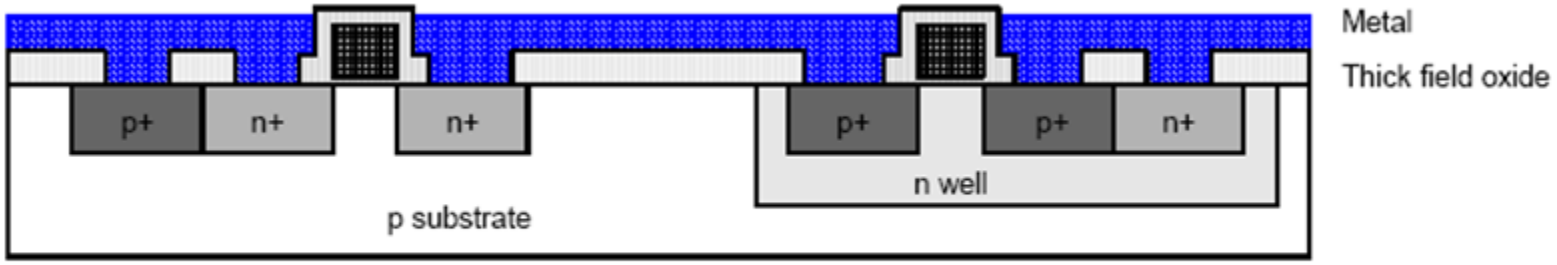
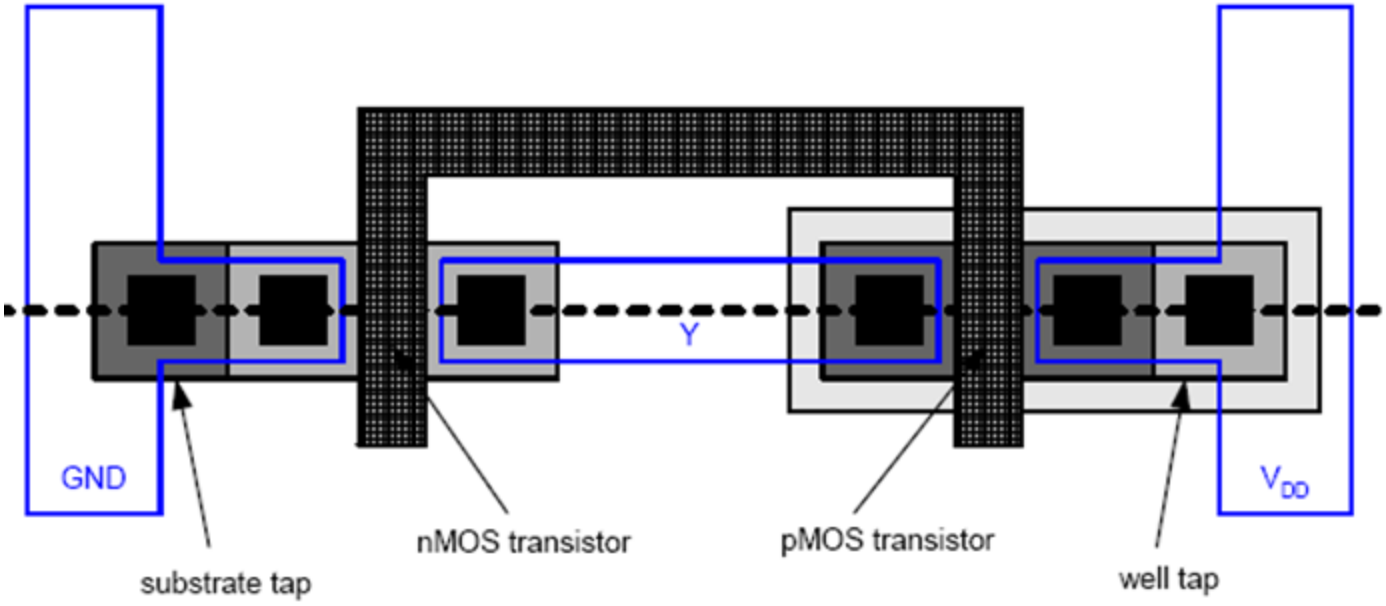
Physical Design / Layout Design

P.K. Shetty, MSIS

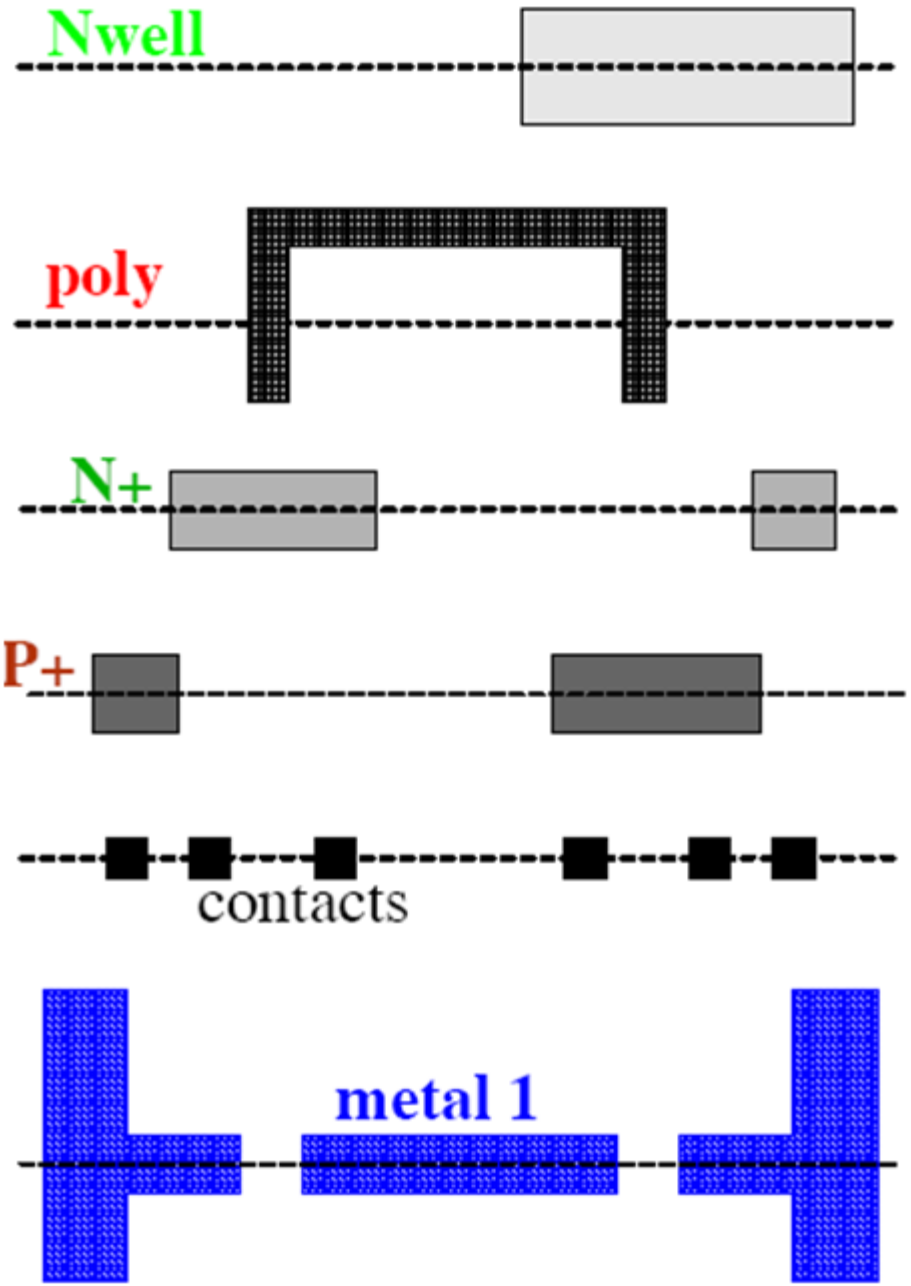
Layout (mask) view of the Inverter



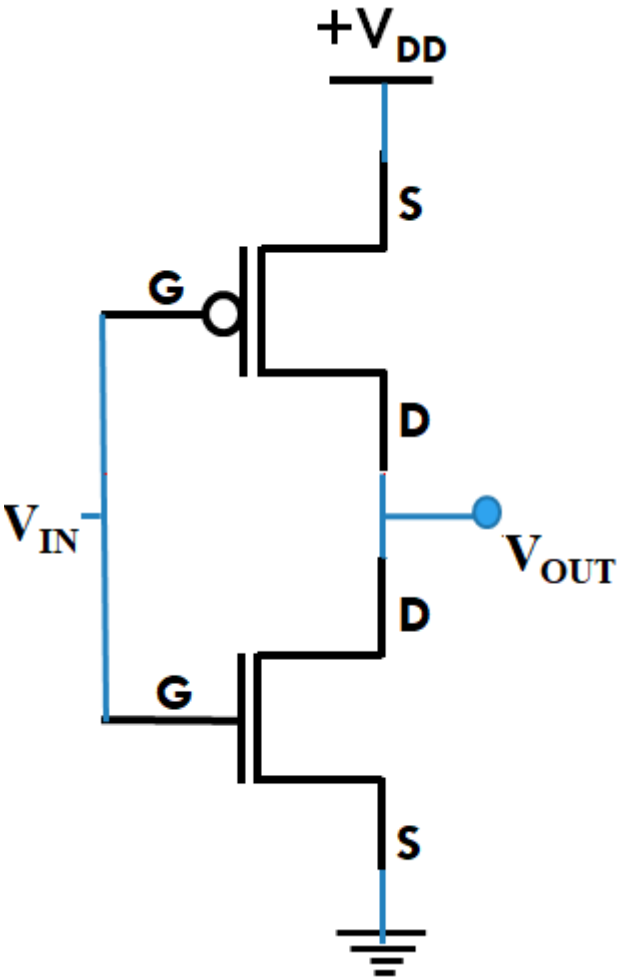
Layout (mask) view of the inverter.



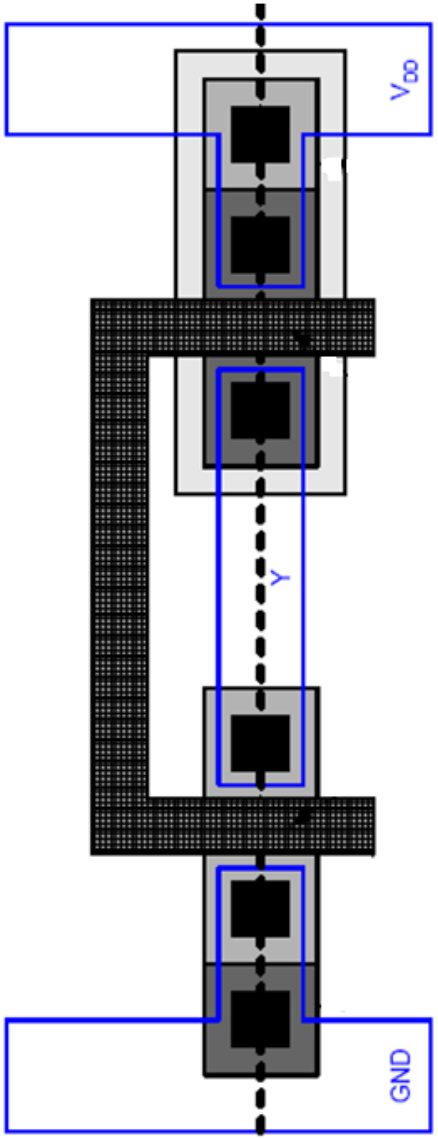
Masks:



Schematic Diagram



Layout Diagram



Design / Layout Rules

Main objective of design rules is to obtain a circuit with optimum yield in as small an area as possible.

They represent a compromise between performance and yield.

- More conservative rules increase probability of correct circuit function
- More aggressive rules increase circuit performance (may at the expense of yield).

