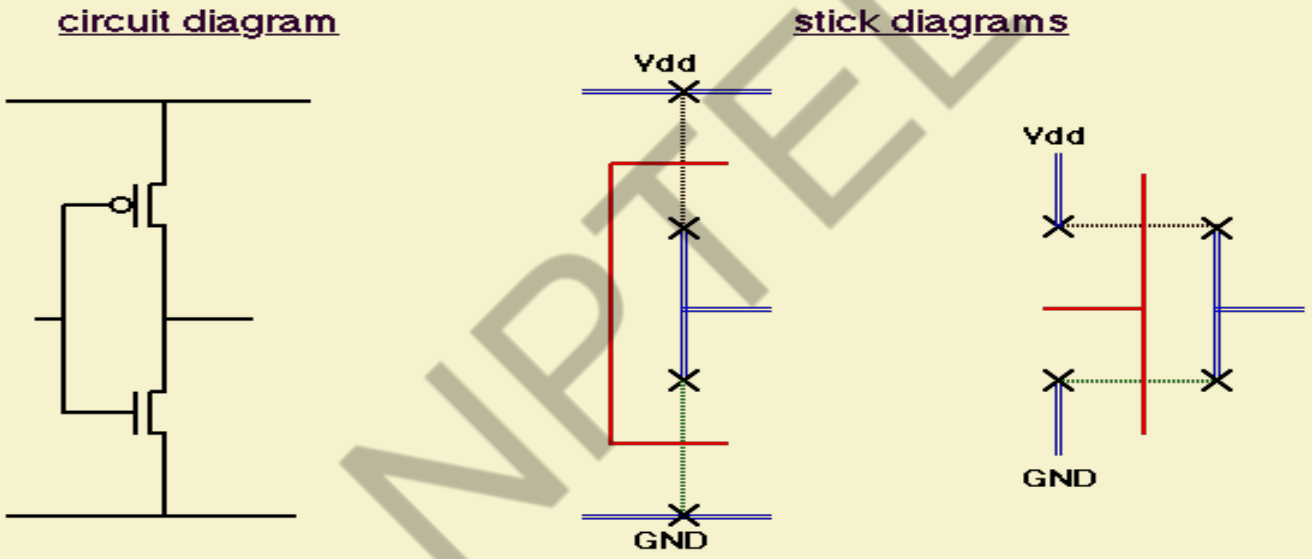
Not allowed in MOSIS rules

CMOS Layout Example

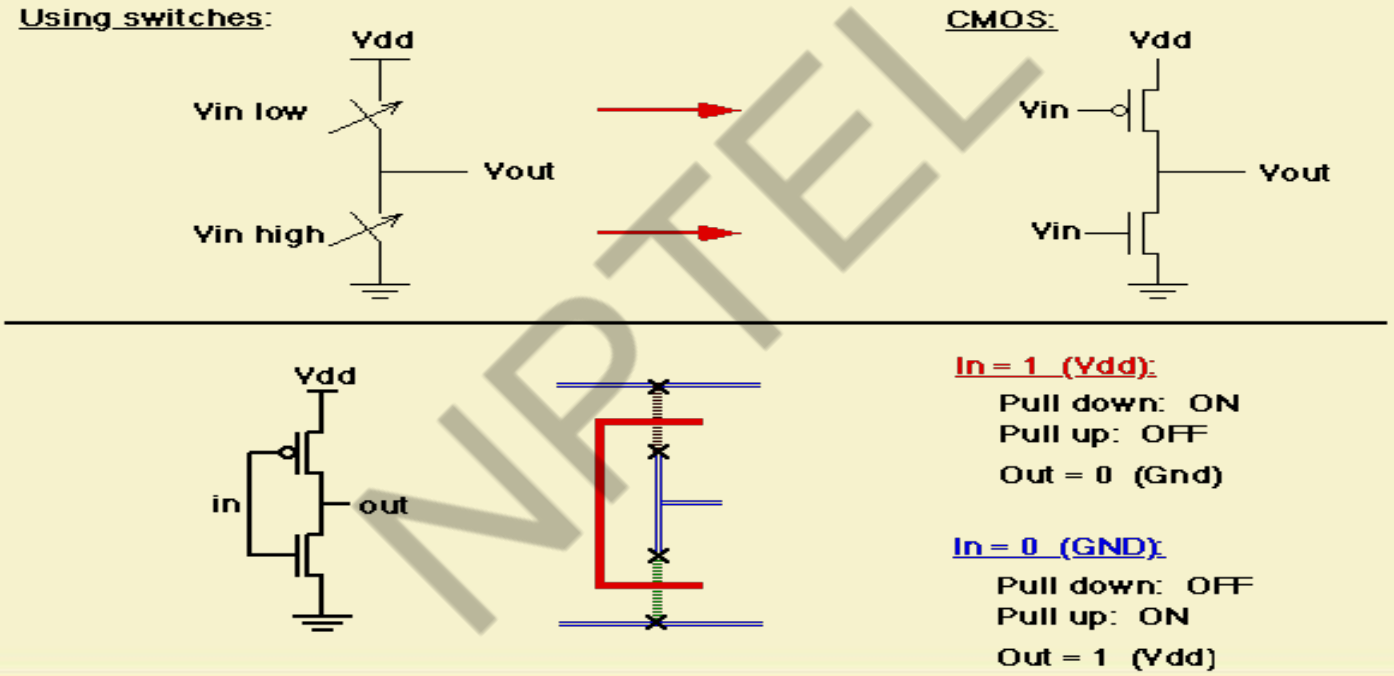




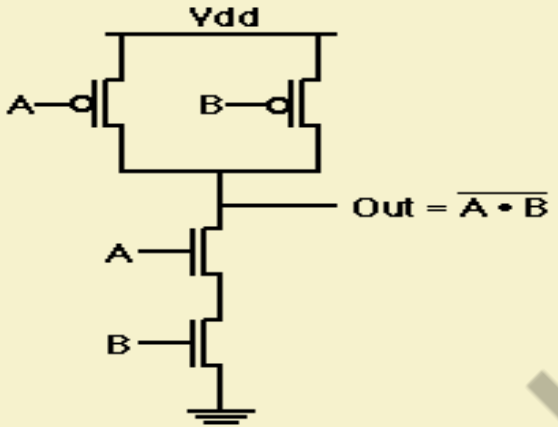
Stick Diagrams



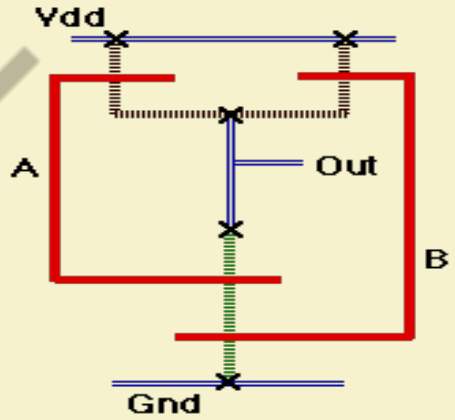
Static CMOS Inverter



Static CMOS NAND Gate



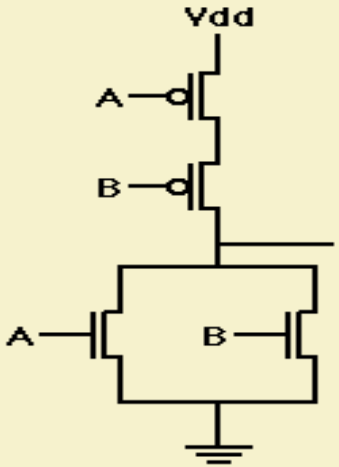
A	B	$\overline{A \bullet B}$
0	0	1
0	1	1
1	0	1
1	1	0



1. Pull-down: Connect to ground If A=1 AND B=1

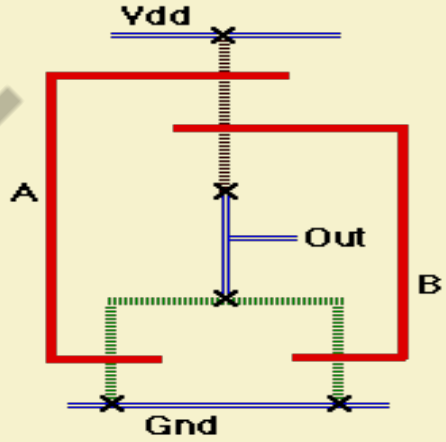
2. Pull-up: Connect to Vdd If A=0 OR B=0

Static CMOS NOR Gate



Out = $\overline{A+B}$

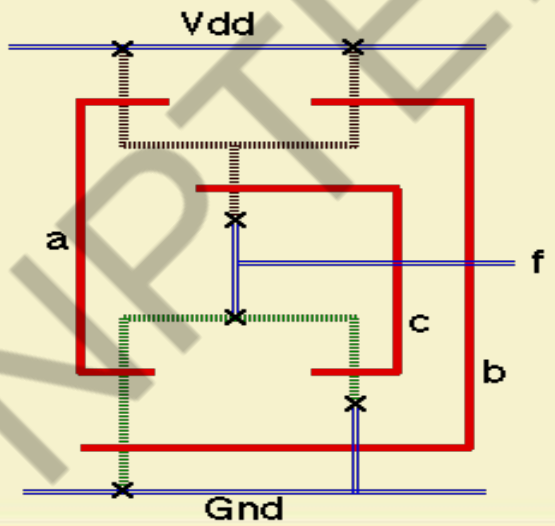
A	B	$\overline{A+B}$
0	0	1
0	1	0
1	0	0
1	1	0



1. Pull-down: Connect to ground If A=1 OR B=1
2. Pull-up: Connect to Vdd If A=0 AND B=0

Static CMOS Design Example Layout

Example: $f = \overline{a \cdot b + c}$



END OF LECTURE 45

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 46: LAYOUT COMPACTION (PART 1)

PROF. INDRANIL SENGUPTA

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Layout Compaction

- A compactor generates layout at the mask level. It attempts to make the layout as dense as possible.
- Applications of compaction:
  - *Area minimization*: remove redundant space in layout at the mask level.
  - *Layout compilation*: generate mask-level layout from symbolic layout.
  - *Redesign*: automatically remove design-rule violations.
  - *Rescaling*: convert mask-level layout from one technology to another.

