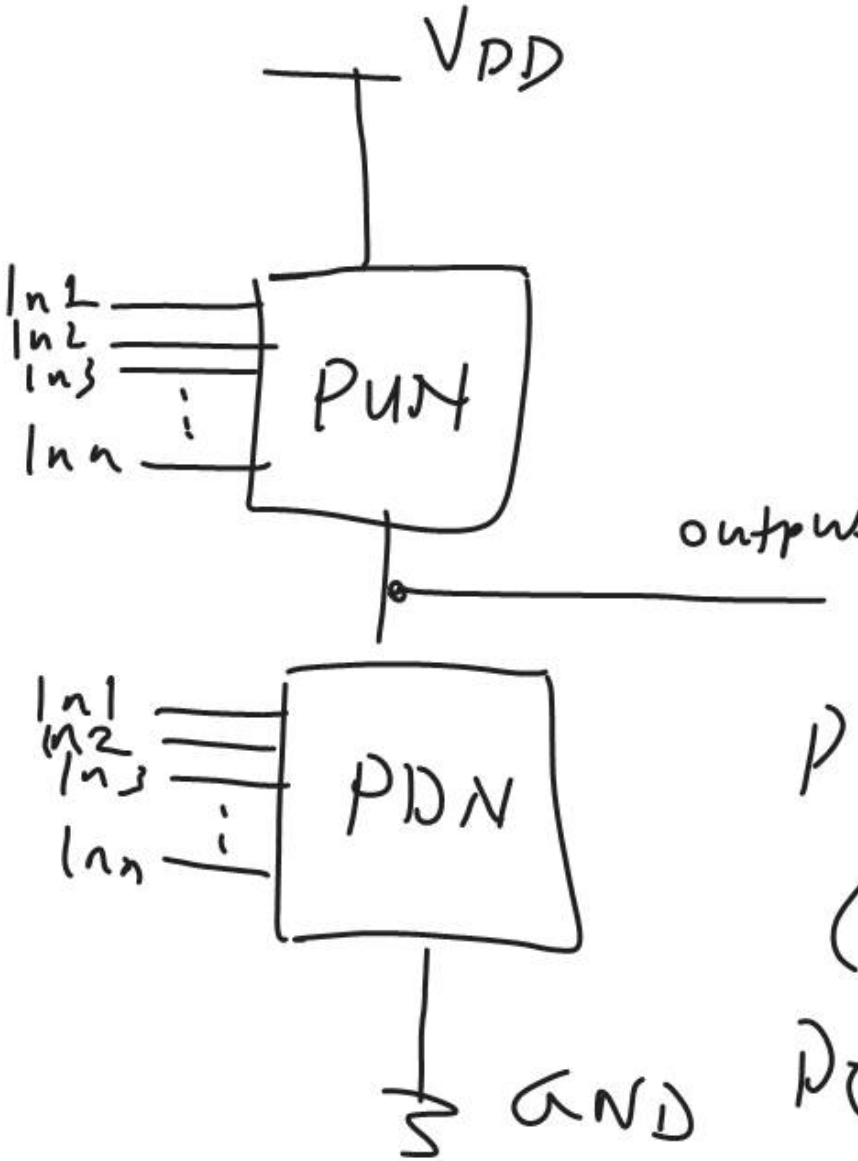
CMOS COMBINATIONAL NETWORKS

03.12.2010 ©



PUN = Pull-up Network

PDN = Pull-down "

PUN, PDN are Dual

(Duality)

De-Morgan Theorems

Boolean Algebra

$\overline{\overline{x}} = x$

$1 + x = 1$, $1 \cdot x = x$, $x + x = x$

$0 \cdot x = 0$ $\quad$ $0 + x = x$ $\quad$ $x \cdot x = x$

$\overline{x+y} = \overline{x} \cdot \overline{y}$

$\overline{xy} = \overline{x} + \overline{y}$

} De-Morgan Theorem

If the DUAL of the statement is TRUE then it is TRUE

$x + (y \cdot z) = (x+y) \cdot (x+z)$

$x \cdot (y+z) = x \cdot y + x \cdot z$

} Distributive Property

$x + (y+z) = (x+y) + z$ associative property

$x + y = y + x$ commutative

Precedence ⇒ NOT, AND, OR

$$x + xy = x$$
$$x \cdot (x+y) = x$$

$$x + xy = x \cdot \underbrace{(1+y)}_{1} = x$$

$$x \cdot (x+y) = \underbrace{x \cdot x}_{x} + x \cdot y = x + xy = x(\underbrace{1+y}_{2}) = x$$

LOGIC DESIGN

a	b	c	f
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0

→ minterm 1 ⇒ m_1

⟹ m_2

⟹ m_3

Sum of Products form (SOP)