Two approaches used:

- Lambda based rules
- Micron based rules

Lambda Based Rules

Also known as ***scalable design rules*** as they allow first order scaling

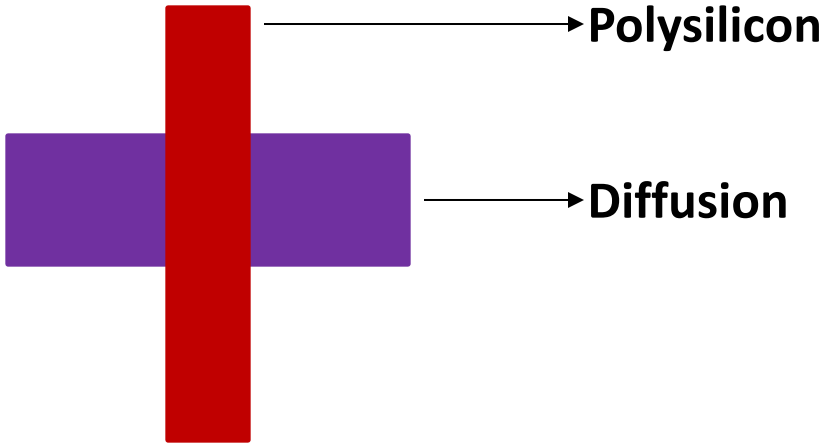
- Moving from one process to another requires only a change in λ
- Worked well for 4µm down to 1.2µm processes
- In general, process rarely shrinks uniformly

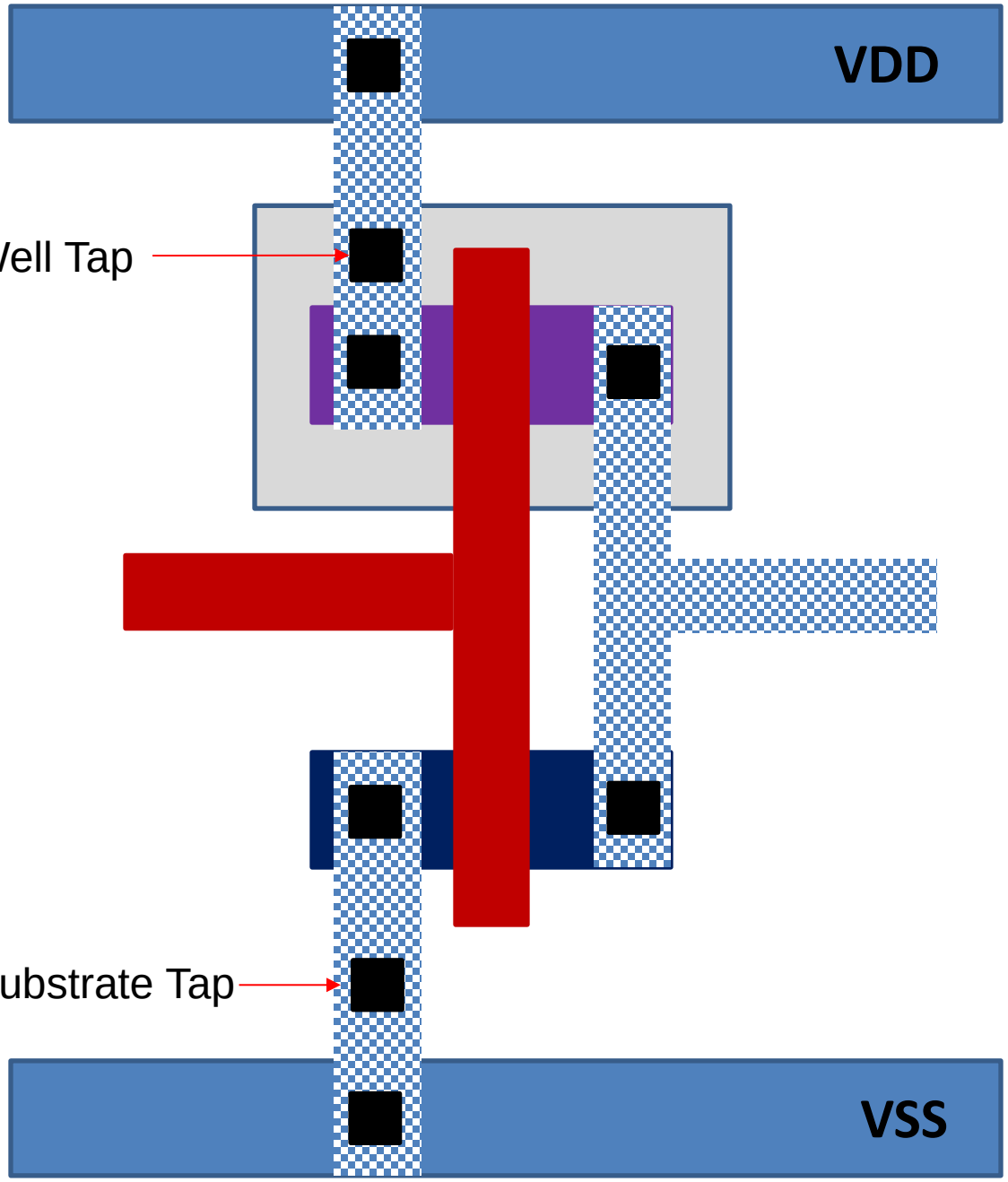
Micron Based Rules

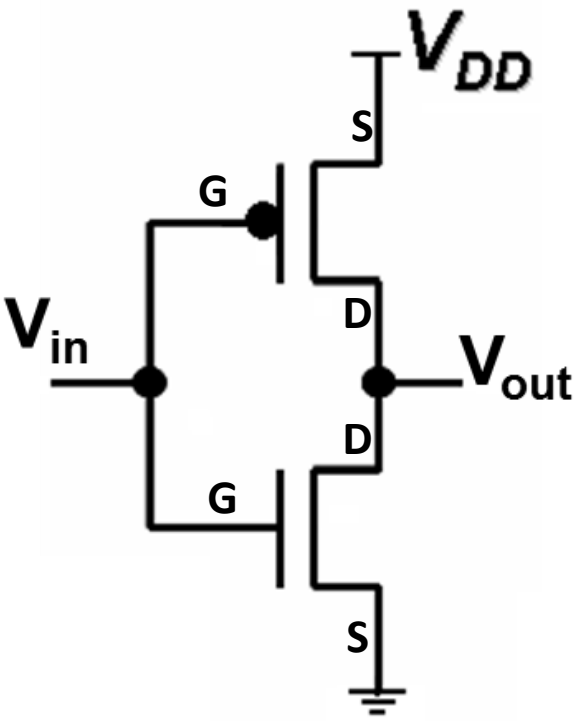
- All minimum sizes and spacing specified in microns
- Rules don't have to be multiples of λ
- Can result in 50% reduction in area over λ based rules
- Standard in industry

Mask Layout

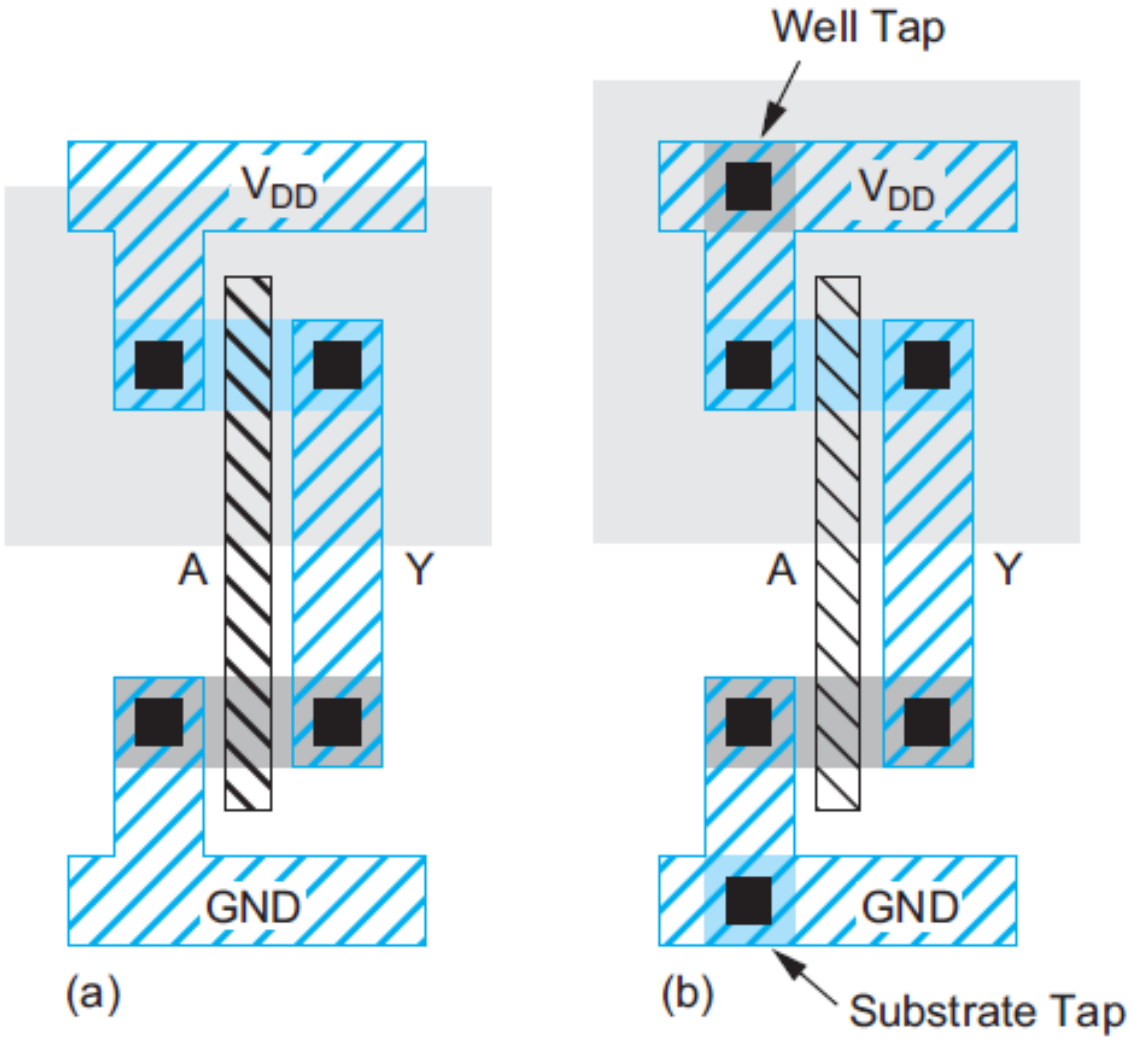
- Whenever a ***poly*** crosses a ***diffusion*** we get a ***transistor***