Features

- It involves compaction of:
  - Space between the features.
  - Size of the features.
  - Shape of the features.
- Accepts symbolic layout as the input, and generates the final layout as output.

Problem Formulation

- Given:
  - A set of geometric features $M = \{M_1, M_2, ..., M_n\}$.
  - The minimum feature size, $s(M_i)$, for all i.
  - The minimum separation between features M_i and M_j, denoted as $d(M_i,M_j)$.
- Objective:
  - Minimize the layout such that

$$size(M_i) \geq s(M_i)$$

$$dist(M_i,M_j) \geq d(M_i,M_j)$$

where $size(M_i)$ and $dist(M_i,M_j)$ are size of M_i and distance between M_i and M_j after the compaction, where $1 \leq i,j \leq n$.

Design Style Specific Issues

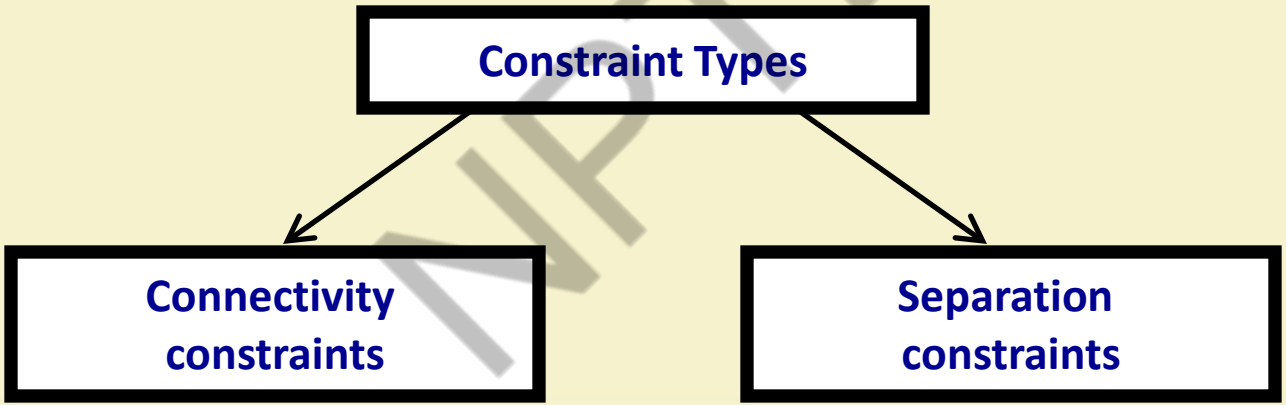
- Full-custom Design Style
  - Compaction is very critical.
  - After placement and routing, a large amount of space is left vacant.
- Standard Cell Design Style
  - Since the heights of the cells are fixed, the height of the layout can be minimized by minimizing channel height.
  - A restricted type of compaction, called *channel compaction*, is used.
- Gate Array Design Style
  - Compaction is not applicable as the position of the gates is fixed.

Compaction Algorithms

- Based on minimum distance between features
  a) Constraint graph based
  b) Virtual grid based
- Based of direction of movement of features
  a) 1-D compaction
  b) 1½-D compaction
  c) 2-D compaction

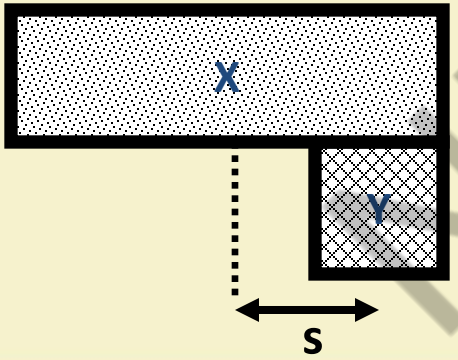
Constraint Graph Based Compaction

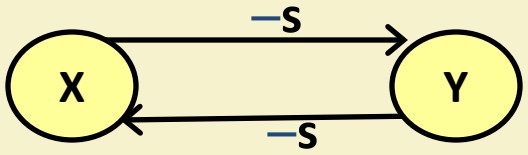
- Constraint graph $G = (V,E)$
  - Each vertex $v \in V$ represents a component.
  - The set of edges E represents constraints.



