EE141-Spring 2010 Digital Integrated Circuits

Lecture 14
Complex CMOS - Cntd

Administrativia

- Hw 5 due Today.
- Hw 6 posted early next week. You get TWO weeks for this one.
- Project phase 1 will be launched today
- Out of town next week
  - We lecture offered by Stanley
  - Fr lecture cancelled – Make-up on Tu March 16 at 3:30pm

Class Material

- Last lecture
  - Inverter energy
  - Project launch
- Today's lecture
  - Optimizing complex CMOS
  - Pass Transistor Logic
- Reading (Ch 5, 6)

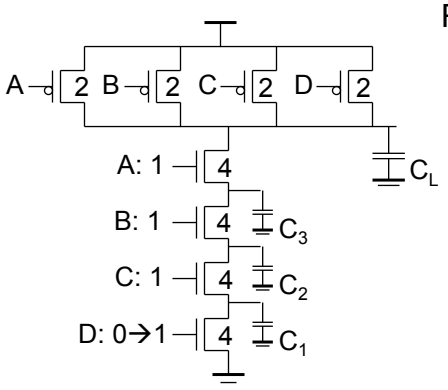
CMOS Logic Revisited

Analyzing and Optimizing Complex CMOS Gates

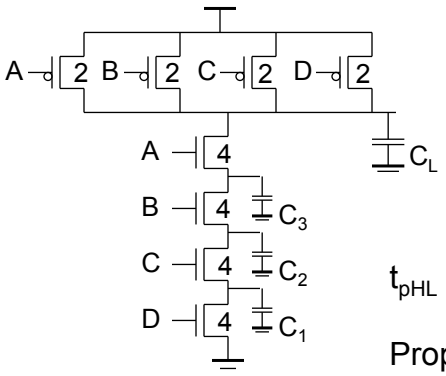
- Techniques very similar to the inverter case
- Logical Effort technique as the means for gate sizing and topology optimization
- However … some other things to be aware of

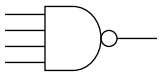
Fan-In Considerations

RC model:



Fan-In Considerations



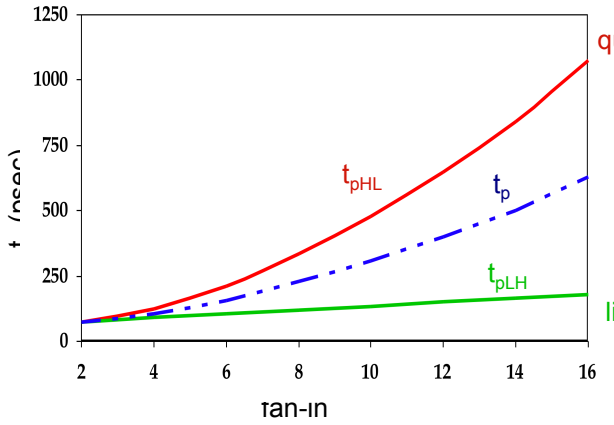


Distributed RC model (Elmore delay)

$t_{pHL} = 0.69\ R_{eqn}(C_1+2C_2+3C_3+4C_L)$

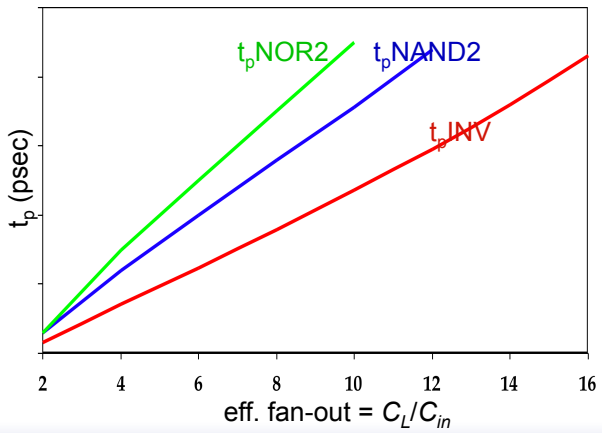
Propagation delay deteriorates rapidly as a function of fan-in – quadratically in the worst case.

t_p as a Function of Fan-In



Gates with a fan-in greater than 4 should be avoided.

t_p as a Function of Fan-Out



All gates have the same drive current.