VDD
Well Tap
Substrate Tap
VSS


V_{DD}
S
G
D
V_{in}
V_{out}
D
G
S

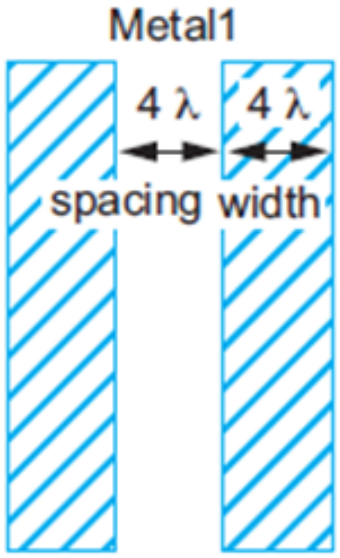
Mask Layout of CMOS Inverter

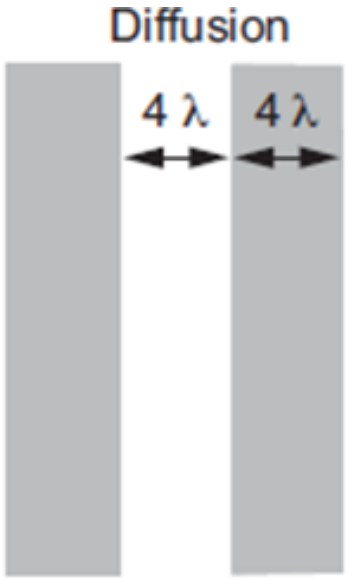


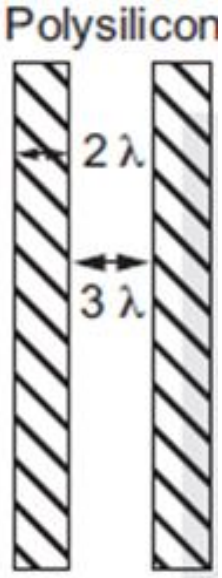
Inverter cell layout

Layout / Design Rules

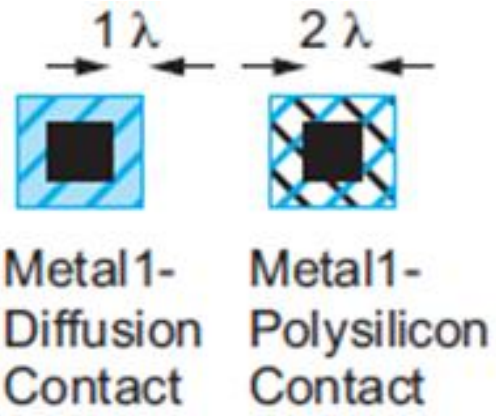
Simplified λ-based design rules

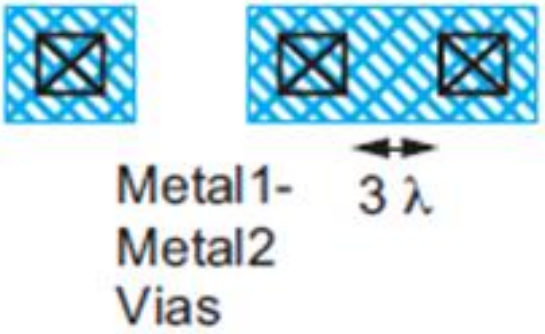


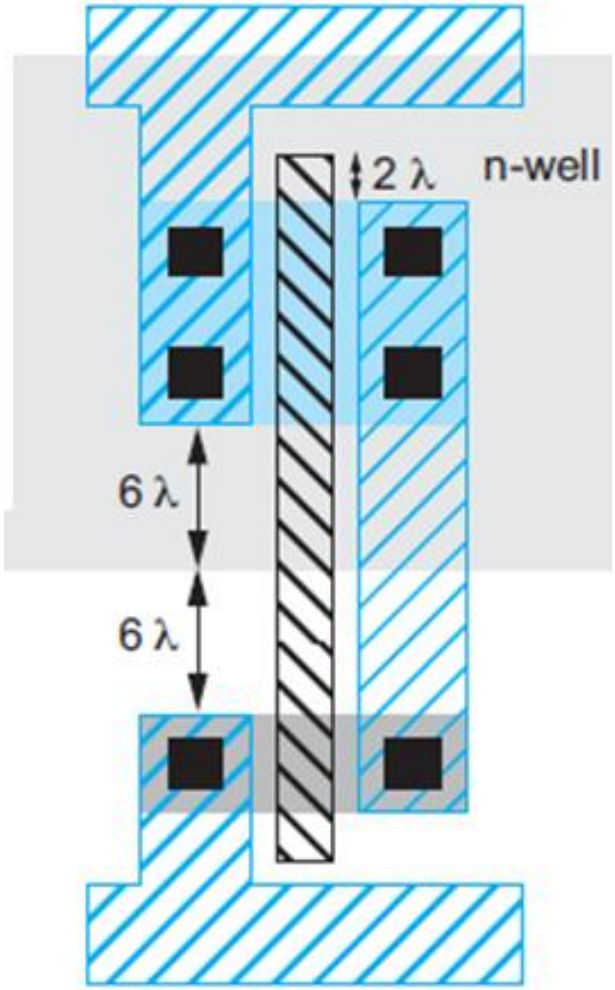




Layout / Design Rules

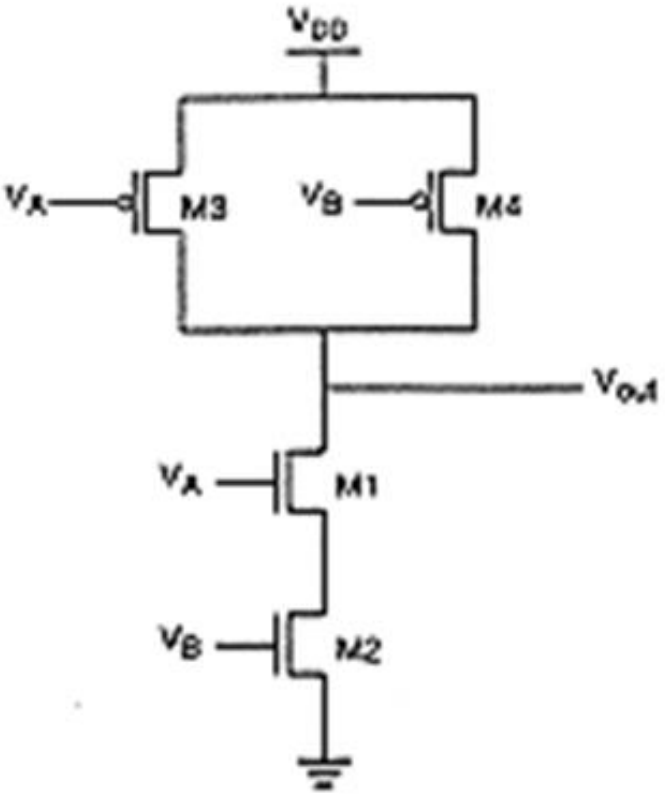


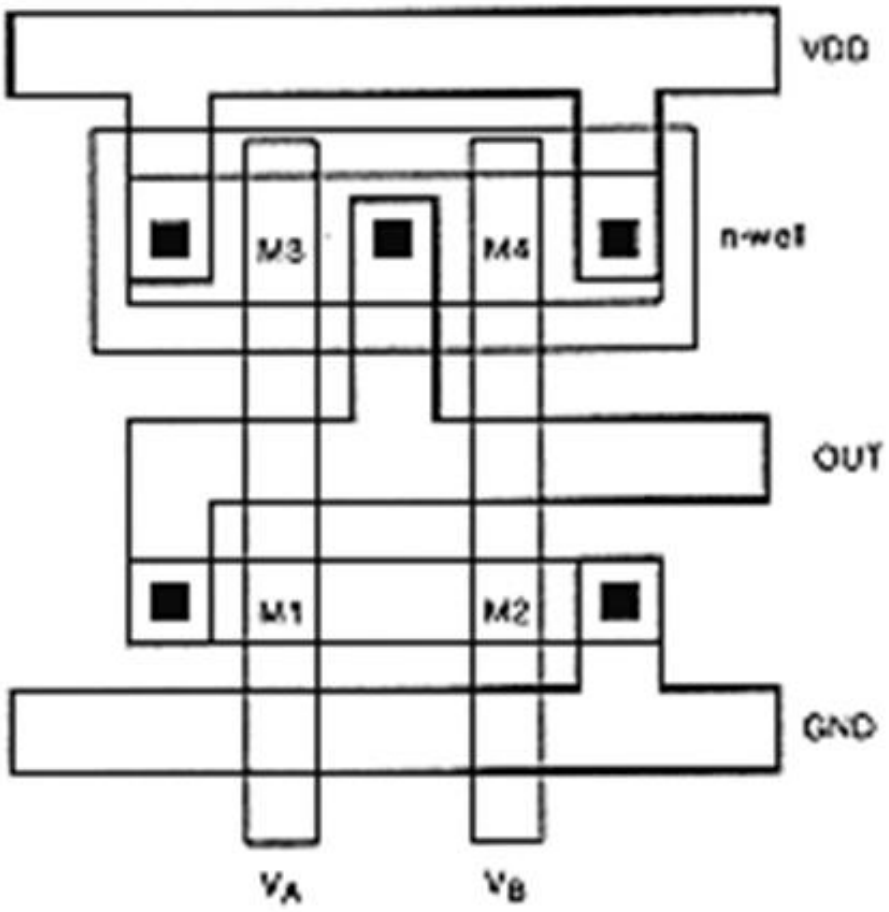




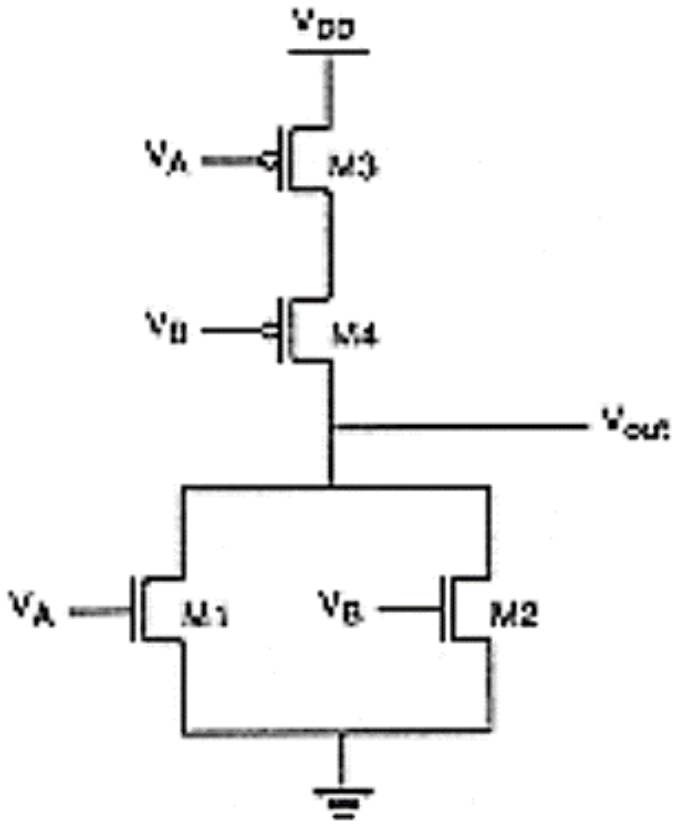
Simplified λ-based design rules

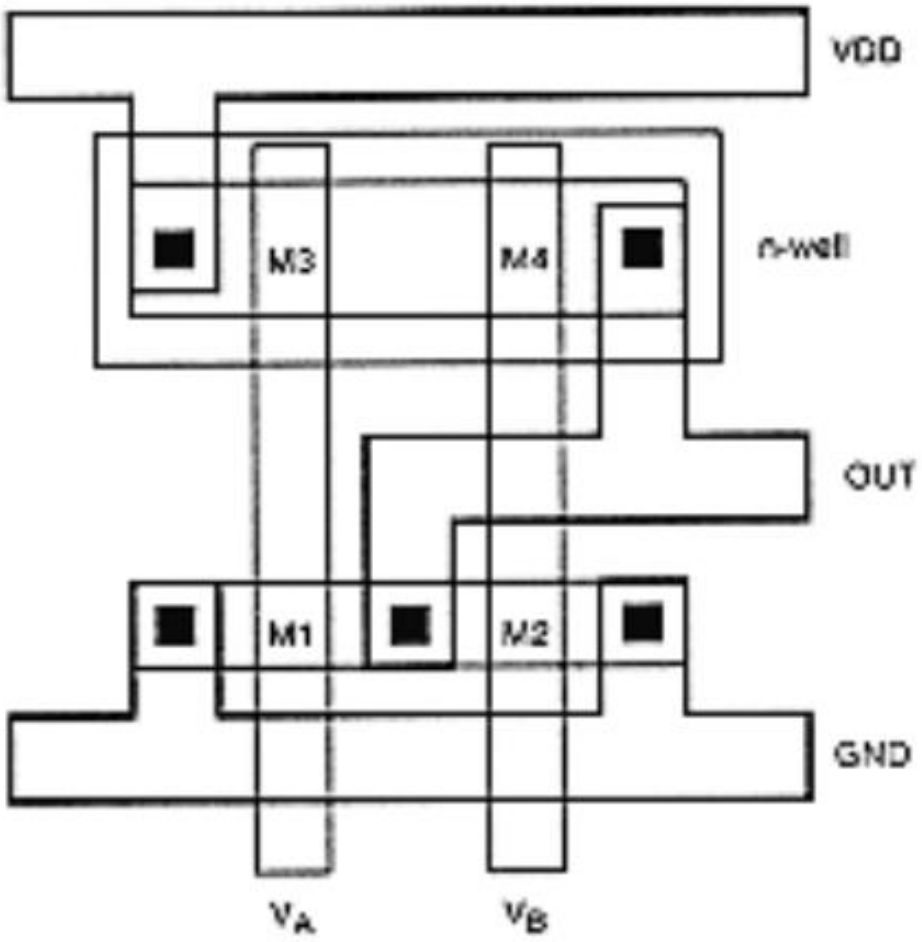
Mask Layout of CMOS NAND Gate





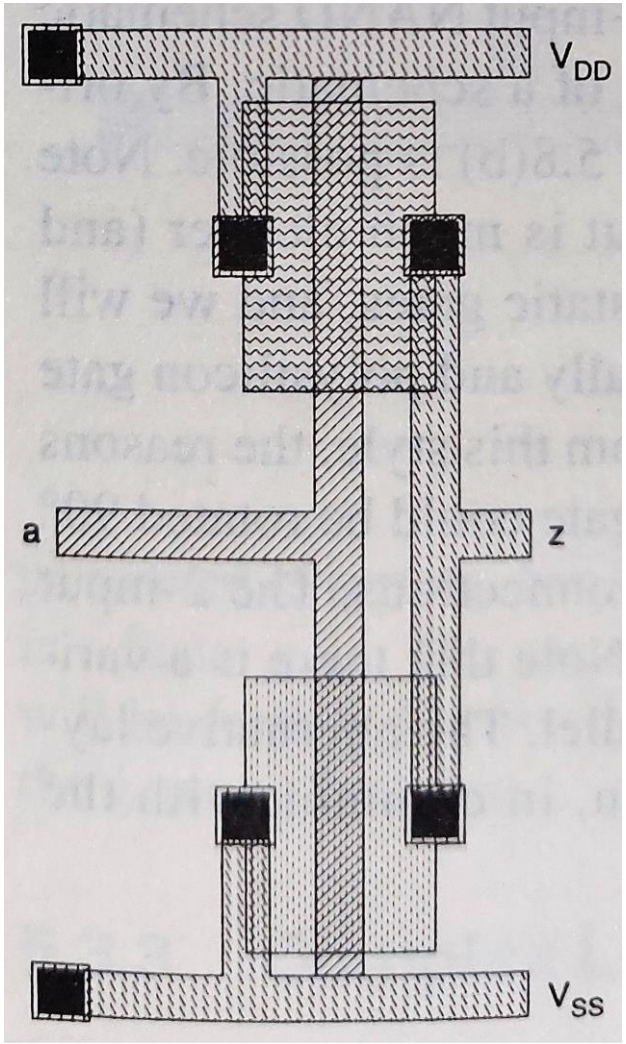
Mask Layout of CMOS NOR Gate





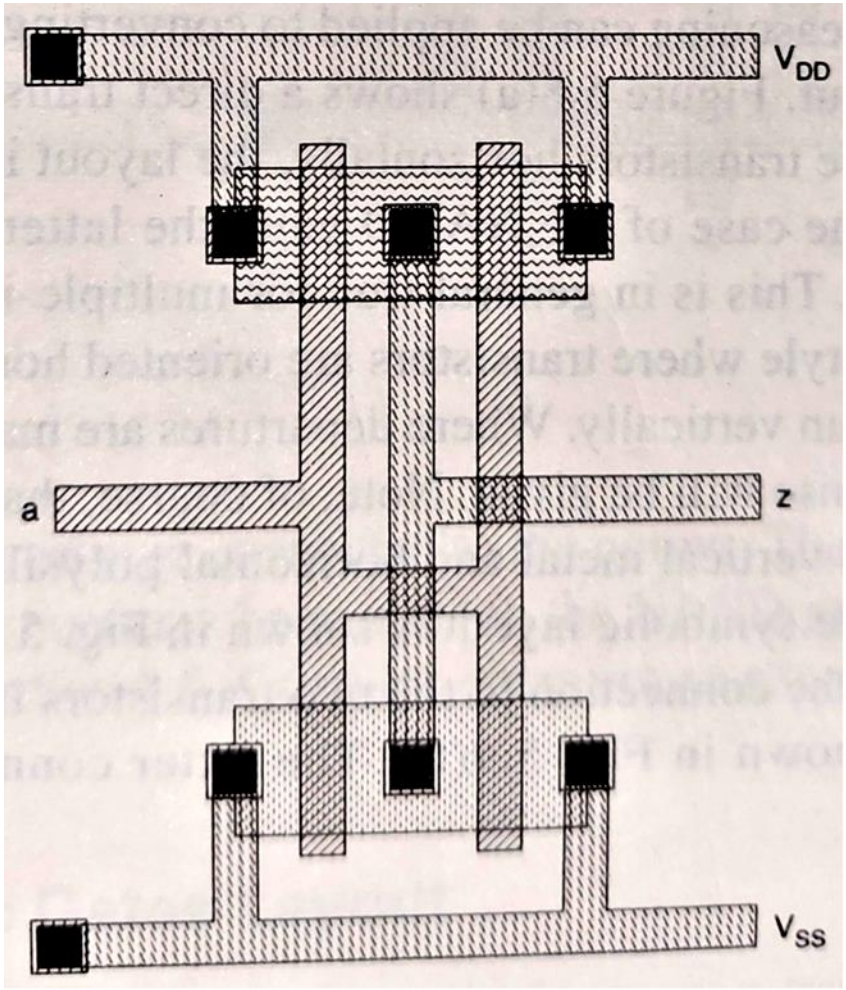
Constructing Large Inverters

- Large inverters are required to handle large power
- Increasing the $\frac{\mathbf{W}}{\mathbf{L}}$ ratio
- Increases the drain area and hence Capacitance



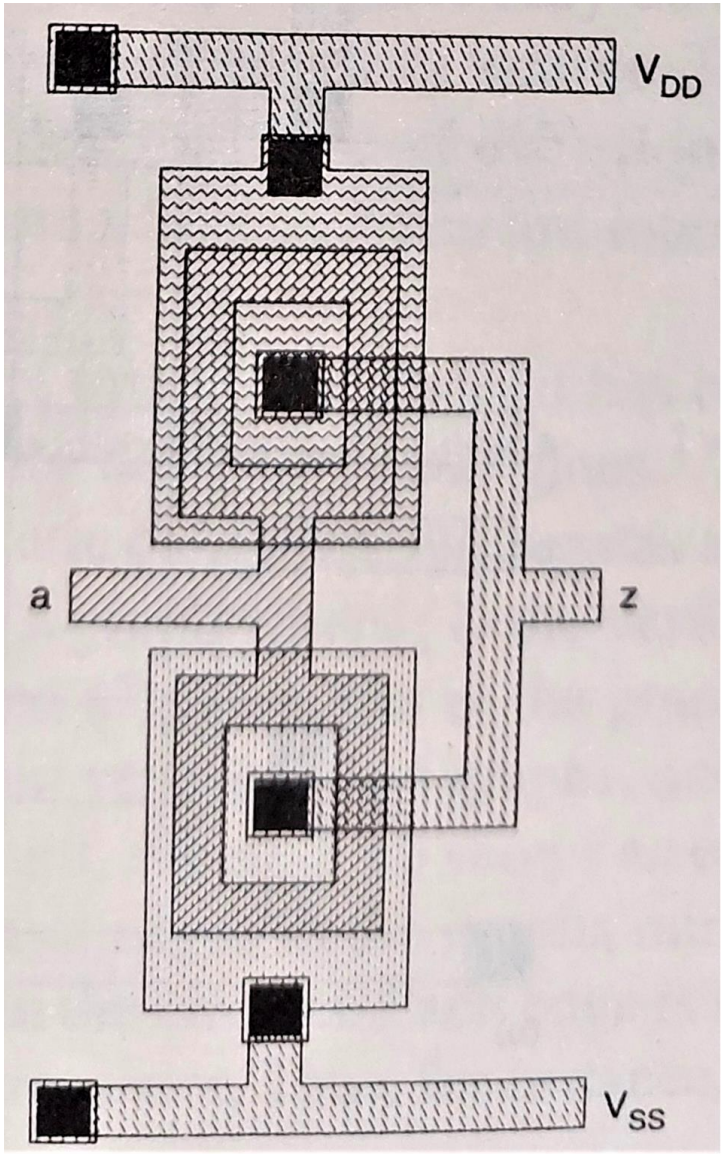
Constructing Large Inverters

- Connecting many smaller inverters in parallel
- Yields a more optimum drain capacitance due to merged drain regions
- β is doubled



Constructing Large Inverters

- Further reduction in drain capacitance is achieved by using the ***donut*** ("round transistor") connection.