(a) Connectivity Constraints

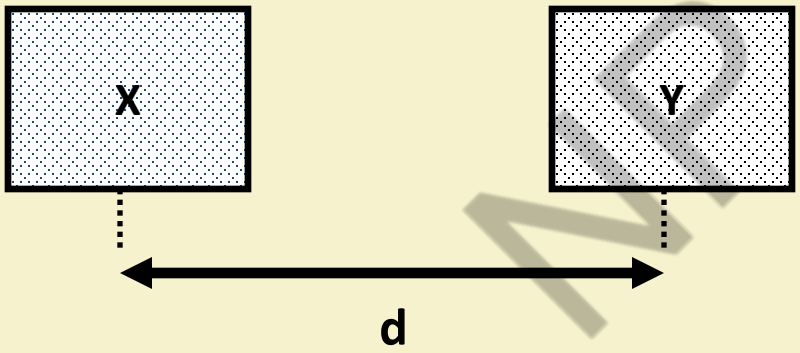
- If two features *X* and *Y* are required to be within a distance *s* of each other.
  - A physical connection can be represented in the graph as a pair of edges between *X* and *Y*, each with weight *–s*.

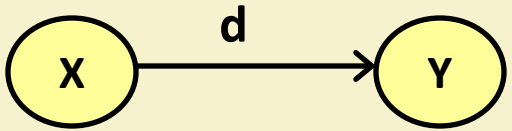




(b) Separation Constraints

- Two features *X* and *Y* are required to be at least *d* units apart from each other.
  - Represented as an edge from *X* to *Y* of weight *d*.



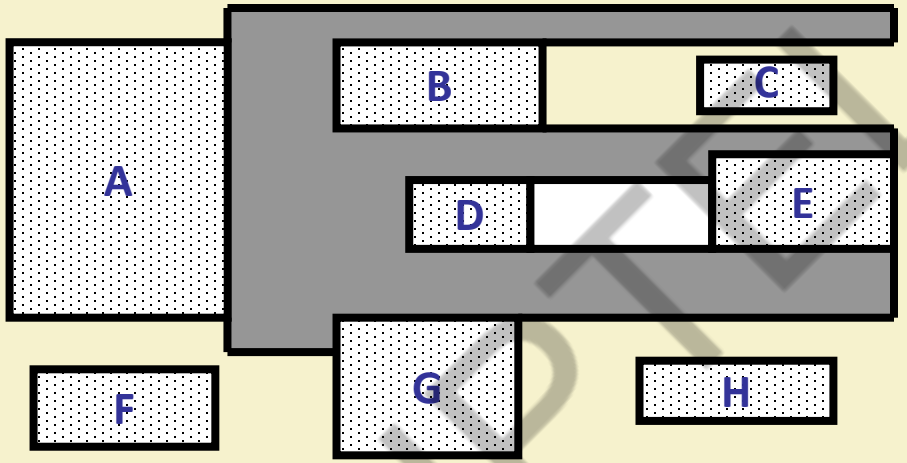


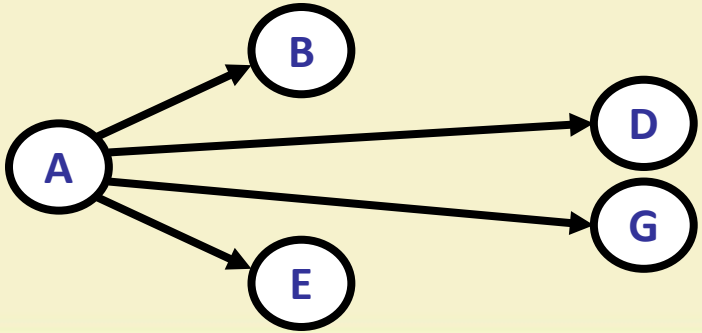
The Shadow Propagation Method

- One of the best known and widely used technique for *generating constraints*.
- Basic idea:
  - The shadow of a feature is propagated along the direction of compaction.
  - The shadow is caused by shining an imaginary light from behind the feature under consideration.
    - The shadow is normally extended on both sides of the feature to consider diagonal constraints.

- Whenever the feature is obstructed by another feature, an edge is added to the graph between corresponding vertices.
- The obstructed part is then removed.
- The process is continued until all of the shadow has been obstructed.
- Process repeated for each feature in the layout.