$$f(a,b,c) = \sum_{i=1}^{n} m_i$$

Karnaugh map

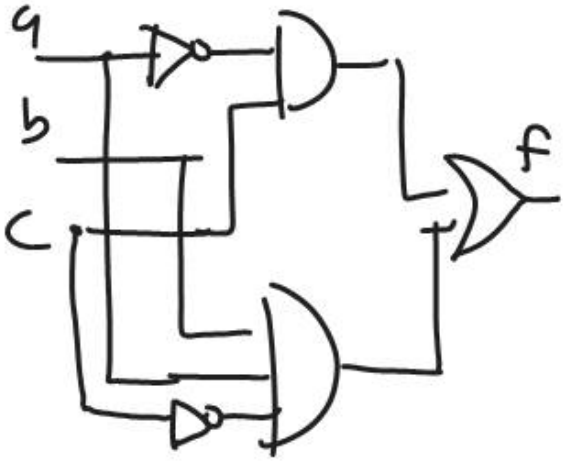
a \ bc	00	01	11	10
0	0	1	1	0
1	0	0	0	1

$$f = \bar{a}c + ab\bar{c}$$

$m_1 = \bar{a}\bar{b}c$

$m_2 = \bar{a}bc$

$m_3 = ab\bar{c}$

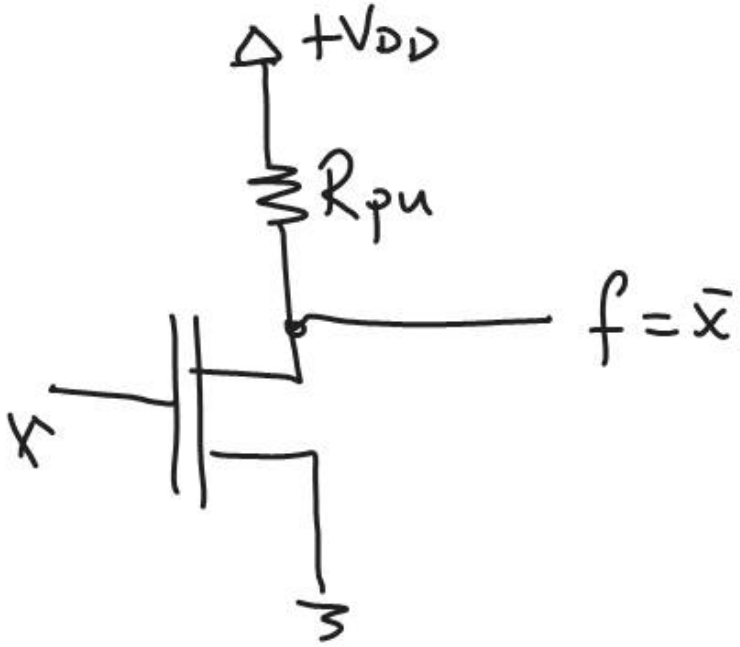


Or, using maxterms

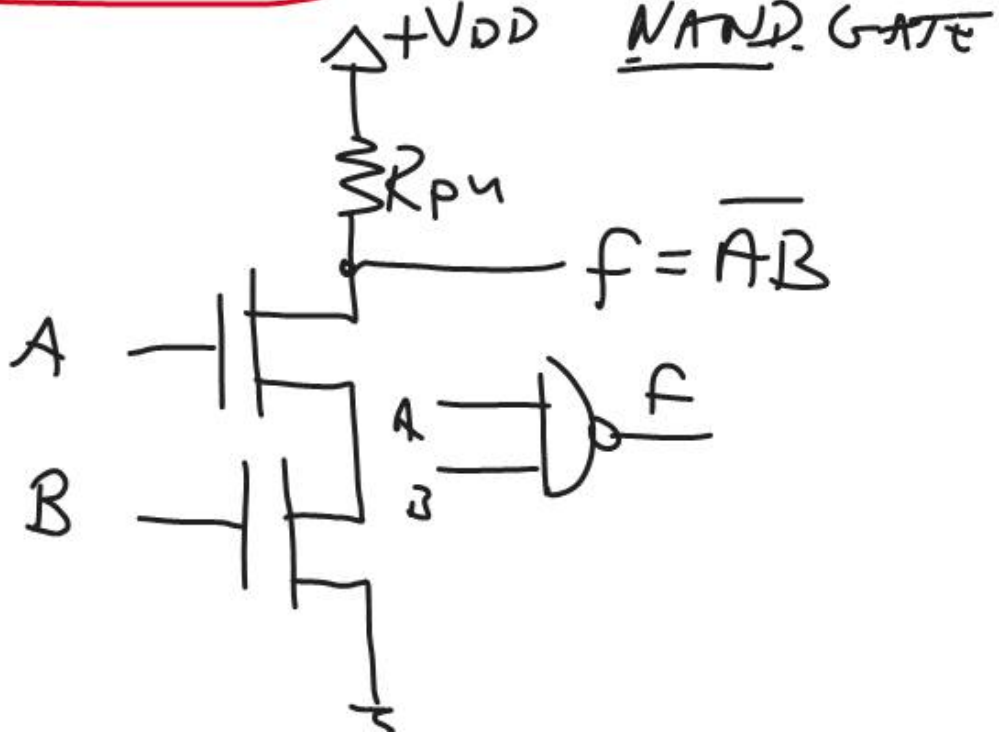
$$f = \prod_{i=1}^{n} M_i$$

Product of Sums form (POS)