- Yields a more optimum drain capacitance due to merged drain regions
- β of the transistor is almost quadrupled



Complex Logic Gates Layout

- All complementary gates may be designed using a single row of n-transistors above or below a single row of p-transistors, aligned at common gate connections
- Most “simple” gates may be designed using an ***unbroken row of transistors*** in which abutting source-drain connections are made
- This is called the ***“line of diffusion”*** rule

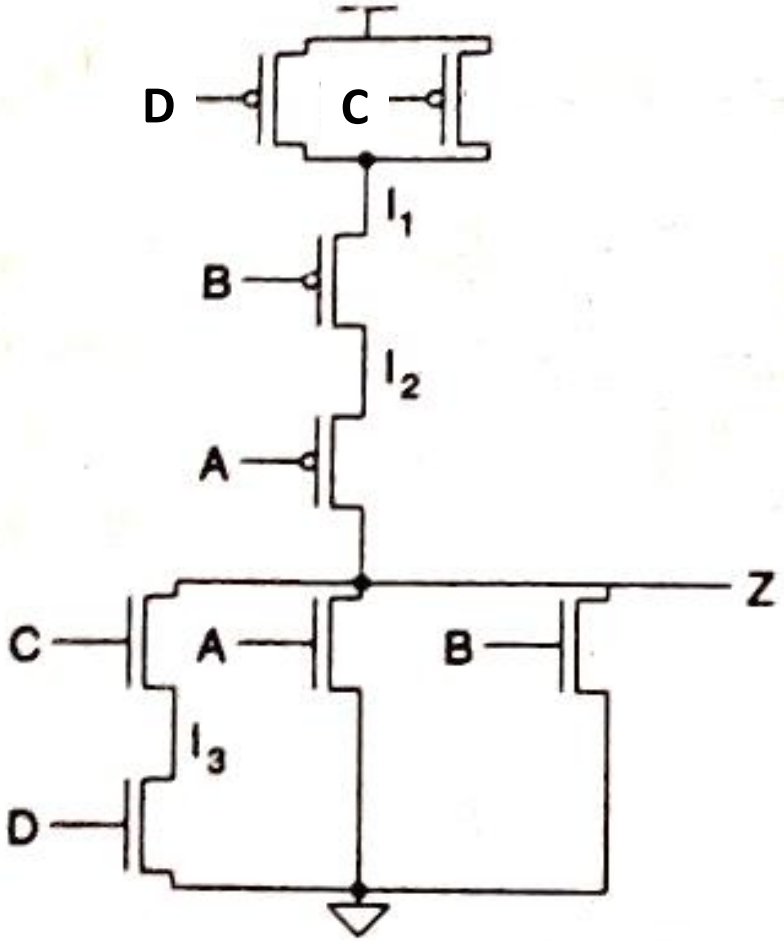
Layout Automation

Automated layout techniques applicable to static complementary:

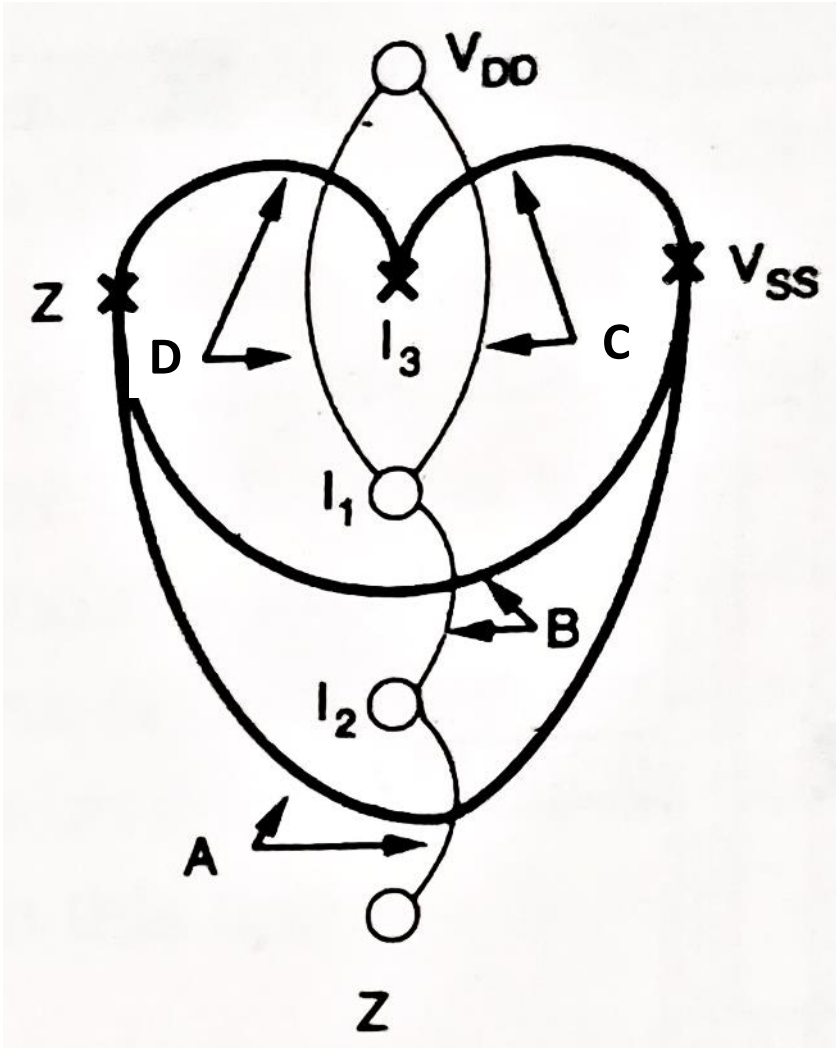
- CMOS circuit is converted into a graph where,
  - the vertices in the graph are the source/drain connections
  - the edges in the graph are transistors that connect particular source-drain vertices

- This results in 2 graphs: one for the n-logic tree and one for the p-logic tree

Example-1: [A + B + CD]’



Logic Circuit



Graph representation

- If two edges are adjacent in the p- or n-graphs, then they may share a common source-drain connection and may be connected by abutment

- If there exists a sequence of edges (containing all edges) in the p-graph and n-graph that have identical labelling, then the gate may be designed with no breaks