Example



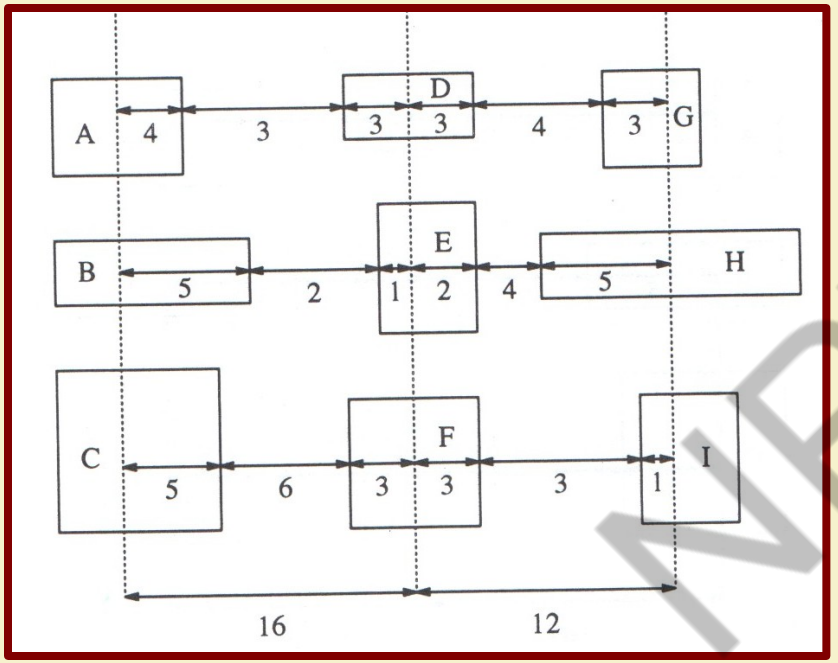


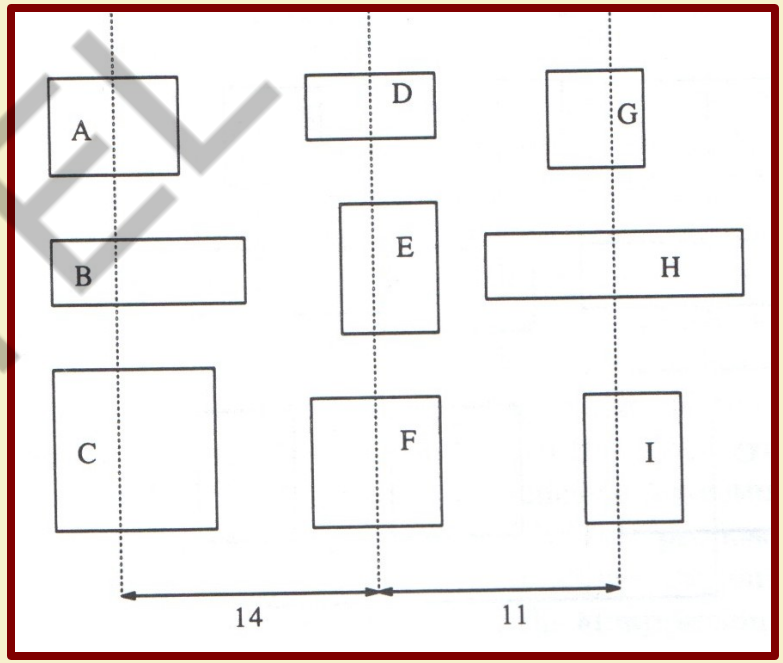
Virtual Grid Based Compaction

- This method assumes that the layout is to be drawn on a grid.
- Each component is considered attached to a grid line.
- The compaction operation compresses the grid along with all components placed on it keeping the grid lines straight along the way.
- The minimum distance between two adjacent grid lines depends on the components on these grid lines.
- X-compaction is followed by Y-compaction.

- Advantage:
  - Simple and easy to implement.
- Disadvantage:
  - Does not produce compact layouts as compared to the constraint graph method.

Example





END OF LECTURE 46

IIT KHARAGPUR

NPTEL ONLINE
CERTIFICATION COURSES

Lecture 47: LAYOUT COMPACTION (PART 2)