Slope is a function of "driving strength"

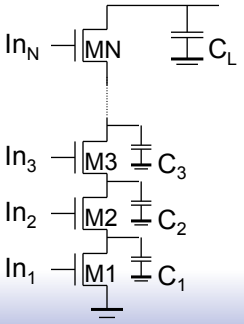
t_p as a Function of Fan-In and Fan-Out

- Fan-in: quadratic due to increasing resistance and capacitance
- Fan-out: each additional fan-out gate adds two gate capacitances to C_L

$$t_p = a_1 FI + a_2 FI^2 + a_3 FO$$

Fast Complex Gates: Design Technique 1

- Transistor sizing
  - as long as fan-out capacitance dominates
- Progressive sizing



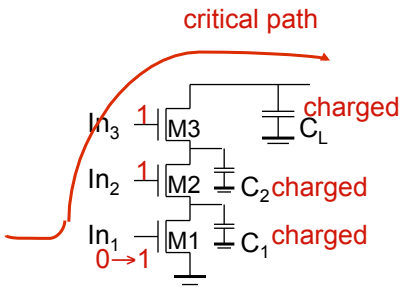
Distributed RC line

M1 > M2 > M3 > … > MN
(the FET closest to the output is the smallest)

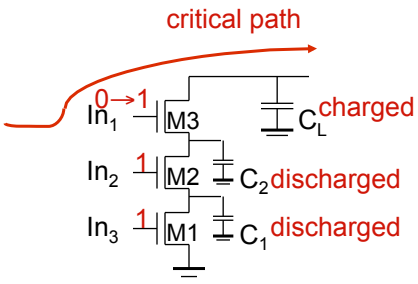
Can reduce delay by more than 20%;
Be careful: input loading, junction caps, decreasing gains as technology shrinks

Fast Complex Gates: Design Technique 2

- Transistor ordering



delay determined by time to discharge C_L, C_1 and C_2

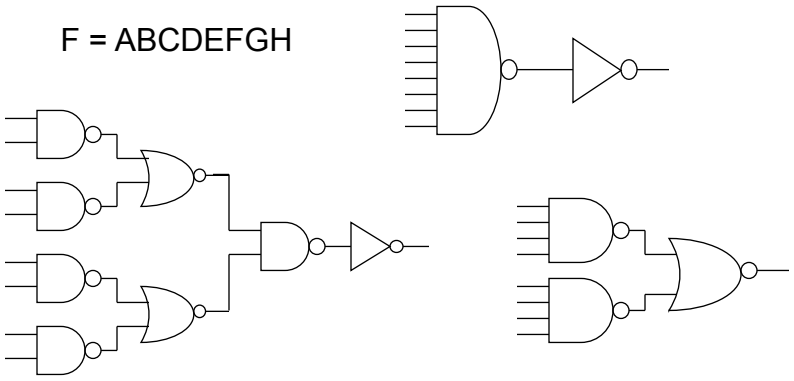


delay determined by time to discharge C_L

Fast Complex Gates: Design Technique 3

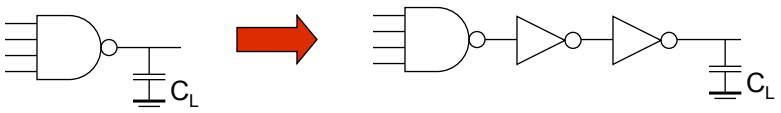
- Alternate logic structures

F = ABCDEFGH



Fast Complex Gates: Design Technique 4

- Isolating fan-in from fan-out using buffer insertion



Fast Complex Gates: Design Technique 5

- Reducing the voltage swing

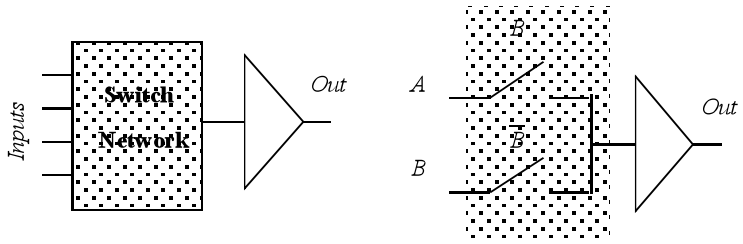
$$t_{pHL} = 0.5\,(C_L\,V_{DD}) / I_{DSATn}$$