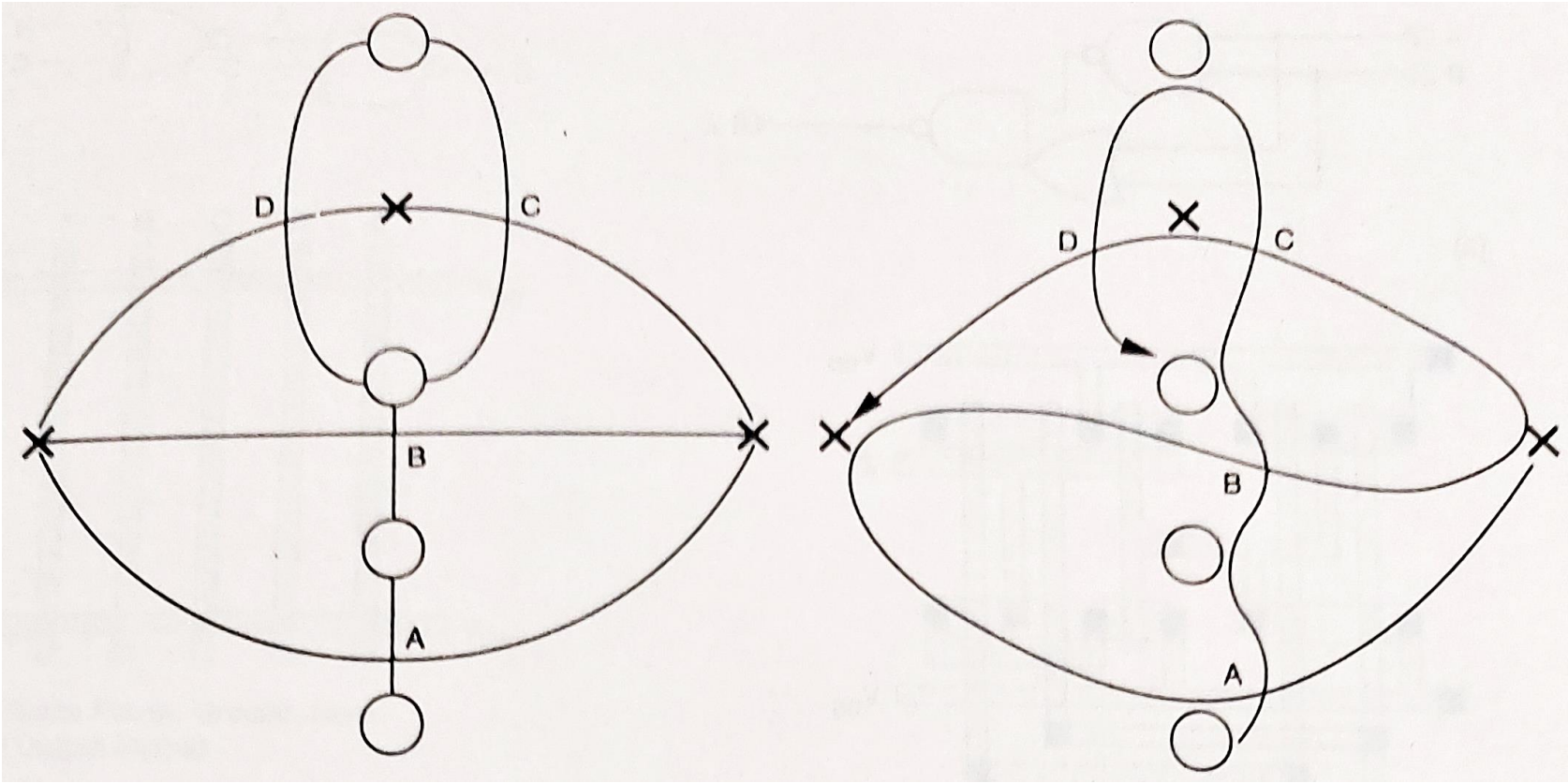
- This path is known as “Euler path”

Algorithm

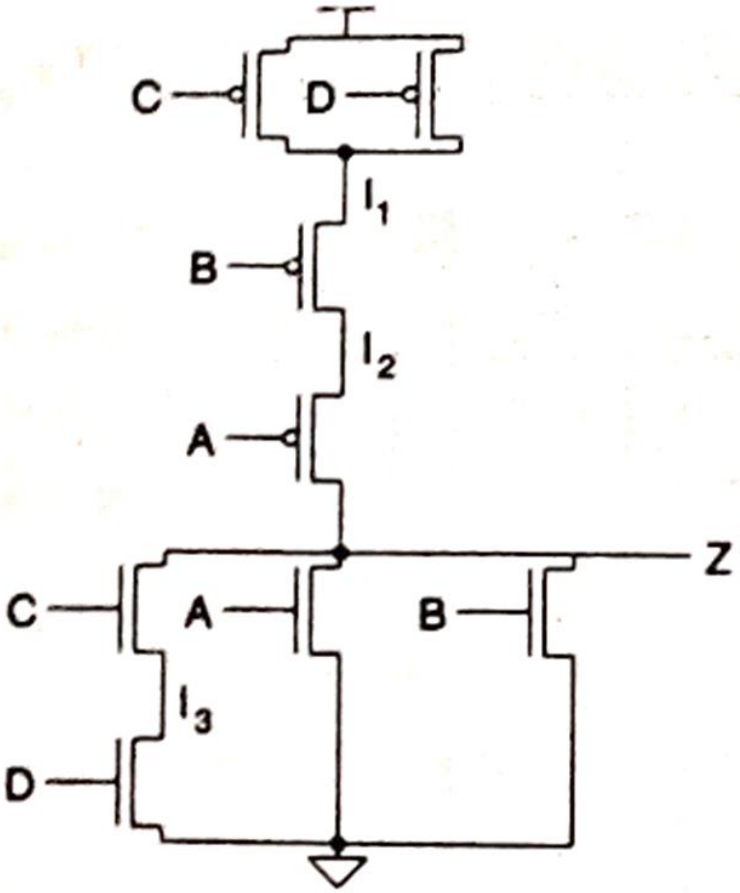
1. Find all Euler paths that cover the graph

2. Find a p- and n-Euler path that have identical labelling (a labelling is an ordering of the gate labels on each vertex)

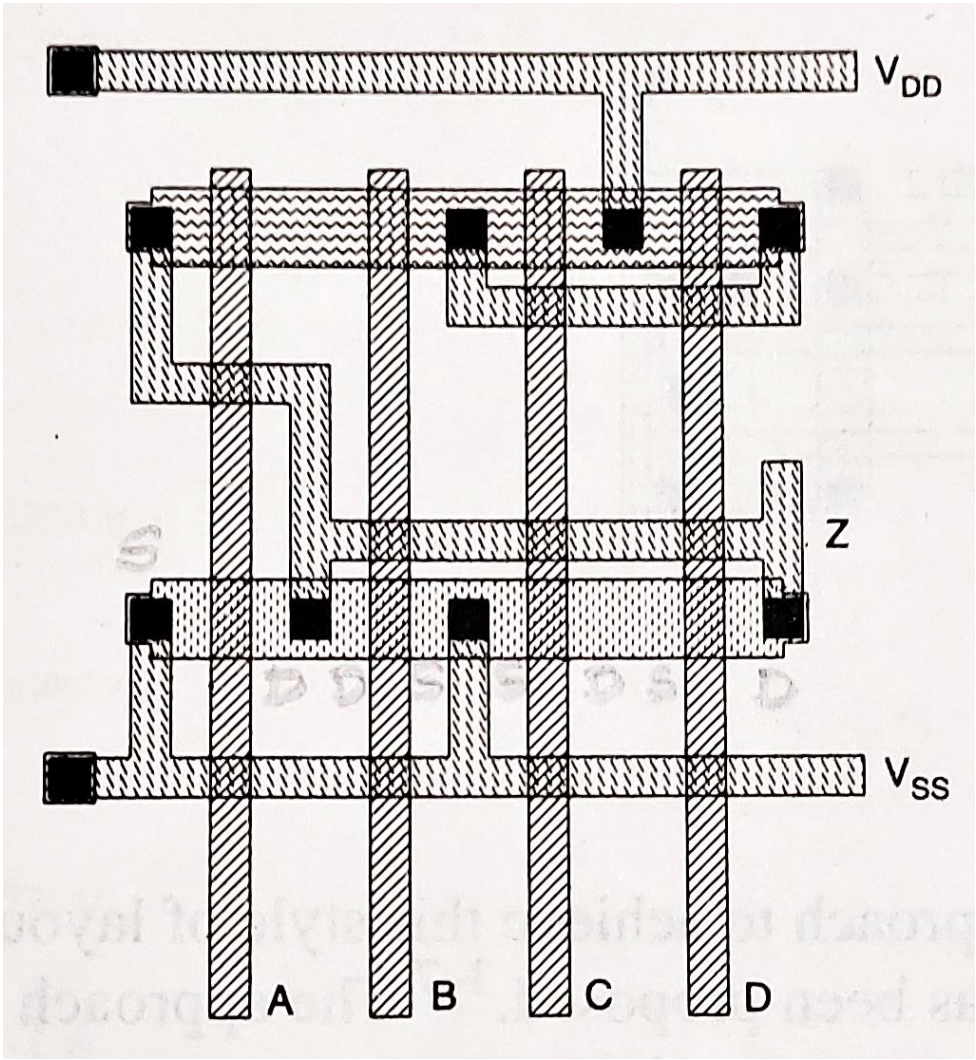
3. If the paths in step 2 are not found, then break the gate in the minimum number of places to achieve step 2 by separate Euler paths



Euler path is A-B-C-D



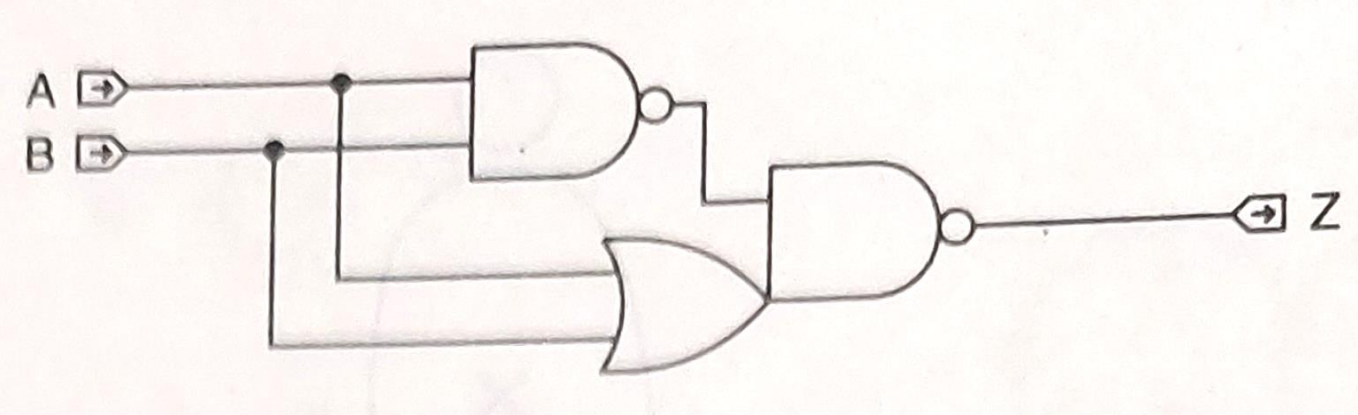
Circuit



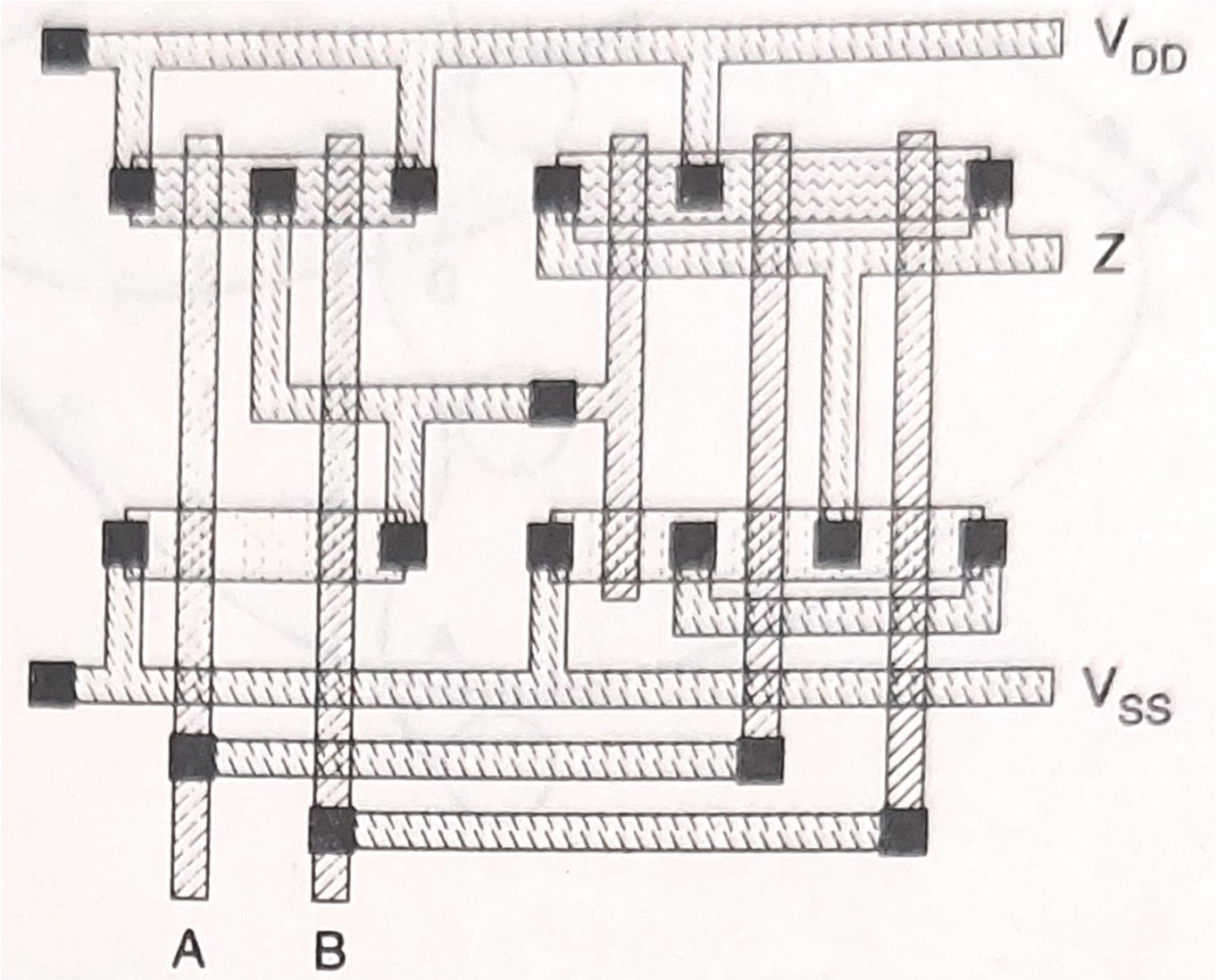
Layout

- A variation of the single line of n- and p- transistors occurs in logic gates where,
  - a signal is applied to the gates of multiple transistors
  - the gates are cascaded