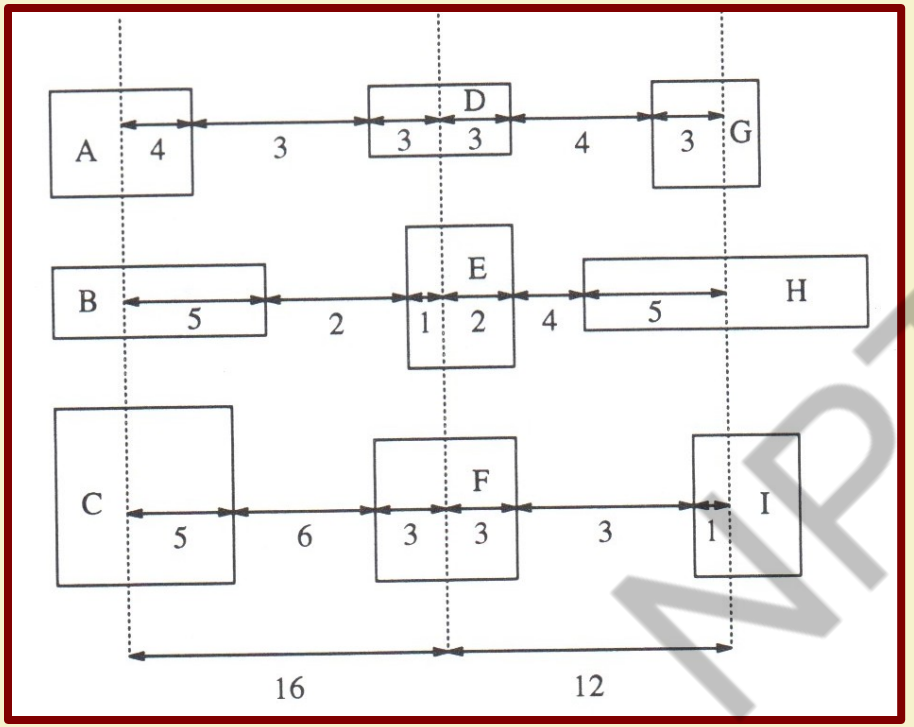
PROF. INDRANIL SENGUPTA

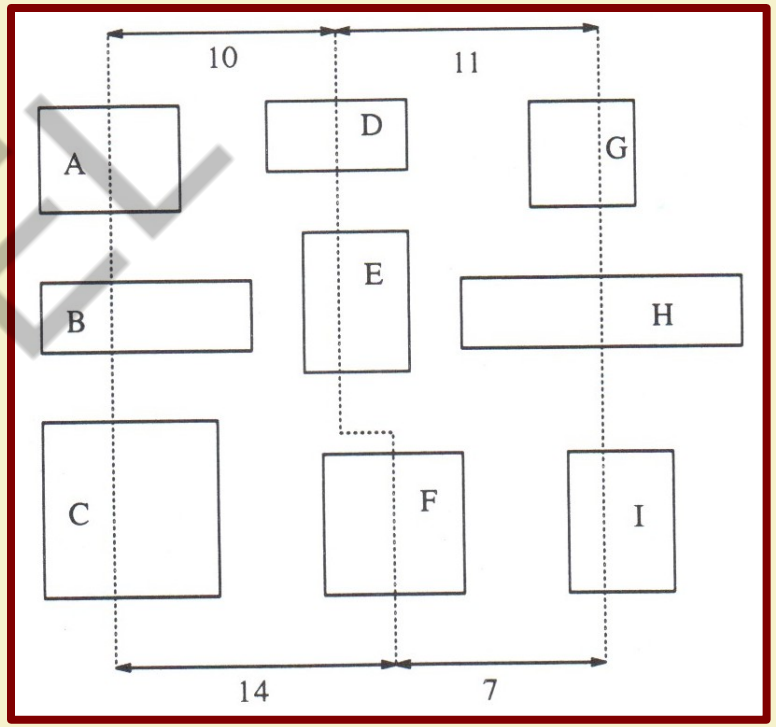
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Split Grid Compaction

- Here we place distinct circuit features that fall on the same virtual grid separately, splitting the virtual grids where necessary.
- Initially, the compactor identifies groups of circuit features falling on the same virtual grid lines that need to be placed together.
  - Local connectivity is used to identify the groups.
- After the circuit features are grouped together, the compaction is done in two passes:
  - First x-compaction, and then y-compaction.

- An example:
  - Shown on the next slide.
  - We group ***A***, ***B*** and ***C*** together on the first grid line.
  - On the second grid, we form two groups:
    - The first group consists of ***D*** and ***E***.
    - The second group consists only of ***F***.





Aspects of Compaction

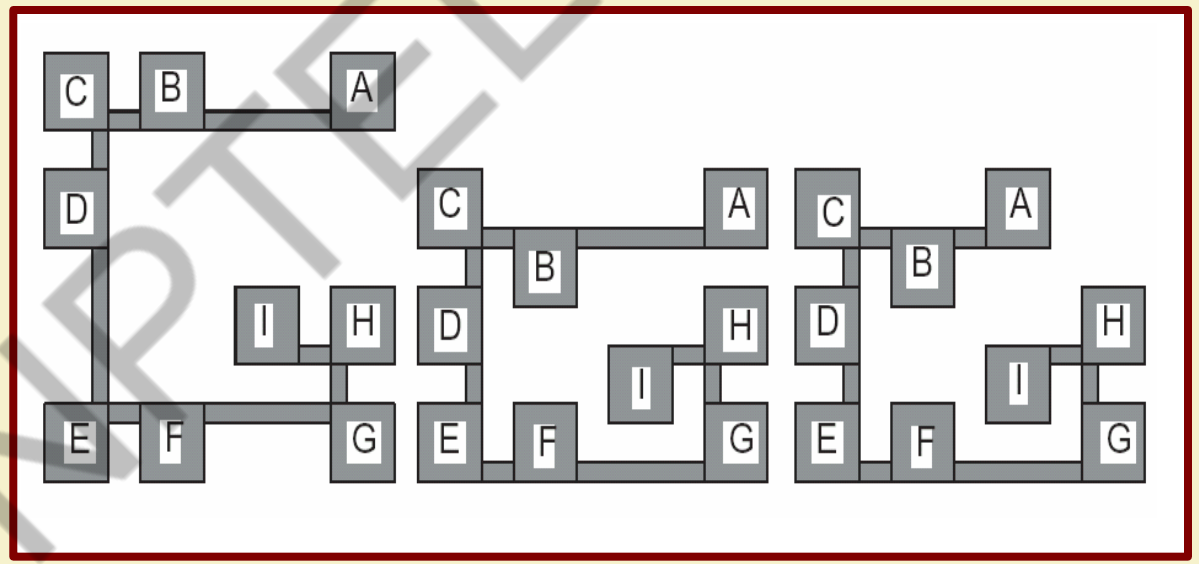
- Dimension:
  - In 1D compaction, layout elements are moved or shrunk only in one dimension (x or y direction).
    - Often performed first in the x-dimension and then in the y-dimension.
  - In 2D compaction, layout elements are moved and shrunk simultaneously in two dimensions.
- Complexity:
  - 1D compaction can be done on polynomial time.
  - 2D compaction is NP-hard.