$$= 0.5\,(C_L\,V_{swing}) / I_{DSATn}$$

  - linear reduction in delay
  - also reduces power consumption
- But the following gate is slower!
- Or requires use of "sense amplifiers" on the receiving end to restore the signal level (memory design)

Pass-Transistor Logic

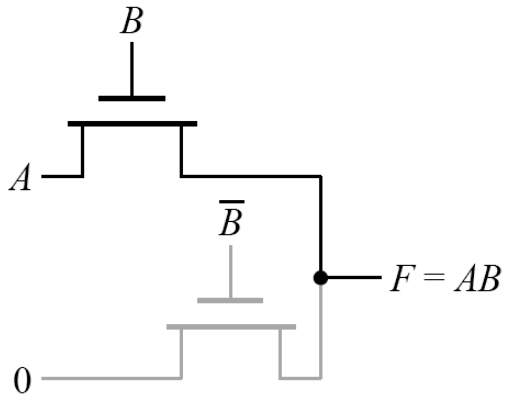
Pass-Transistor Logic



- N transistors
- No static consumption

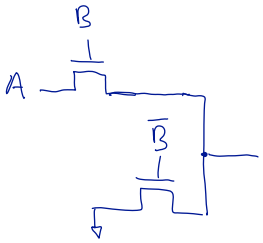
Example: AND Gate

Example: AND Gate



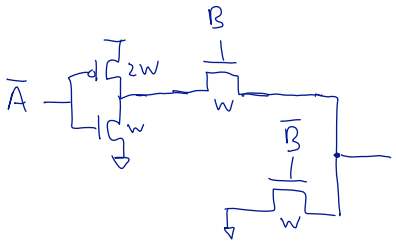
Pass Transistor Logic LE

- What is LE of "gate" shown below for A and B inputs?
  - Hint: Can you answer this question with only the information shown below?



Pass Transistor Logic LE

- In CMOS, a "gate" is defined only when trace a connection all the way back to a supply
  - Otherwise don't know what drive resistance really is