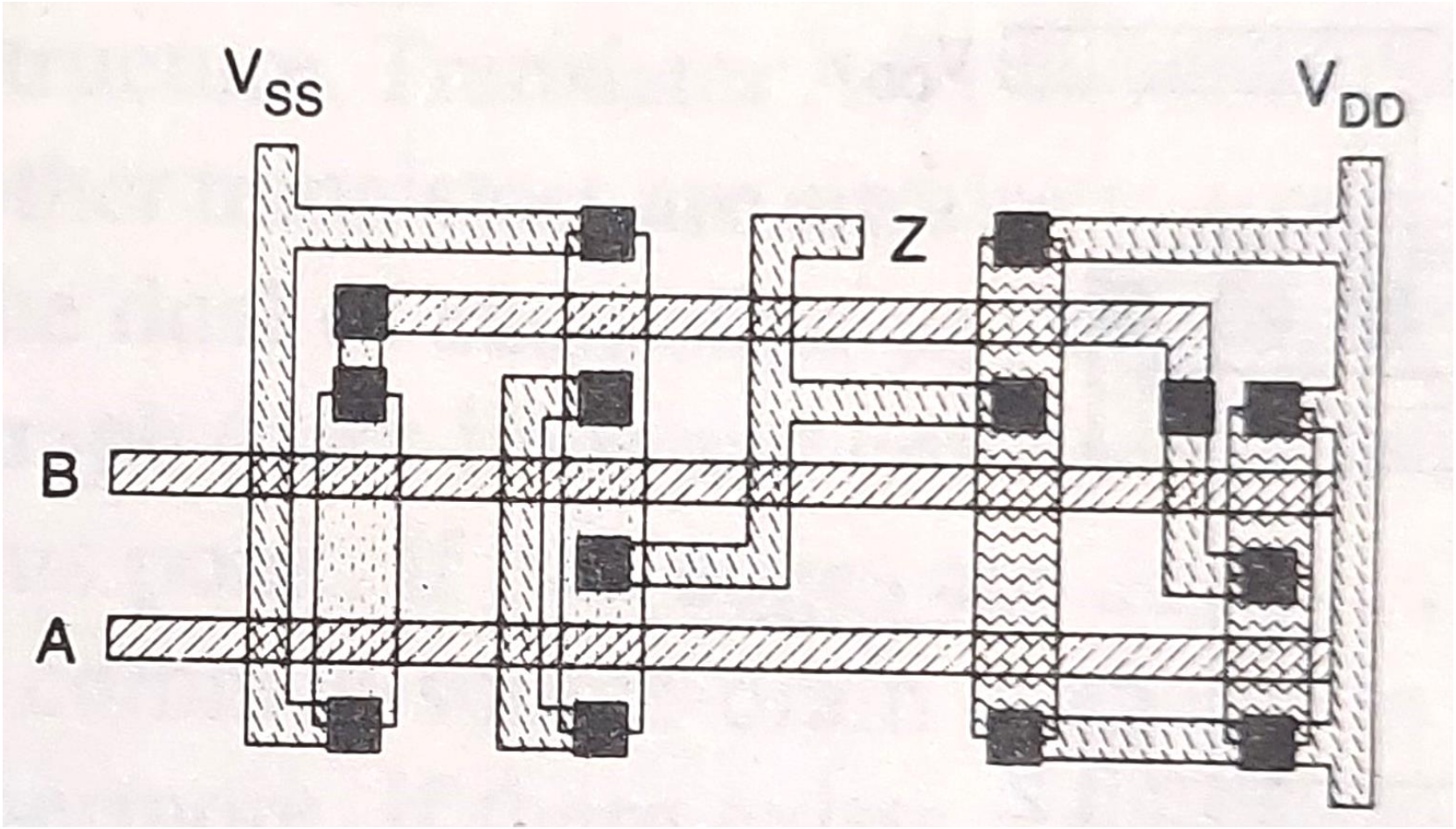
Example-2: ((AB)'.(A+B))' → XNOR gate



- Transistors may be stacked on the appropriate gate signal



With a single row of N and P transistors with a break



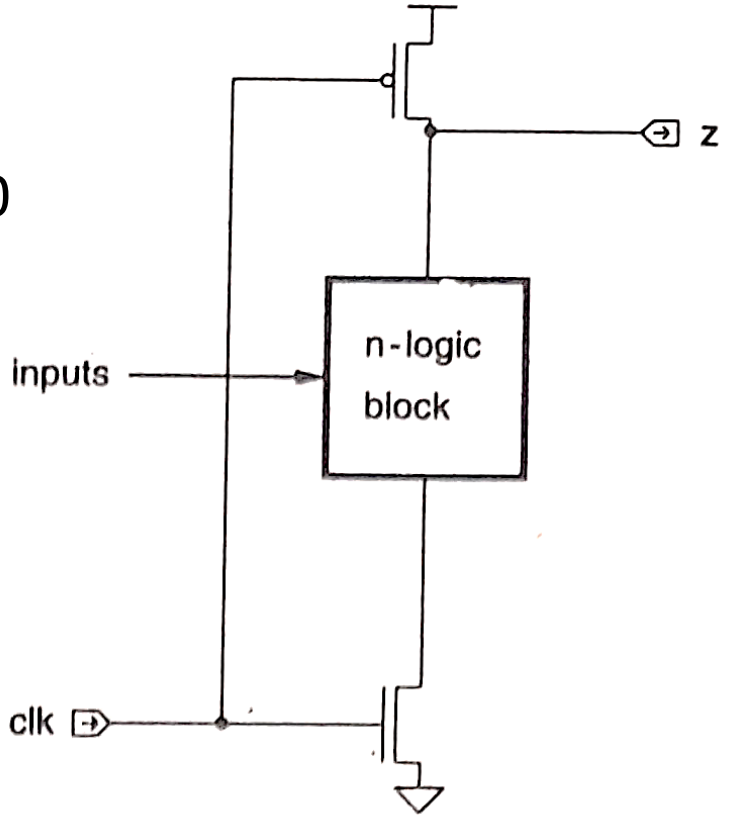
Uses a Stacked Layout

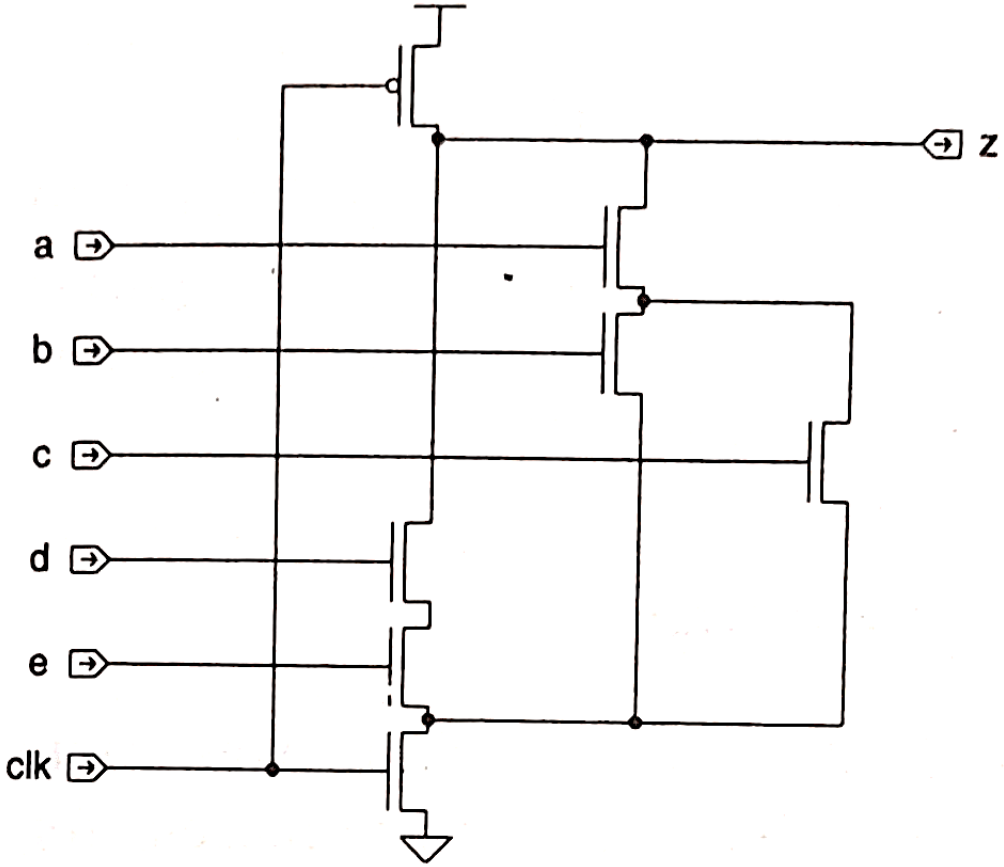
- The selection of the styles would depend on the overall layout – whether a
  - Short, fat cells were needed
  - Long, thin cells were needed

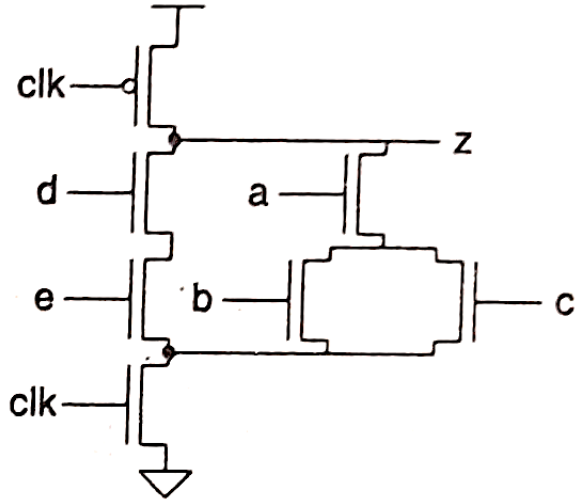
- The gate segments that are maximally connected to the supply and ground rails should be placed adjacent to these signals

Dynamic CMOS Logic

1. Single phase clock is used
2. The precharge phase occurs when clk = 0
3. The i/p capacitance of this gate is the same as the pseudo –NMOS gate
4. Pull-up time is improved by virtue of the active switch
5. Pull-down time is increased due to ground switch
6. Ground switch may be omitted if the inputs are guaranteed to be zero during precharge