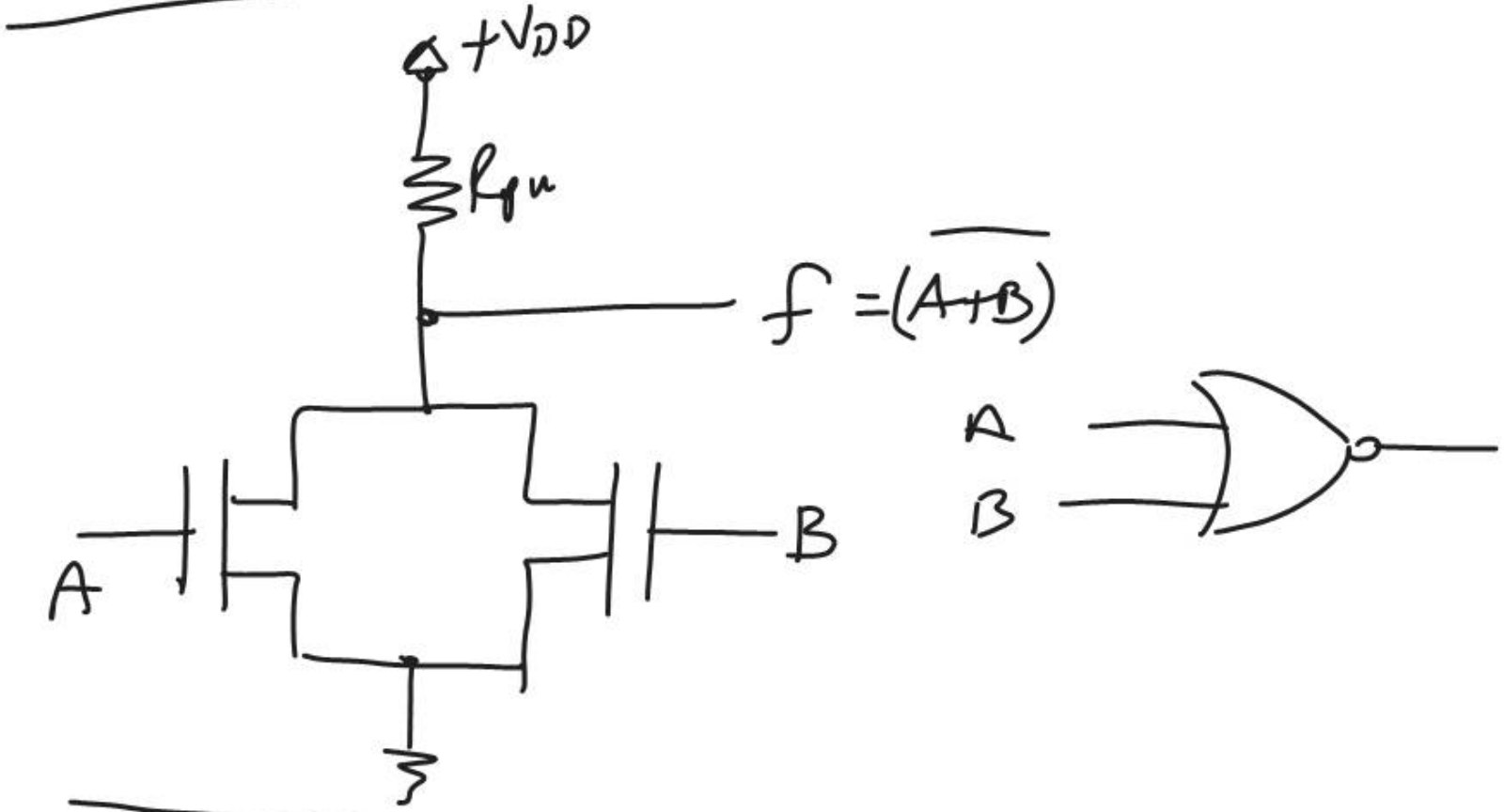
NMOS LOGIC GATES



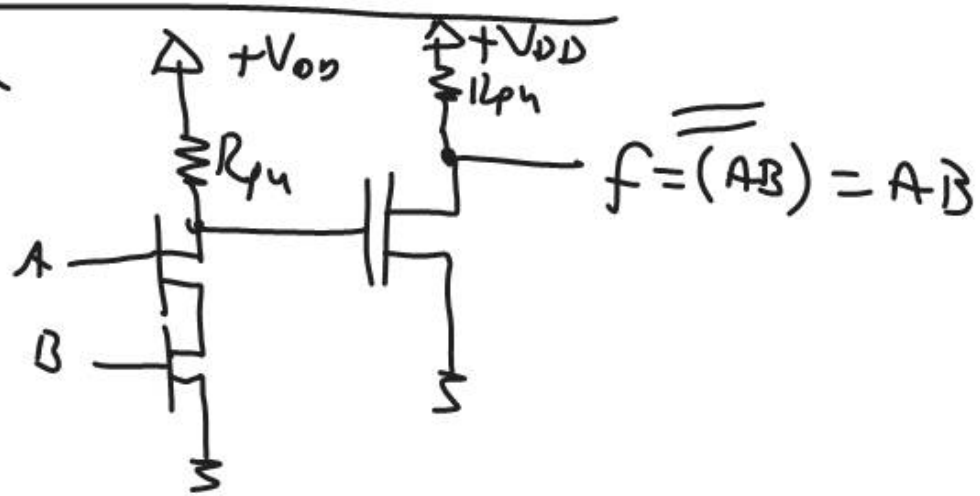
a NAND GATE

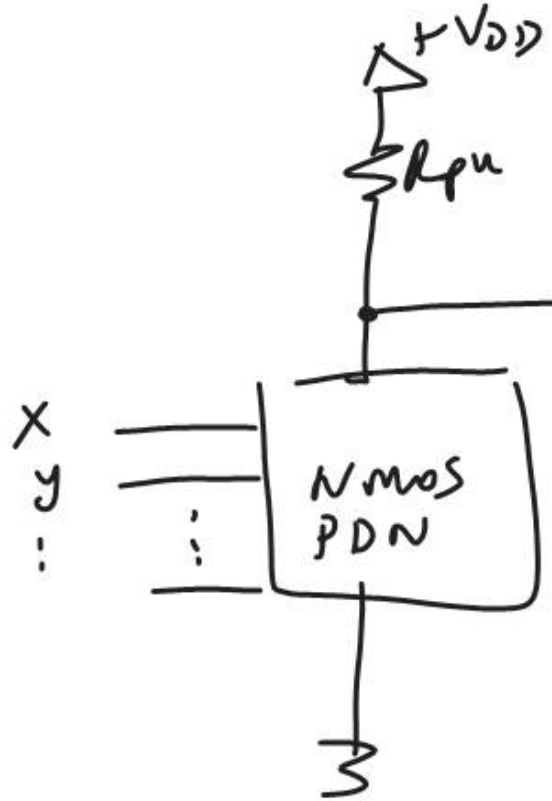


a NOR gate:



an AND Gate

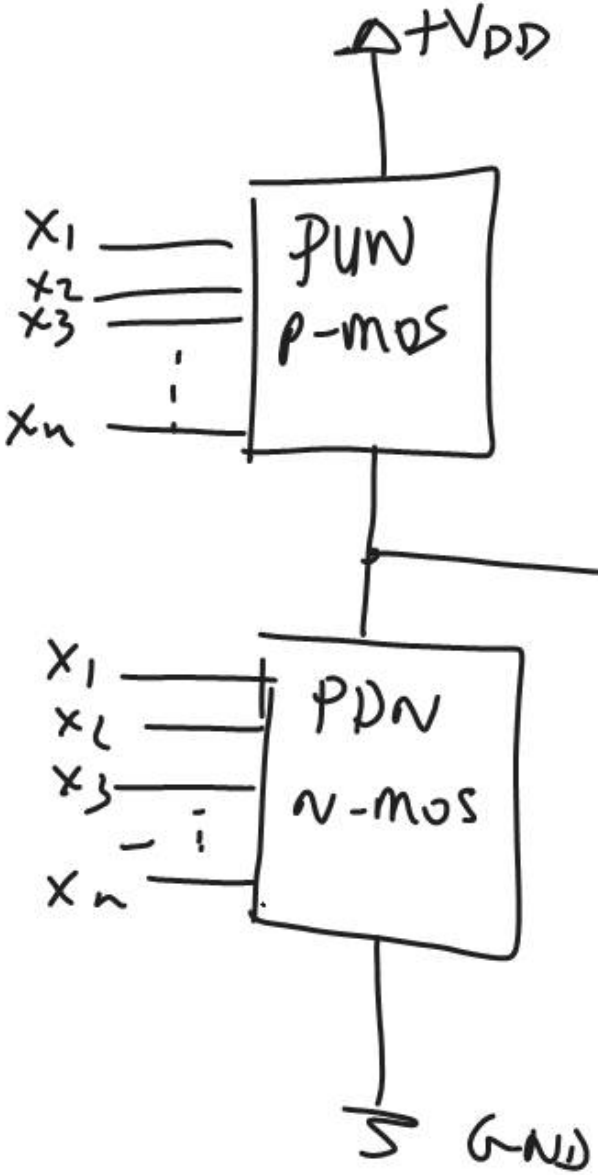




$f = \sum_{i=1}^{n} M_i$ in the SOP form.

However, NMOS logic is not good regarding energy consumption, effectively utilizing the die area and technology etc ---