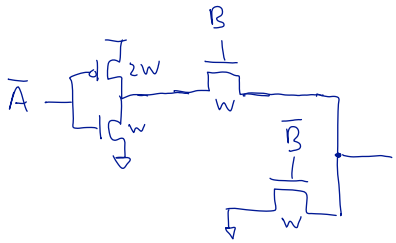


$C_{gate\bar{A}} =$

$R_{gate\bar{A}} =$

$LE_{\bar{A}} =$

Pass Transistor Logic LE

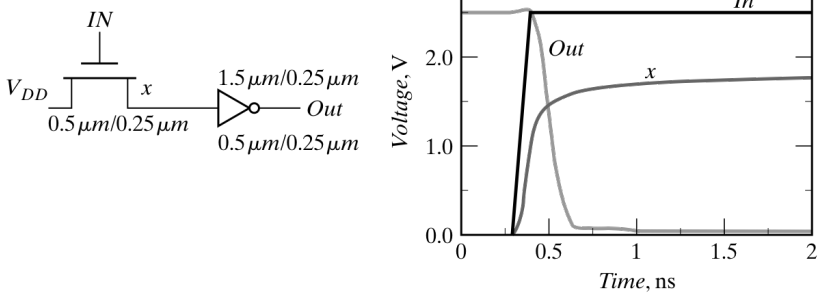


$C_{gate B} =$

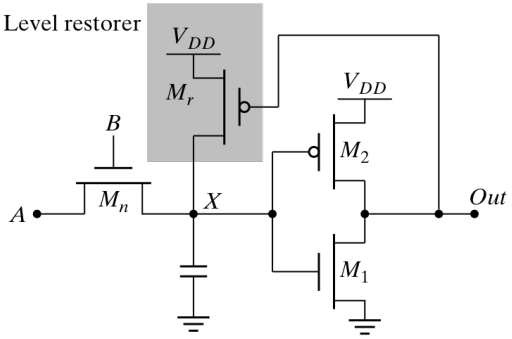
$R_{gate B} =$

$LE_B =$

NMOS-Only Logic

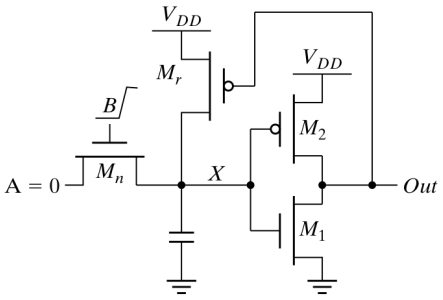


NMOS Only Logic: Level Restoring Transistor



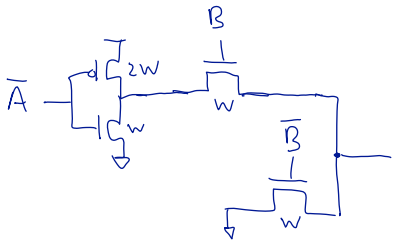
- Advantage: Full Swing
- Restorer adds capacitance, takes away pull down current at X
- Ratio problem

Restorer Sizing



- Upper limit on restorer size
- Pass-transistor pull-down can have several transistors in stack

Pass Transistor Logic LE

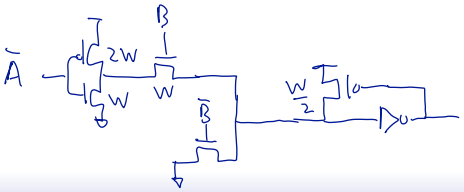


$C_{gate\,B}$ =

$R_{gate\,B}$ =

LE_B =

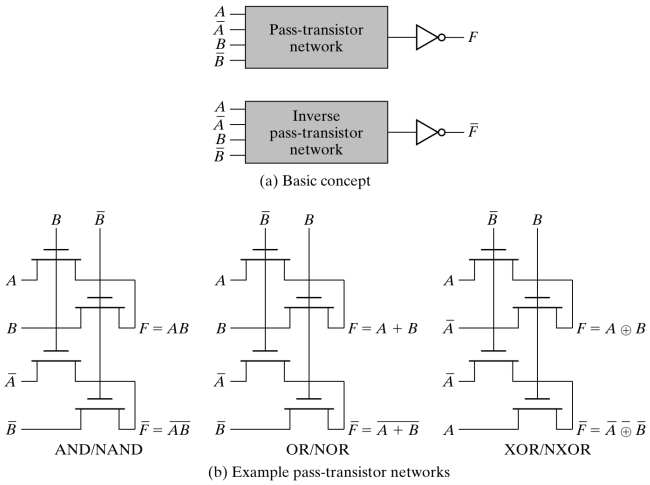
With level restore:



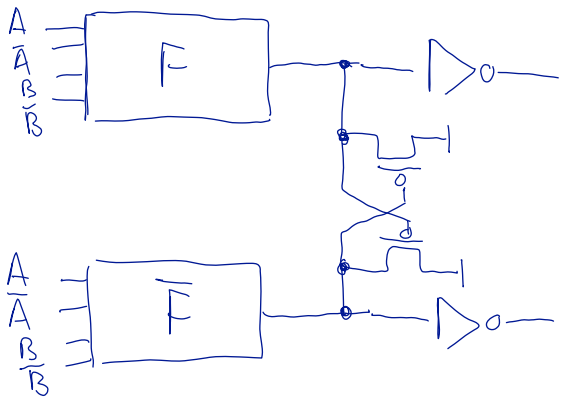
$R_{gate\,B}$ =

LE_B =

Complementary Pass Transistor Logic

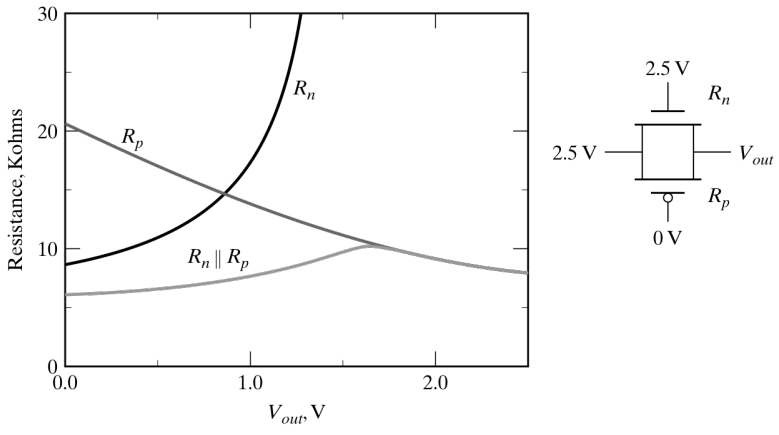


CPL Level Restore

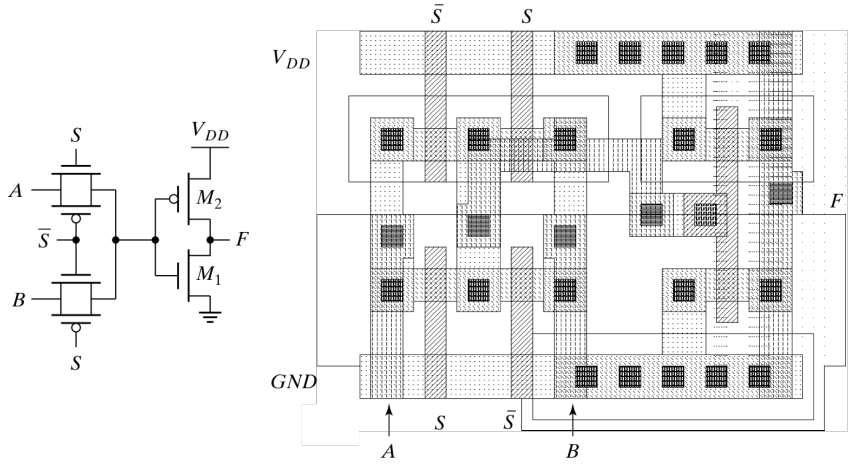


Solution 2: Transmission Gate

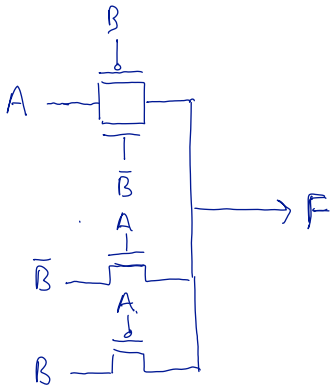
Resistance of Transmission Gate



Pass-Transistor Based Multiplexer

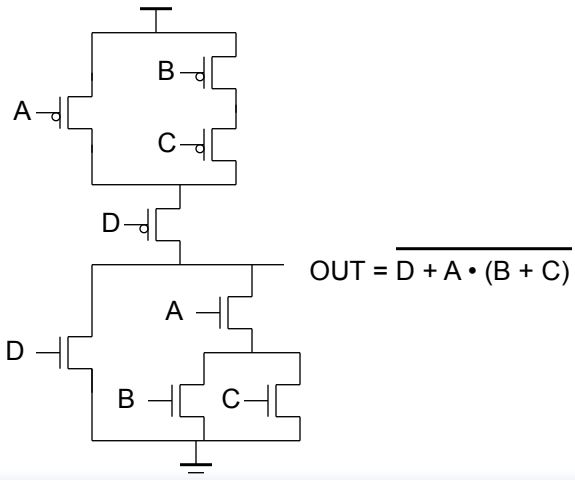


Transmission Gate XOR



CMOS Layout

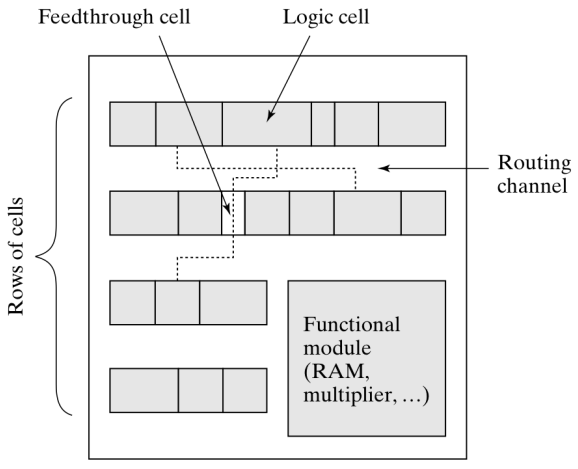