1D Compaction: X followed by Y

- Each square is 2λ x 2λ, minimum separation is 1λ.
- Initially, layout area is 11λ x 11λ.
- After compacting along the x-direction, and then along the y-direction, layout area reduces to 8λ x 11λ.

1D Compaction: Y followed by X

- Each square is 2λ x 2λ, minimum separation is 1λ.
- Initially, layout area is 11λ x 11λ.
- After compacting along the y-direction, and then along the x-direction, layout area reduces to 11λ x 8λ.



2D Compaction

- Each square is 2λ x 2λ, minimum separation is 1λ.
- Initially, layout area is 11λ x 11λ.
- After 2D compaction, layout area reduces to 8λ x 8λ.

2D Compaction (contd.)

- 2D compaction is in general much better than performing 1D compaction.
- 2D compaction, if solved optimally, produces minimum-area layouts.
  - It is very time consuming.
  - Thus 1½-D compaction techniques have been proposed.
    - Perform x-direction compaction moves while making small moves in the y-direction.

1½-Dimensional Compaction

- A deterministic algorithm.
  - Key idea is to provide enough lateral movements to blocks during compaction to resolve interferences.
- This is called 1½-dimensional compactor, since the geometry is not as free as in true 2-dimensional compaction.

- The algorithm maintains an X-Y adjacency graph.
  - Vertices represent blocks.
  - Edges represent horizontal and vertical adjacency.
    - Two blocks have a horizontal edge if they share a vertical boundary.
    - Two blocks have a vertical edge if they share a horizontal boundary.
    - The labels on the edges represent the minimum allowable distance between blocks.

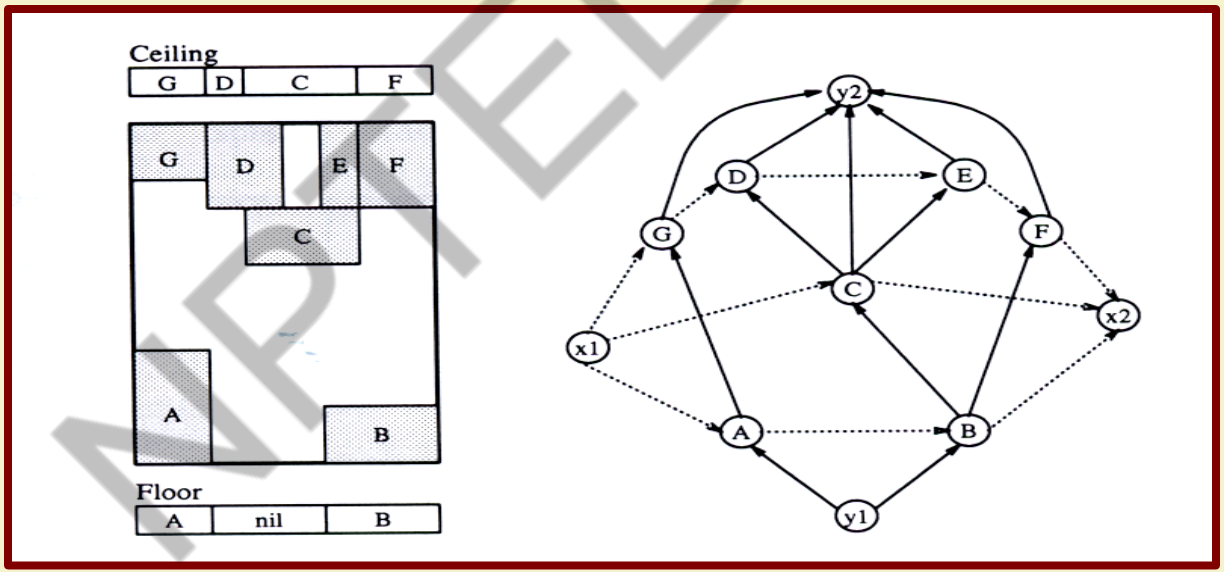
  - Four additional vertices are added to keep all the blocks within the required bounded rectangle.
  - Free space is ignored in computing the neighborhood edges between blocks.

- The algorithm assumes that the input is a partially completed layout, obtained by two applications of a 1-D compactor.