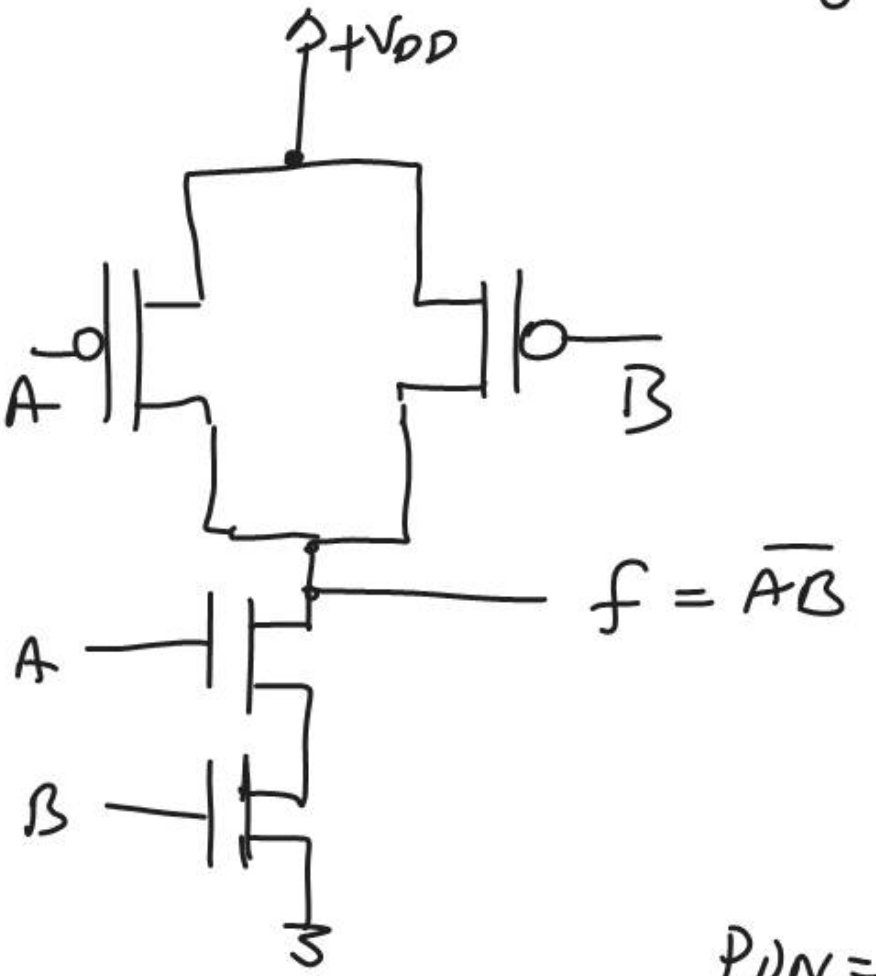
CMOS LOGIC GATE REALIZATIONS



PUN and PDN are DUAL

$$f(x_1, x_2, \ldots x_n) = \sum_{i=1}^{\hat{}} m_i(x_1, x_2 \ldots x_n)$$

ex a CMOS NAND gate

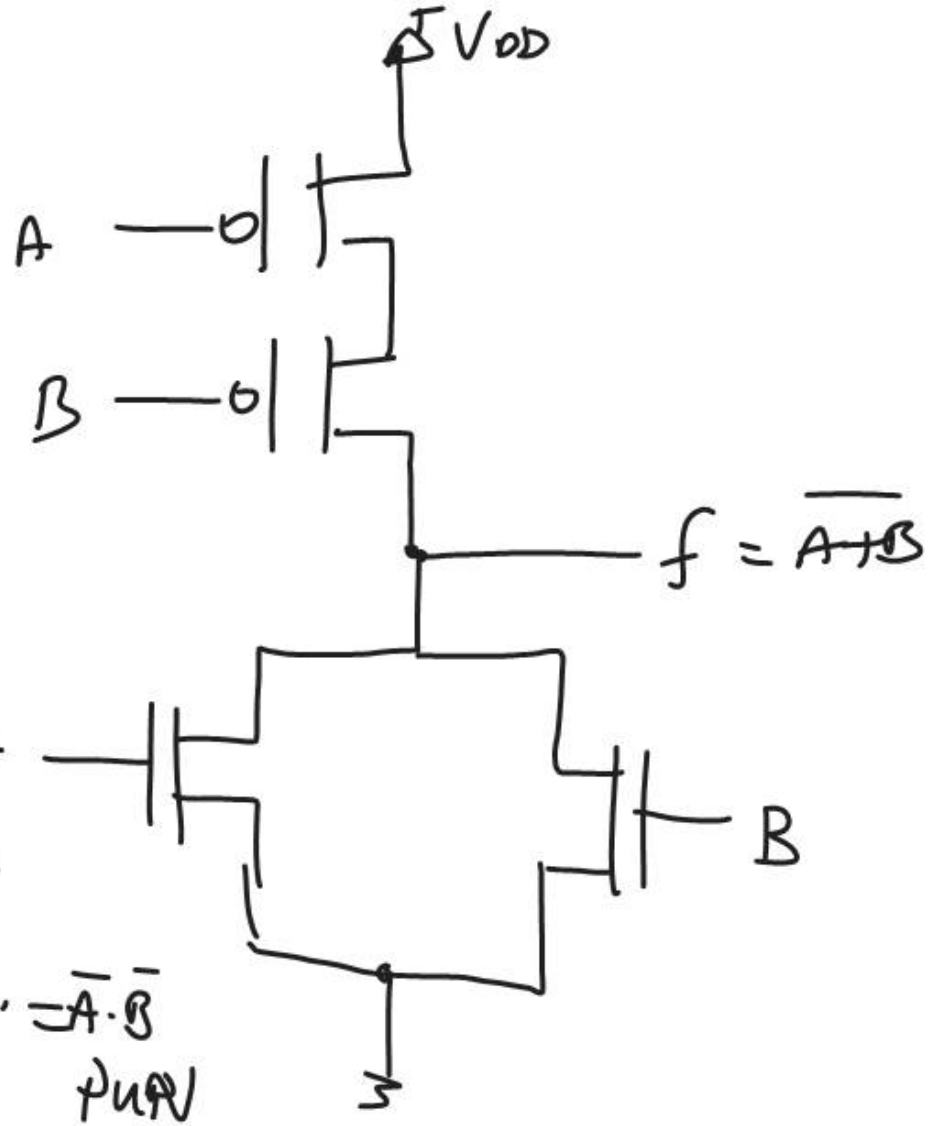


$\overline{AB} = \bar{A} + \bar{B}$

$PDN = AB$

$PUN = \bar{A} + \bar{B}$ DUAL

a NOR gate



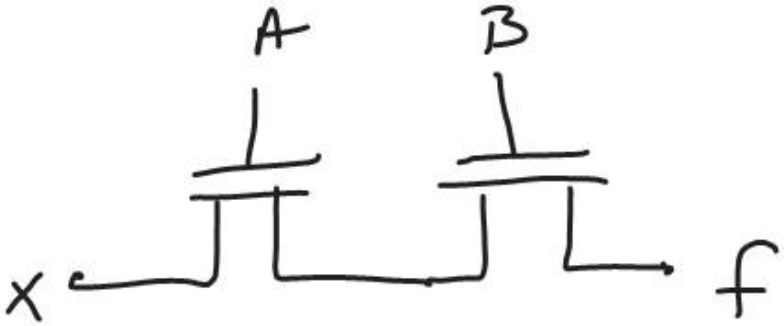
$PDN = A+B$