Complex CMOS Gate



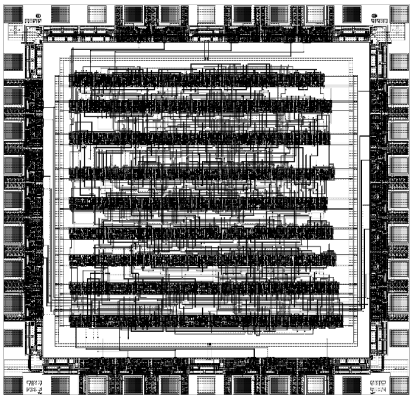
Cell Design

- Standard Cells
  - General purpose logic
  - Used to synthesize RTL/HDL
  - Same height, varying width
- Datapath Cells
  - For regular, structured designs (arithmetic)
  - Includes some wiring in the cell

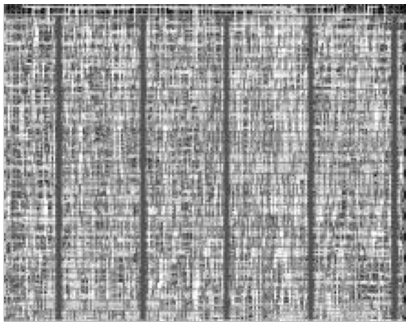
Standard Cell Methodology



Standard Cells – Then and Now

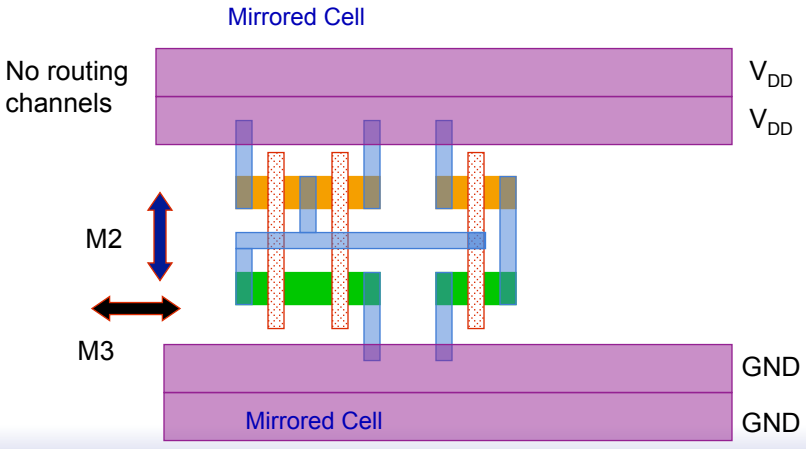


(a)

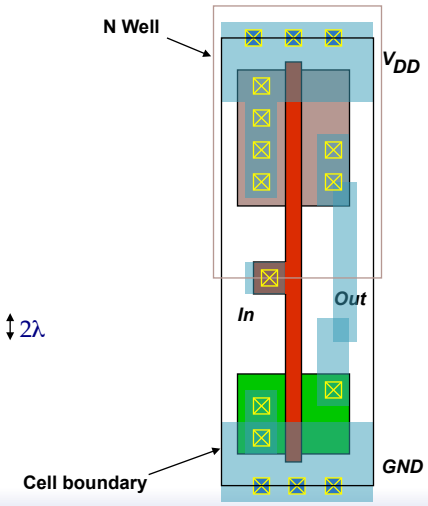


(b)