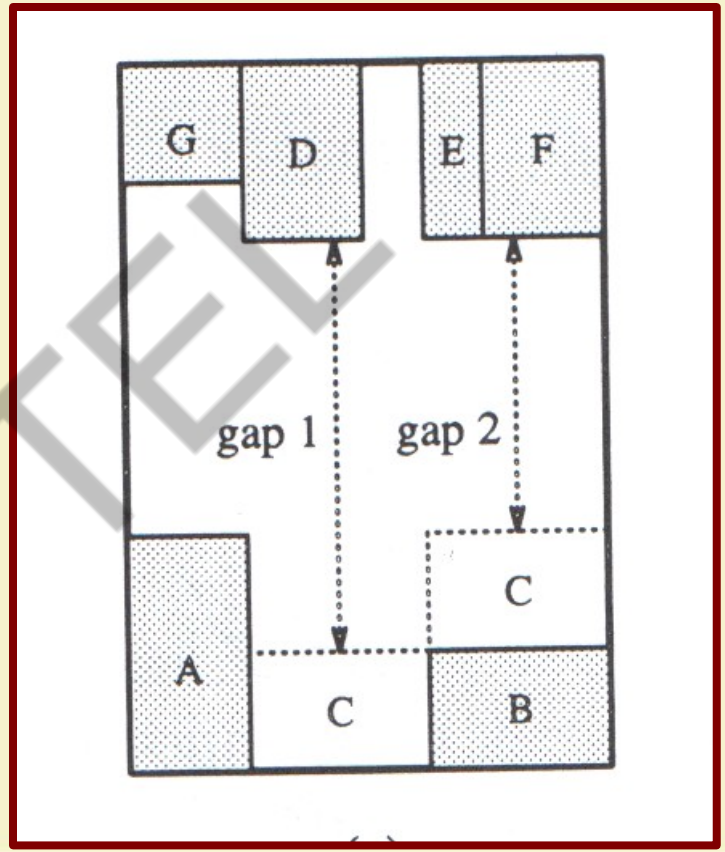
- Maintains two lists "*floor*" and "*ceiling*".
  - Floor consists of all the blocks which are visible from the top, and may become a neighbor of future block.
  - Ceiling is a list of all blocks which can be moved immediately (namely, those which are visible from the bottom).
- Selects the lowest block in the ceiling list and moves it to the place on the floor which maximizes the gap between *floor* and *ceiling*.
- The process is continued until all blocks are moved from *ceiling* to *floor*.

Example

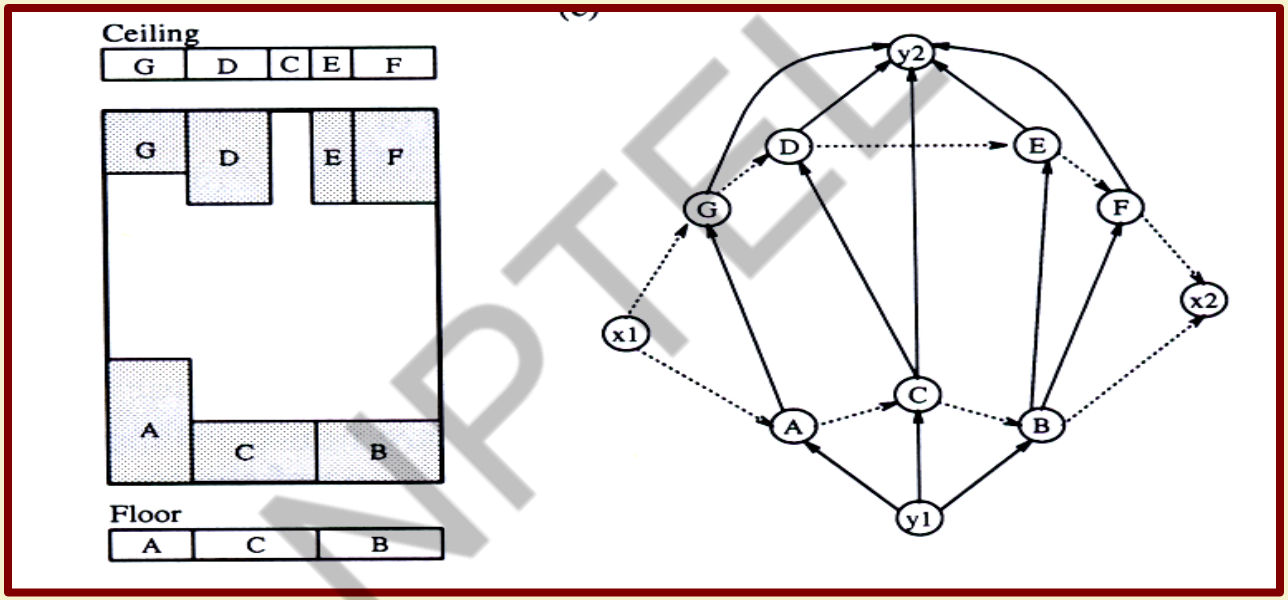
- Since *C* is the lowest block in the ceiling list, it is selected for the move.



- The gap is maximum at the boundary between blocks *A* and *B*.



- The modified layout and the adjacency graph is shown.



END OF LECTURE 47