$\overline{PDN} = \bar{A} \cdot \bar{B}$

PUN

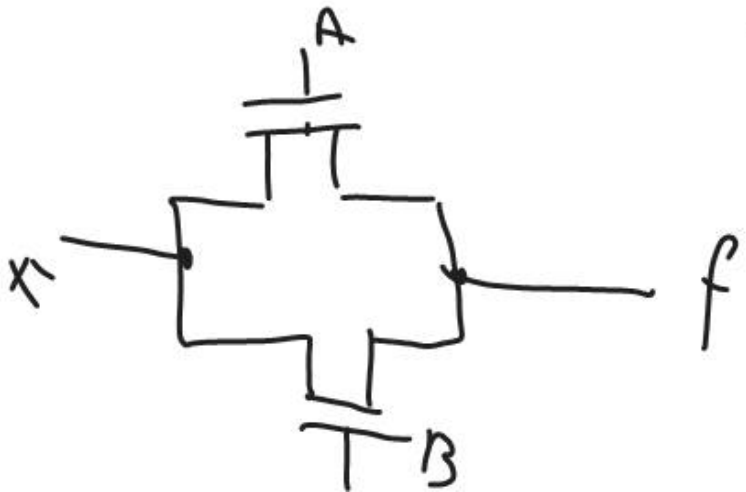
DUAL

NMOS TRANSISTORS in SERIES, PARALLEL CONNECTION



A	B	f
0	0	No change
0	1	// //
1	0	// //
1	1	X

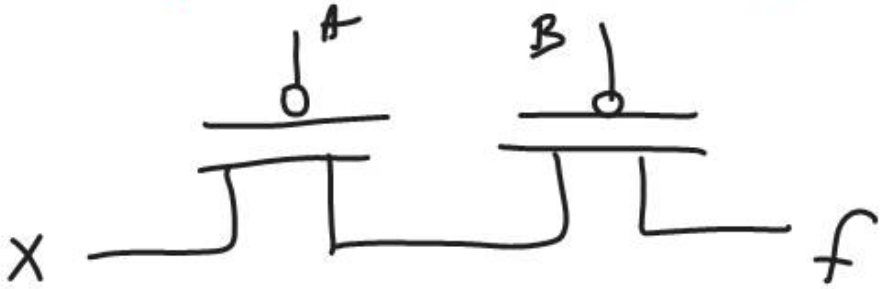
$f = x$ IF $A \cdot B = 1$



A	B	f
0	0	No change
0	1	X
1	0	X
1	1	X

$f = x$ IF $A + B = 1$

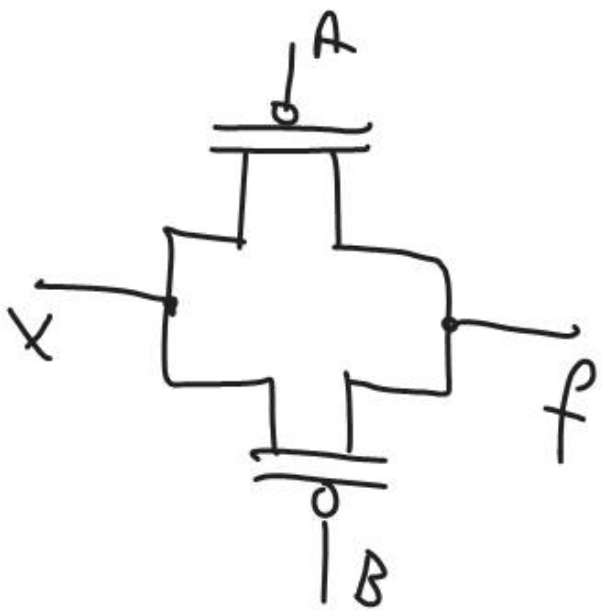
PMOS Transistors



A B	f
0 0	X
0 1	No change
1 0	"
1 1	"