Standard Cell Layout Methodology – 1990s - Today



Standard Cells



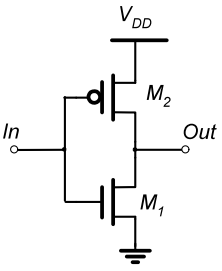
Cell height 12 metal tracks
Metal track is approx. 3λ + 3λ
Pitch =
repetitive distance between objects

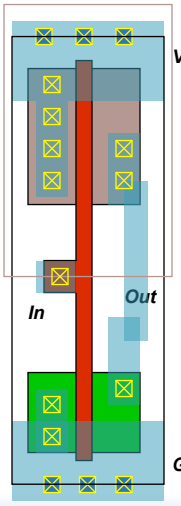
Cell height is "12 Mn pitch"

Rails ~10λ

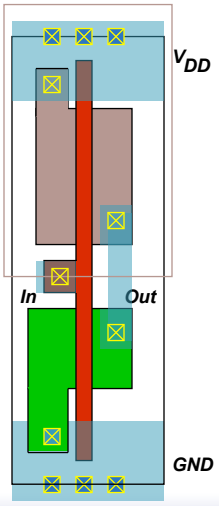
Standard Cells

With minimal diffusion routing

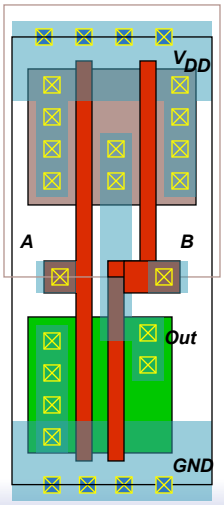




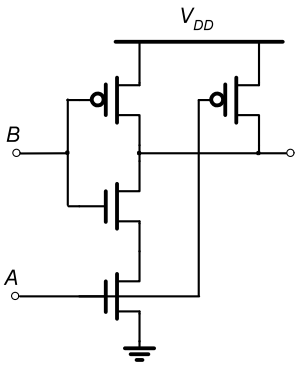
With silicided diffusion



Standard Cells



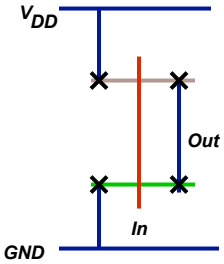
2-input NAND gate



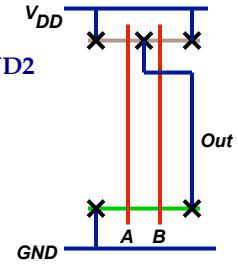
Stick Diagrams

Contains no dimensions
Represents relative positions of transistors

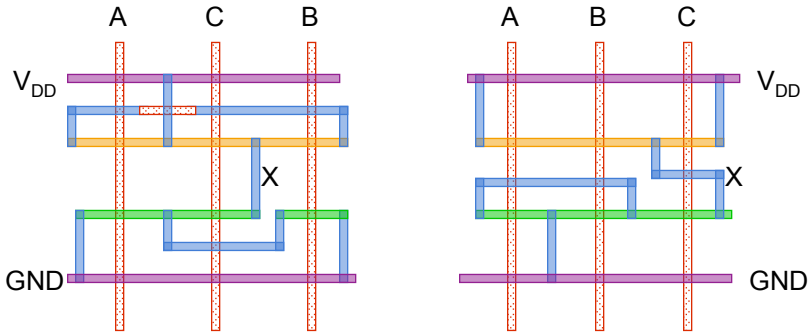
Inverter