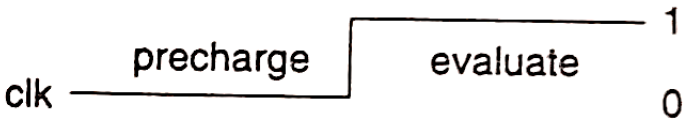






$Z = \overline{A.(B + C) + (D.E)}$ clk = 1

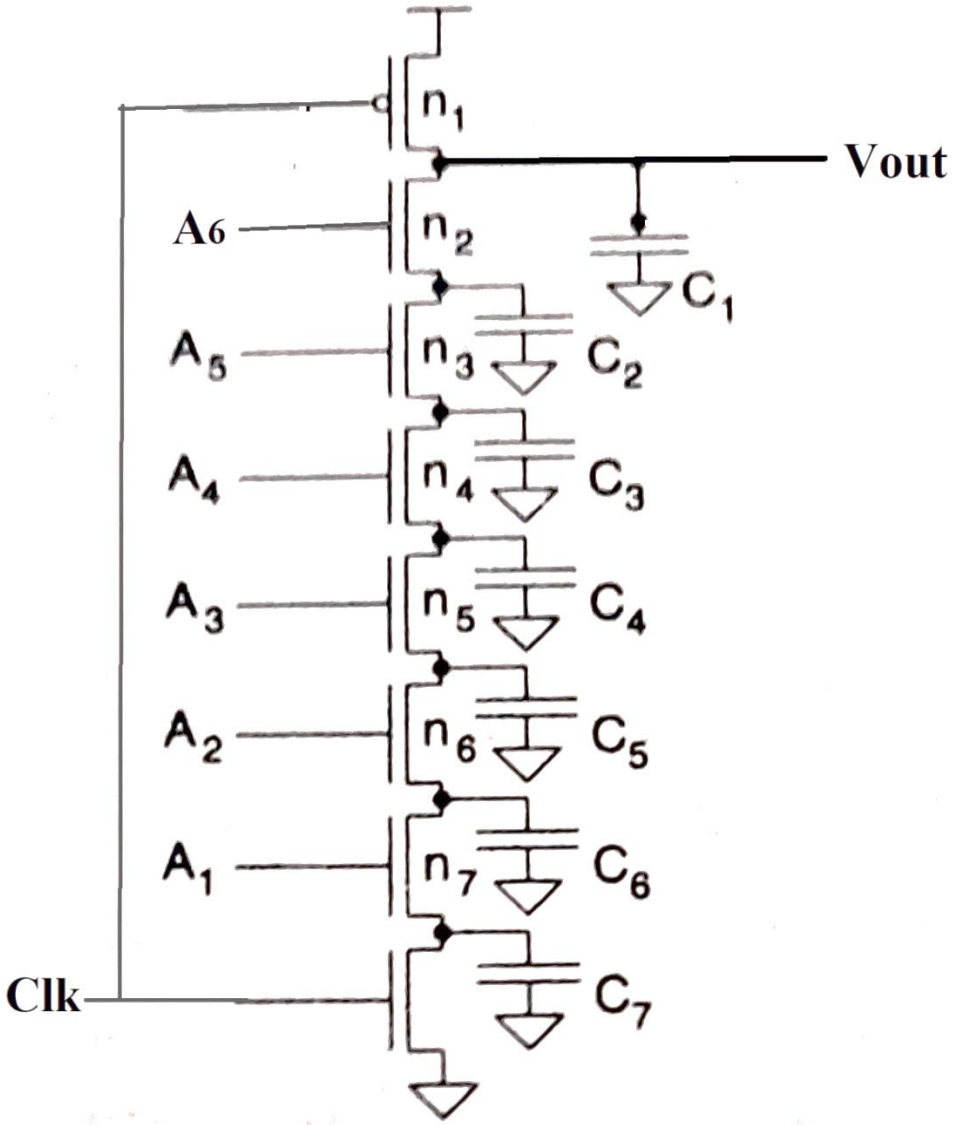
Z = HIGH clk = 0

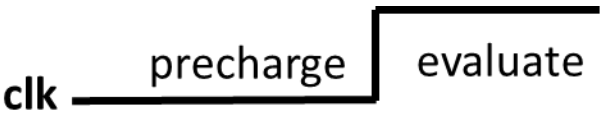


Limitations of Dynamic CMOS Logic

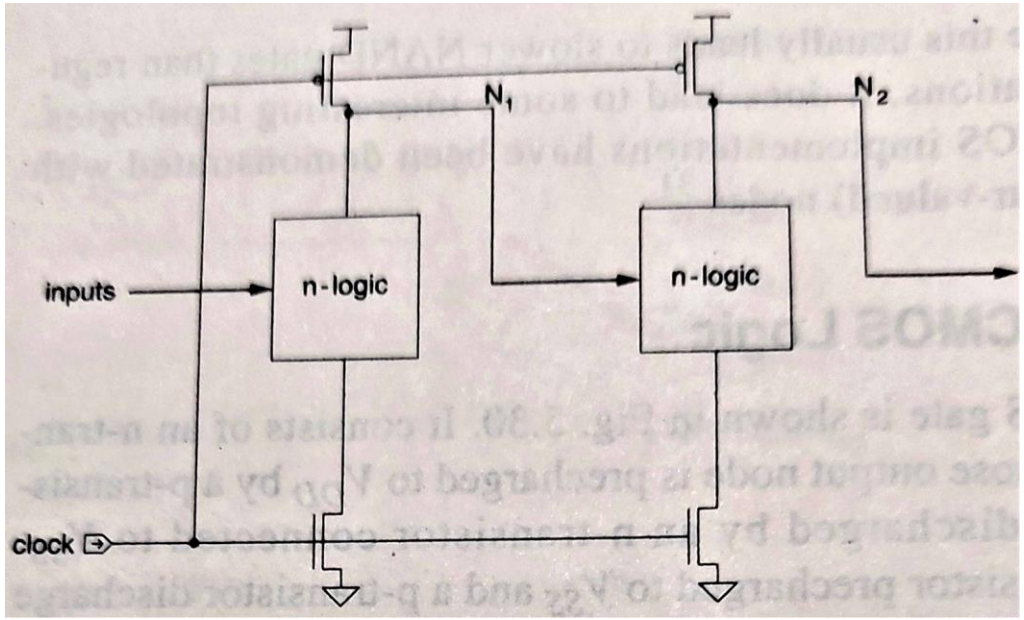
1. The inputs can only change during the precharge phase and must be stable during evaluate portion of the cycle
2. Simple, single-phase dynamic CMOS gates cannot be cascaded

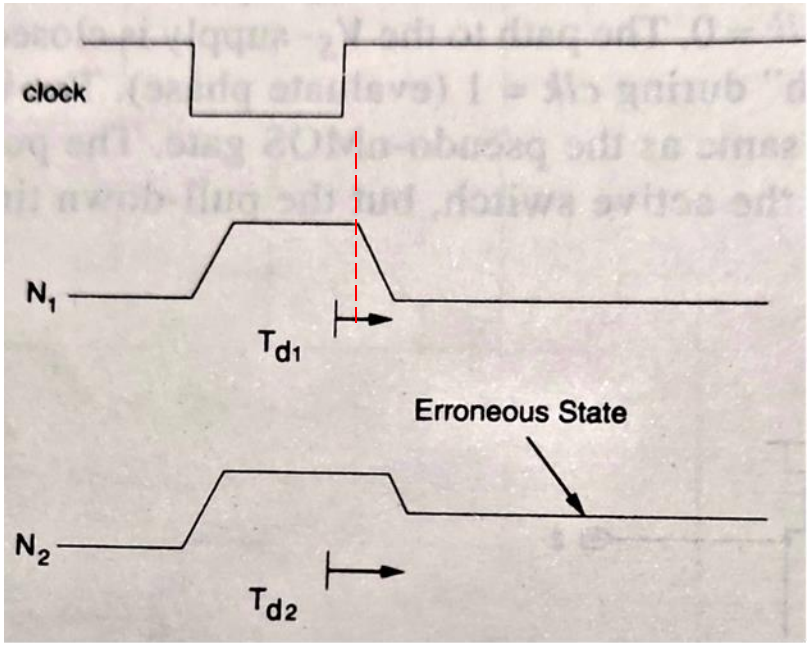
1. The inputs can only change during the precharge phase and must be stable during evaluate phase





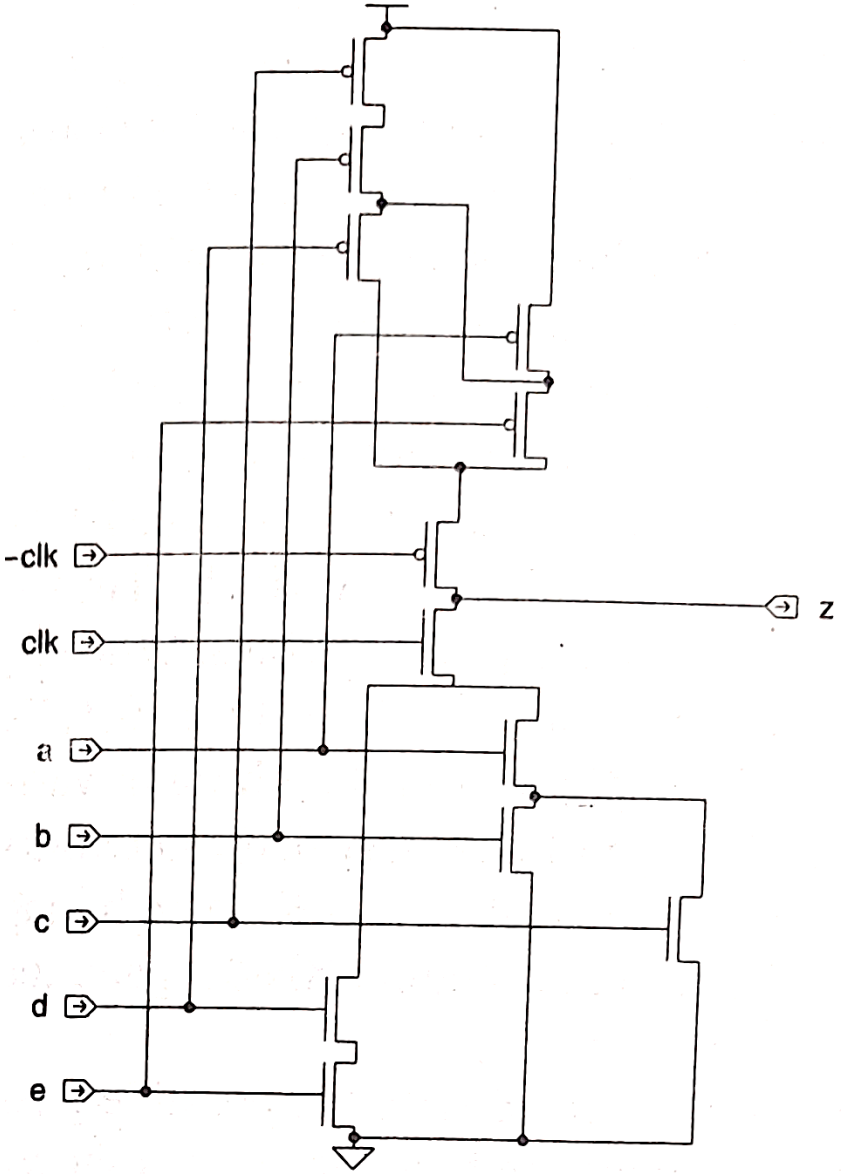
2. Simple single-phase dynamic CMOS gates cannot be cascaded



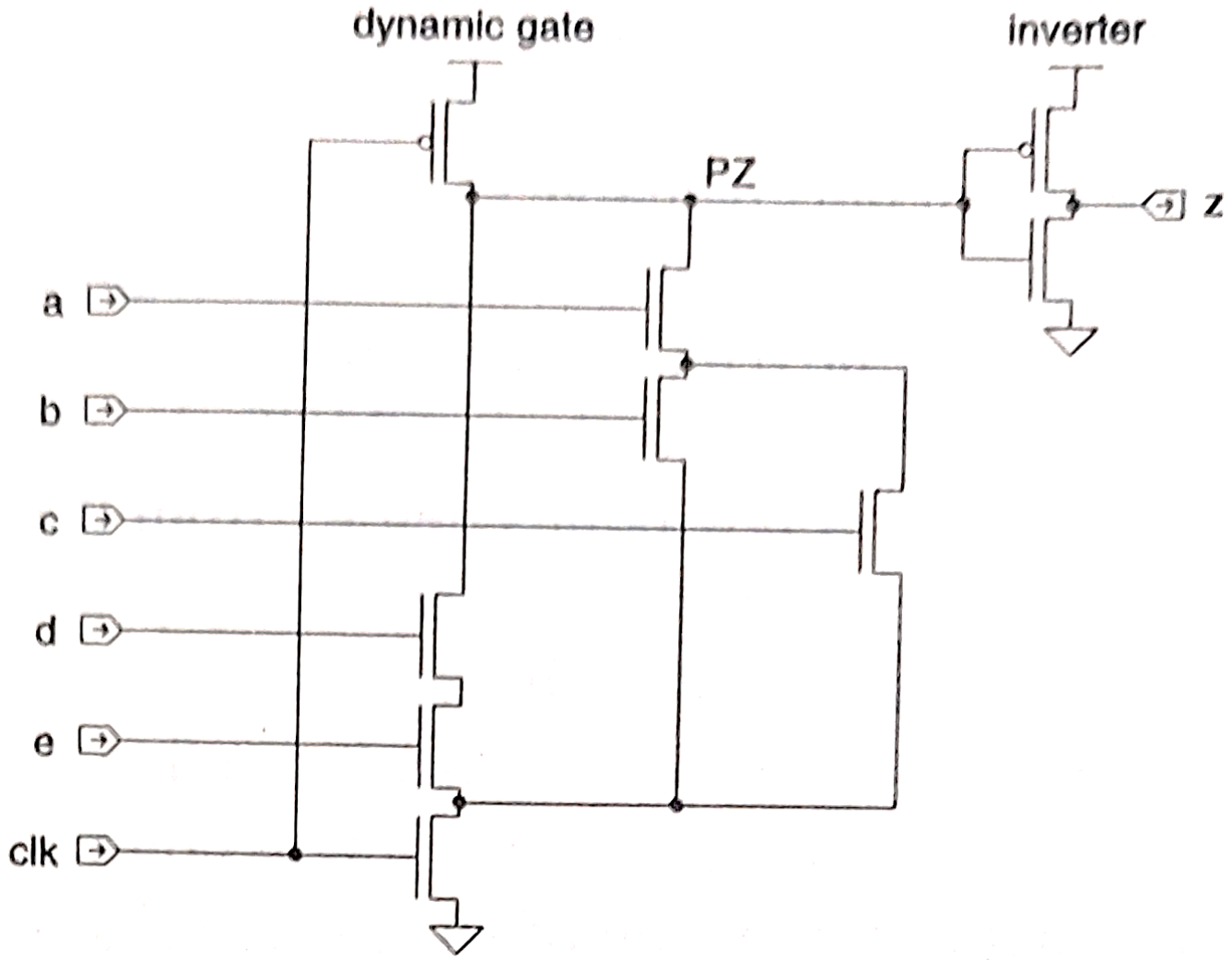


Clocked CMOS Logic (C^2MOS)

- The main use of this logic structure is to form clocked structures that incorporate latches or that interface with other dynamic forms of logic
- Has larger rise and fall times due to the series clocking transistors
- The series clock transistors can either be at the output of the gate or at the power supply ends