$f = X$ IF $\overline{A + B} = 1$

$\overline{A} \cdot \overline{B} = 1$



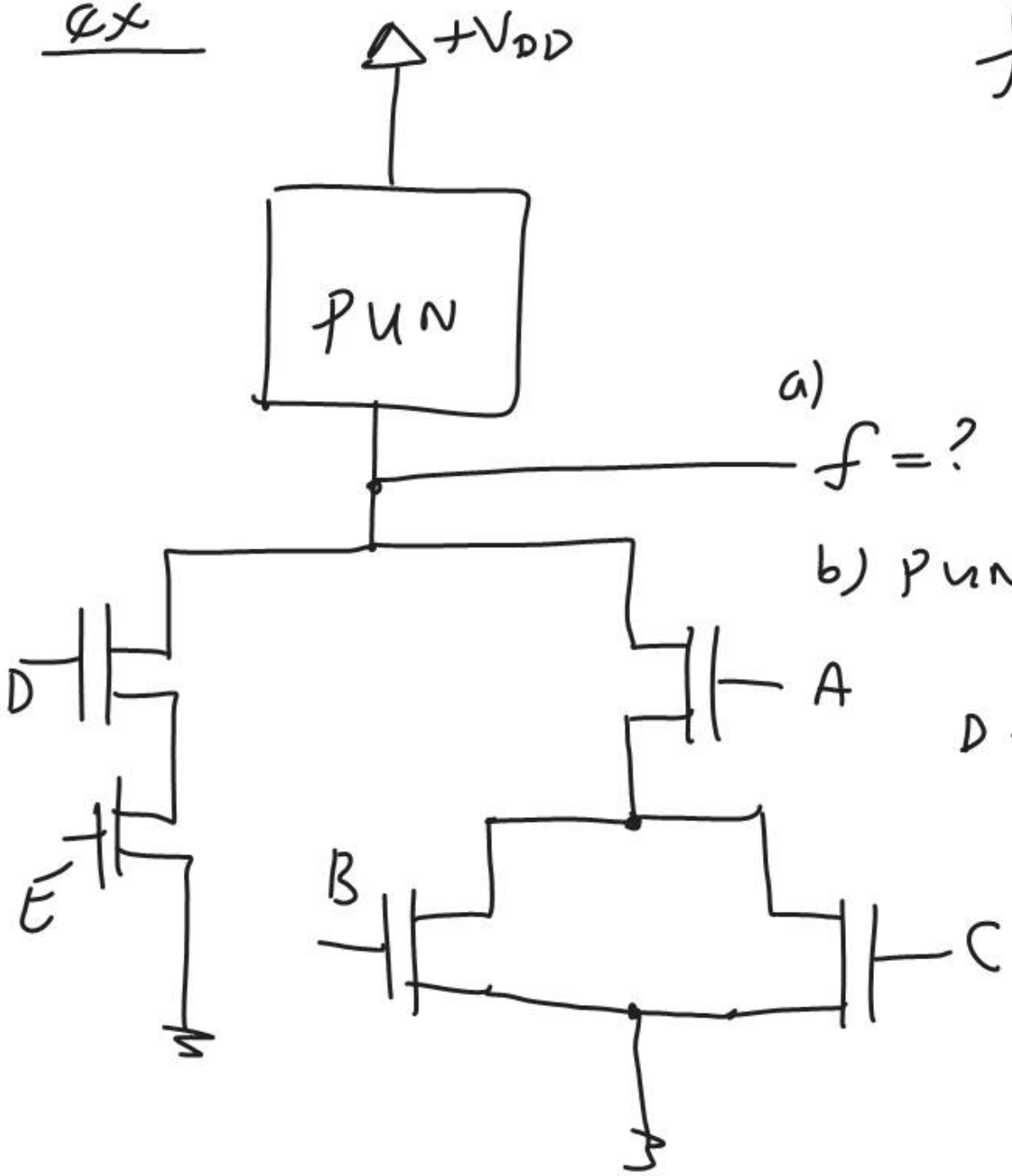
A B	f
0 0	X
0 1	X
1 0	X
1 1	No-change

$f = X$ IF $\overline{A \cdot B} = 1$

$\overline{A} + \overline{B} = \perp$

Ex

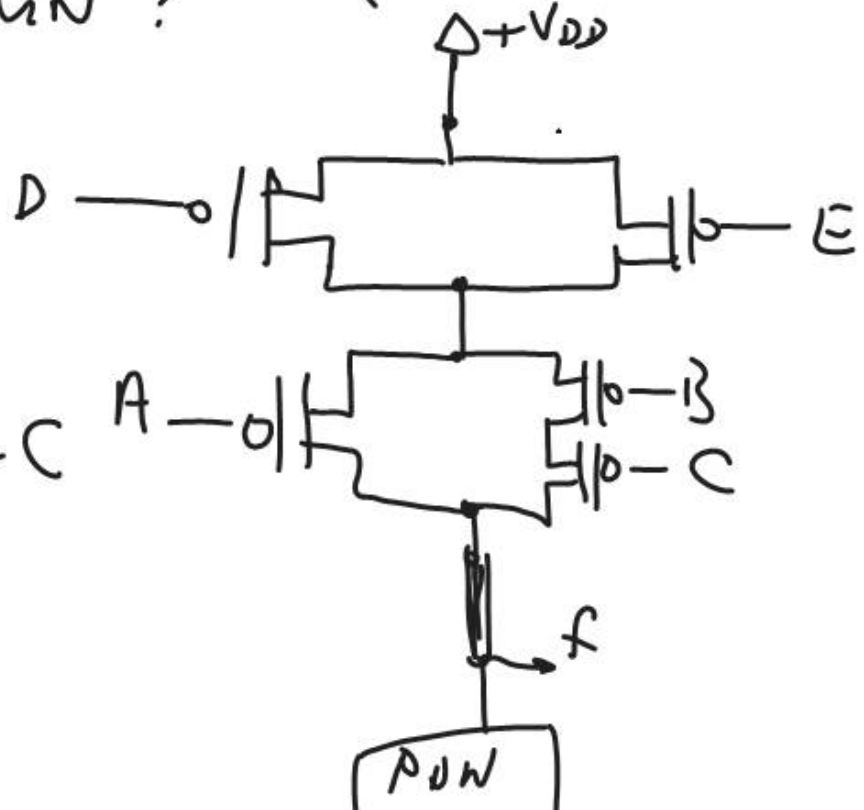
$$f = \overline{A\cdot(B+C) + D\cdot E}$$



$$P_{UN} = \overline{A\cdot(B+C)}\cdot\overline{(DE)}$$

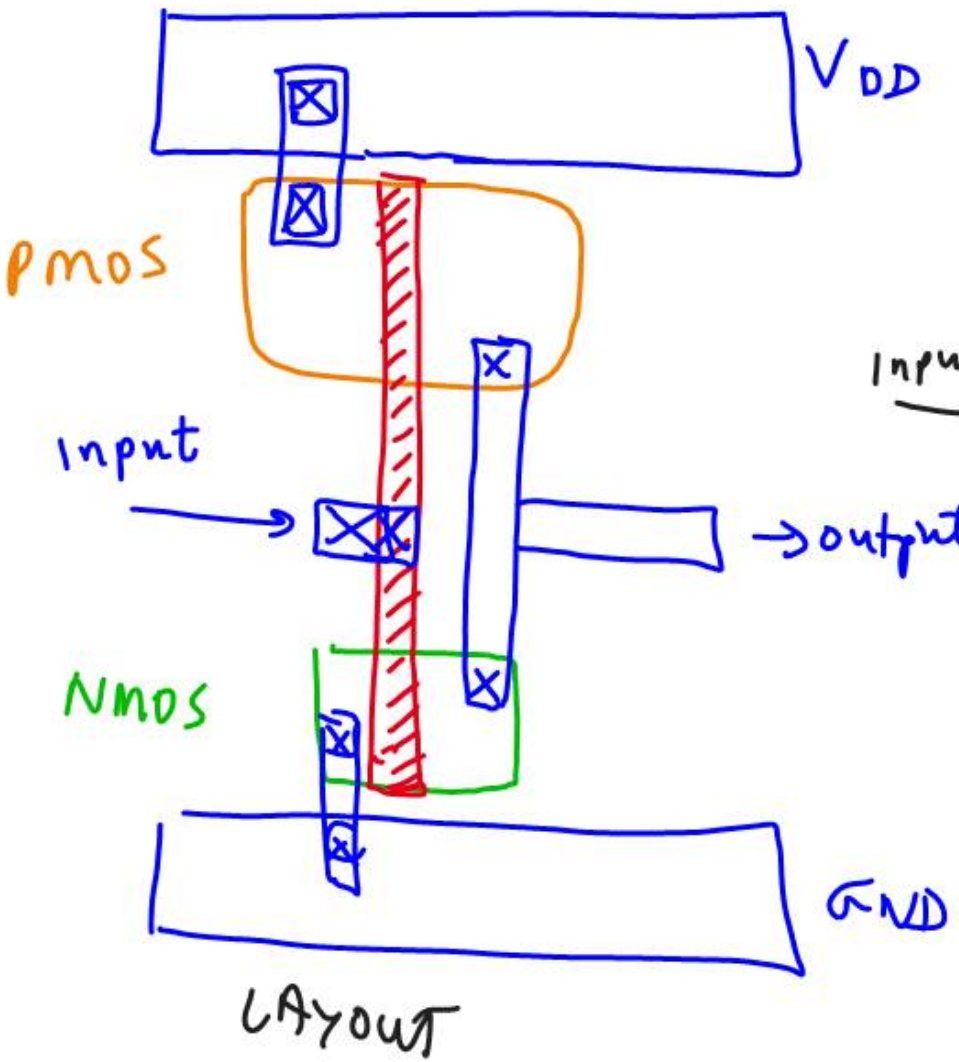
$$= \bar{A} + \overline{(B+C)}\cdot(\bar{D}+\bar{E})$$