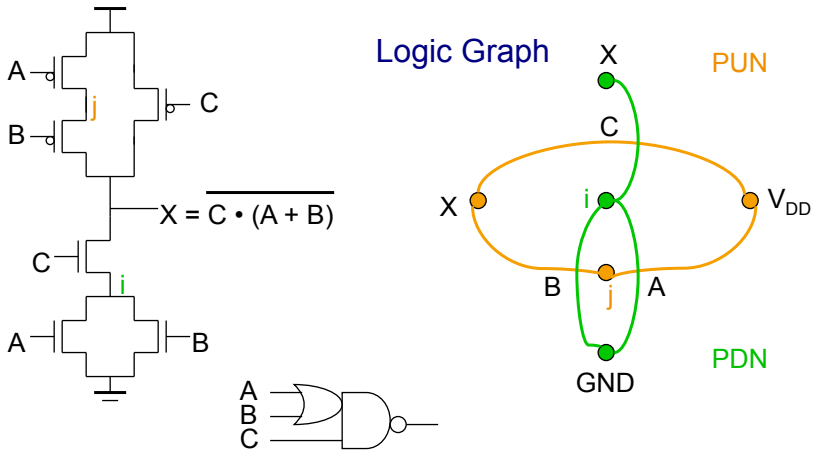
NAND2



Two Versions of C • (A + B)



Logic Graphs



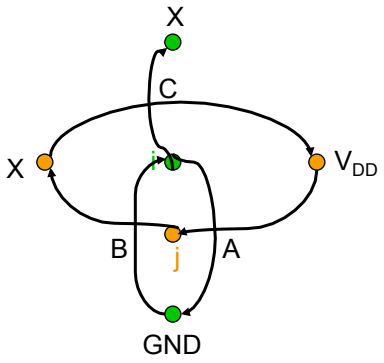
Consistent Euler Path

A B C

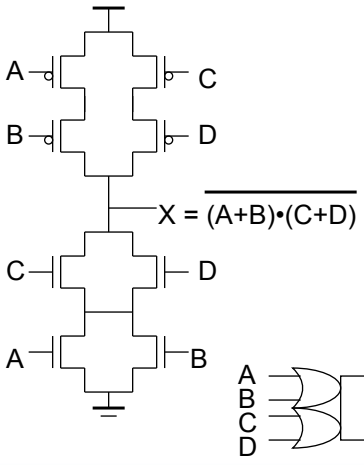
Has PDN and PUN

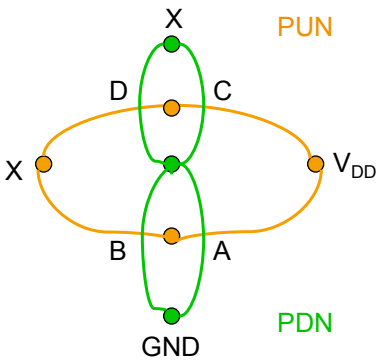
B C A

Has PUN, but no PDN

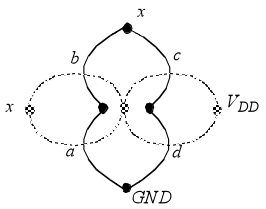


OAI22 Logic Graph

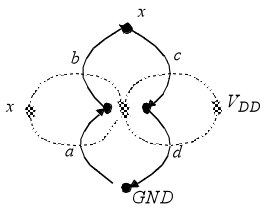




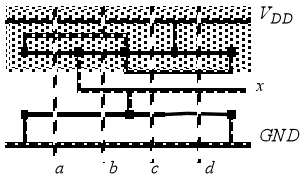
Example: x = ab+cd



(a) Logic graphs for $\overline{(ab+cd)}$



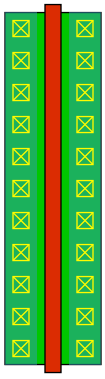
(b) Euler Paths {a b c d}



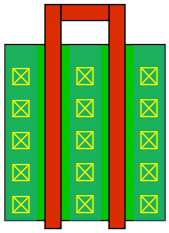
(c) stick diagram for ordering {a b c d}

Multi-Fingered Transistors

One finger



Two fingers (folded)



Less diffusion capacitance