CMOS Domino Logic

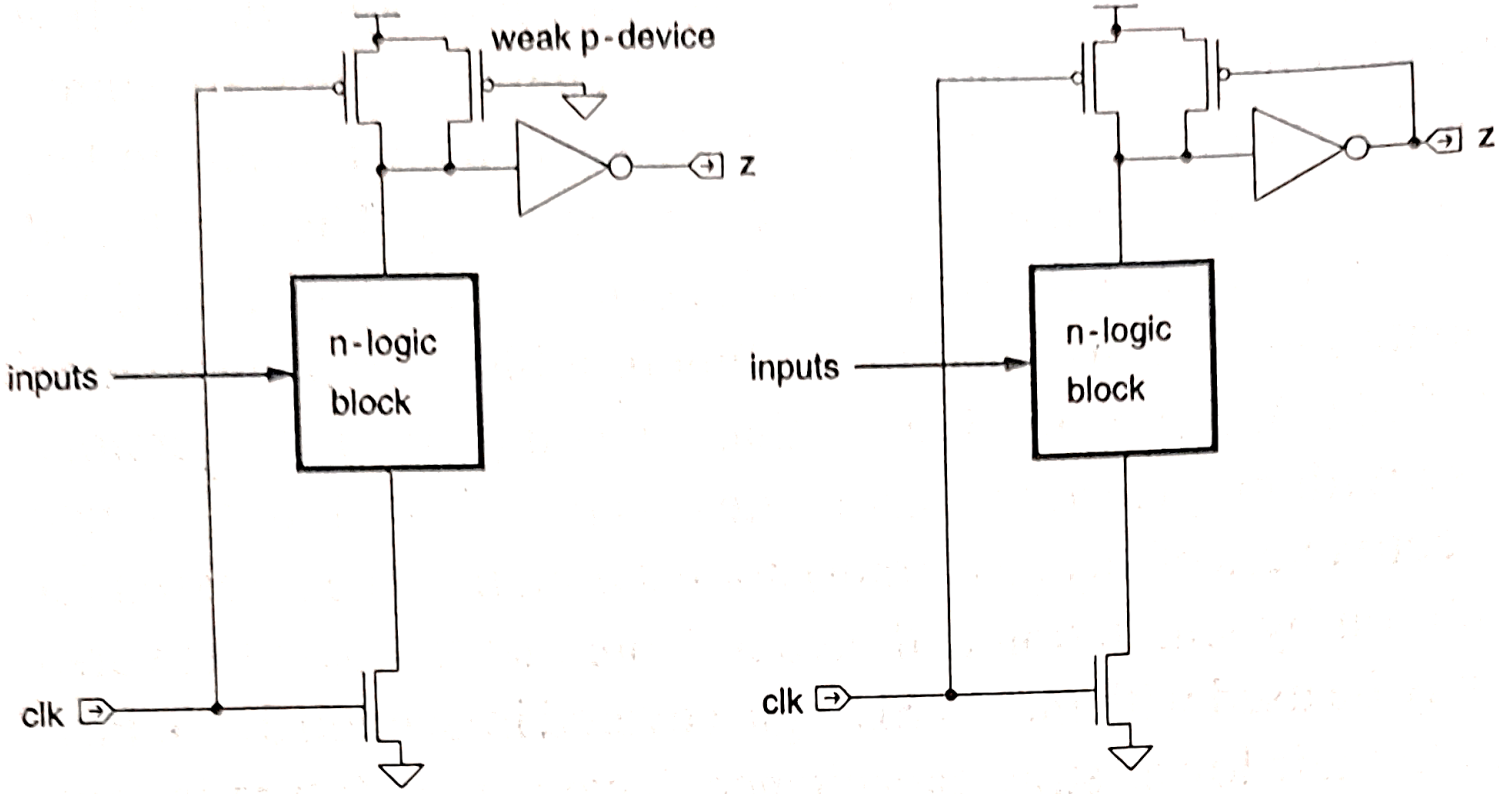


Limitations

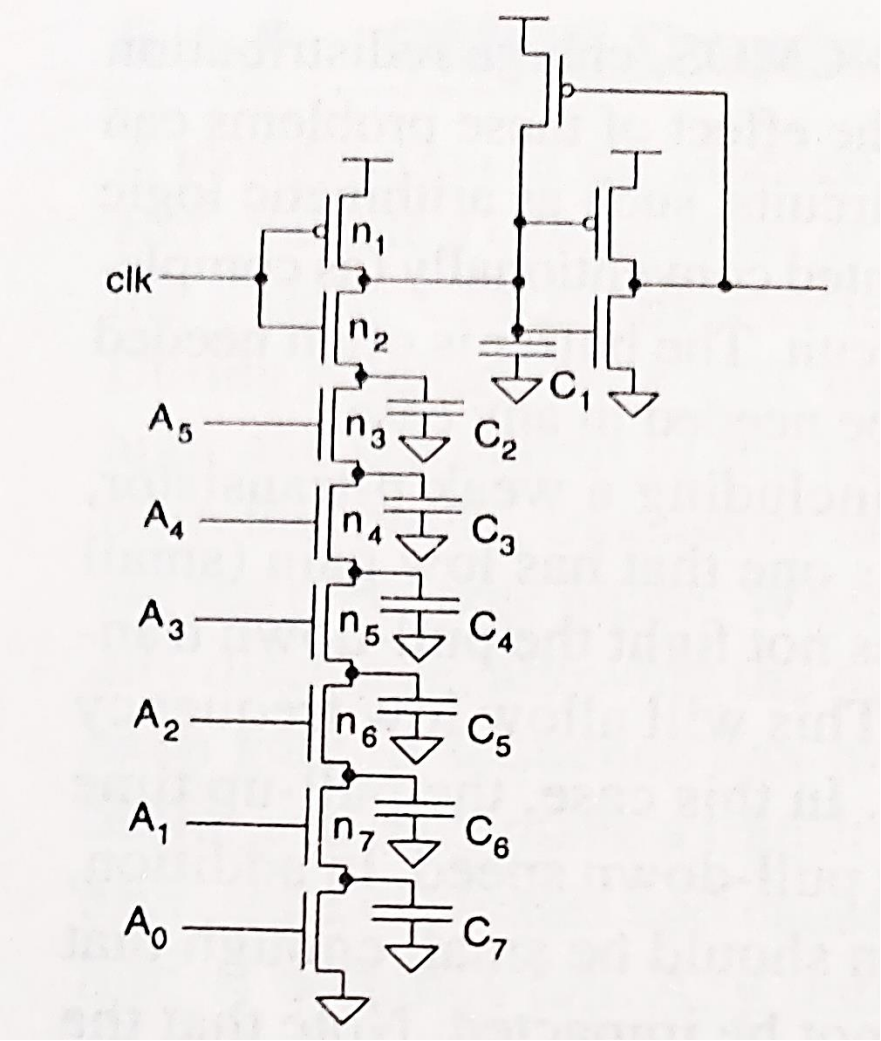
1. Each gate must be buffered
2. Only noninverting structures are possible
3. Charge redistribution is still a problem

Converting Domino Gate into Static

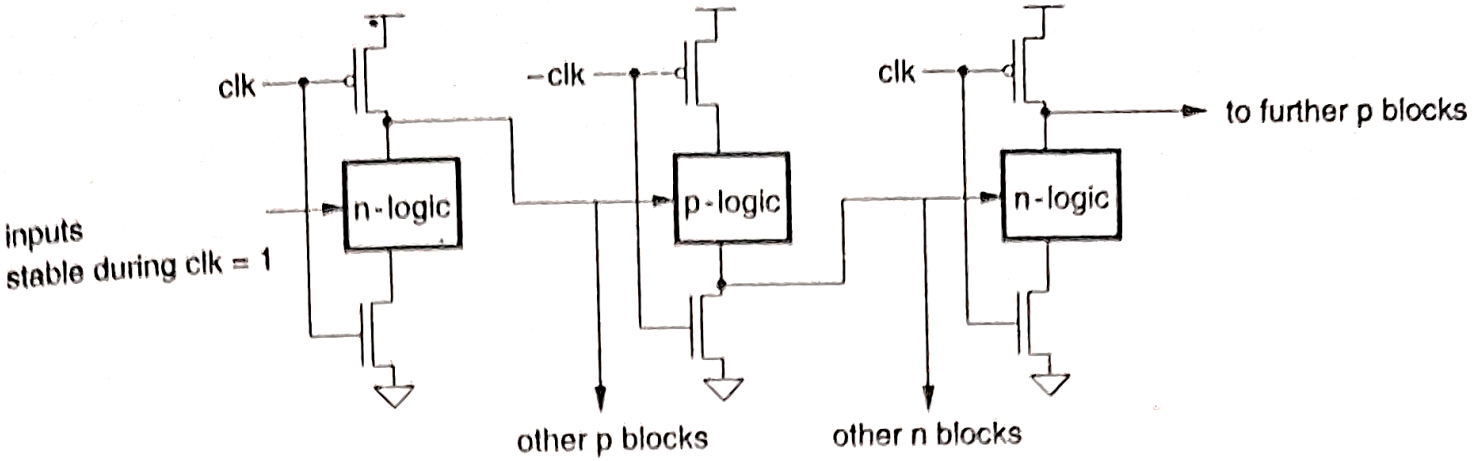
- This allows low frequency or static operation
- Precharge transistor may be eliminated if the time between evaluation phases is long enough to allow the weak pull-up to charge the output node.



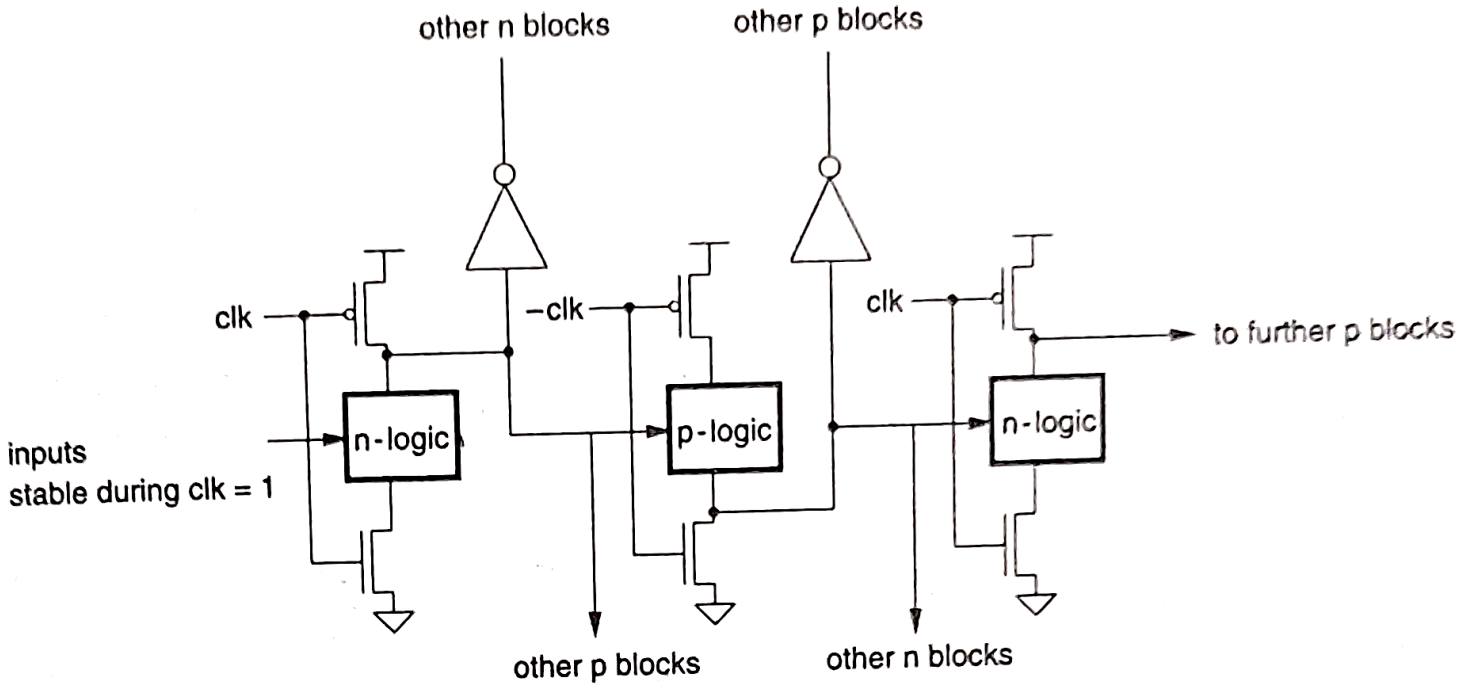
What happens if clocked n-transistor is placed closest to the output?



NP Domino Logic (Zipper CMOS)


clk
n-logic
inputs
stable during clk = 1
–clk
p-logic
clk
n-logic
to further p blocks
other p blocks
other n blocks

(a)


other n blocks
other p blocks
clk
n-logic
inputs
stable during clk = 1
–clk
p-logic
clk
n-logic
to further p blocks
other p blocks
other n blocks

(b)

Advantages of Dynamic Logic

- Smaller area than fully static gates
- Smaller parasitic capacitances, hence higher speed
- Glitch free operation if designed carefully