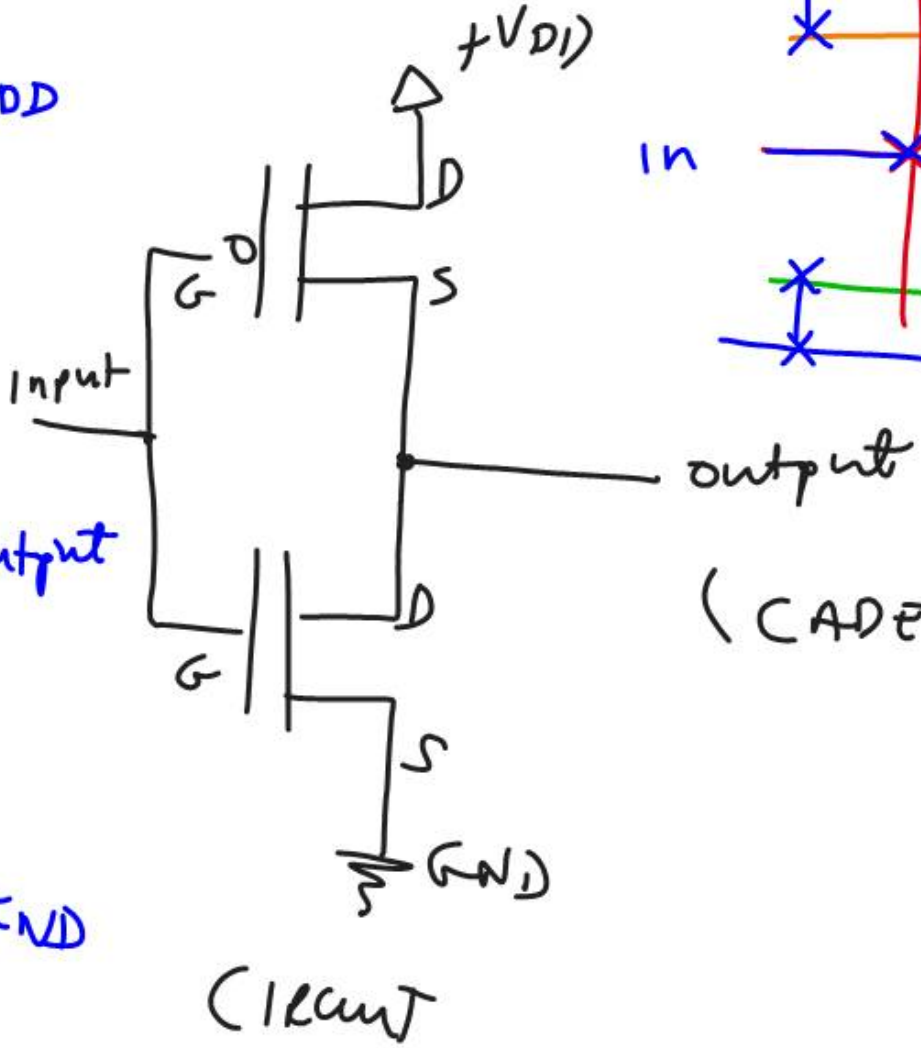
$$= (\bar{A} + \bar{B}\bar{C})\cdot(\bar{D}+\bar{E})$$



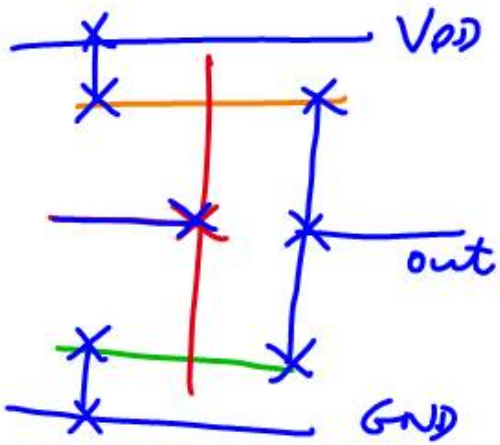
STANDART CELL LAYOUT METHODOLOGY

INVERTER



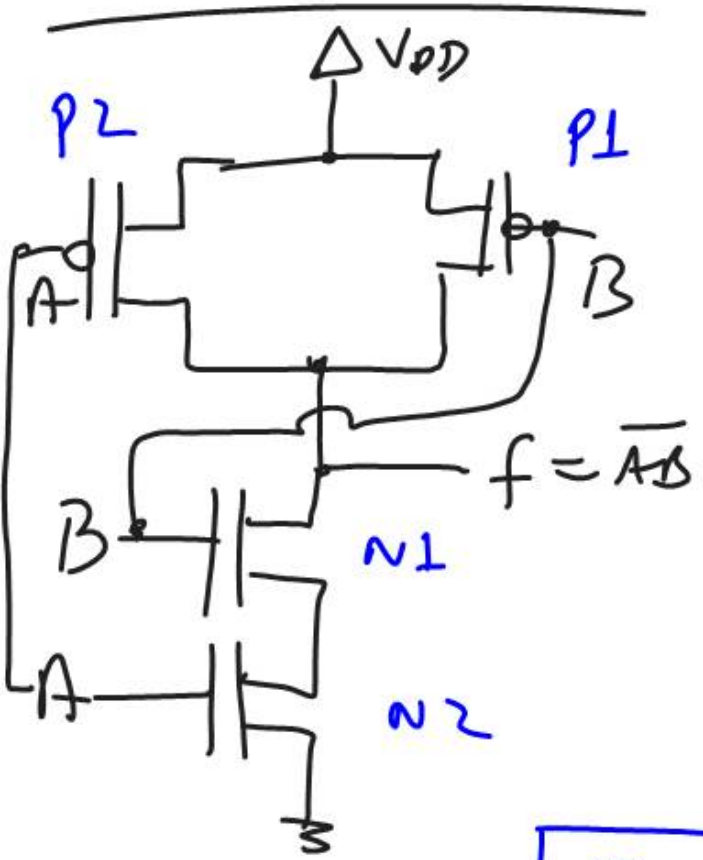


STICK DIAGRAM

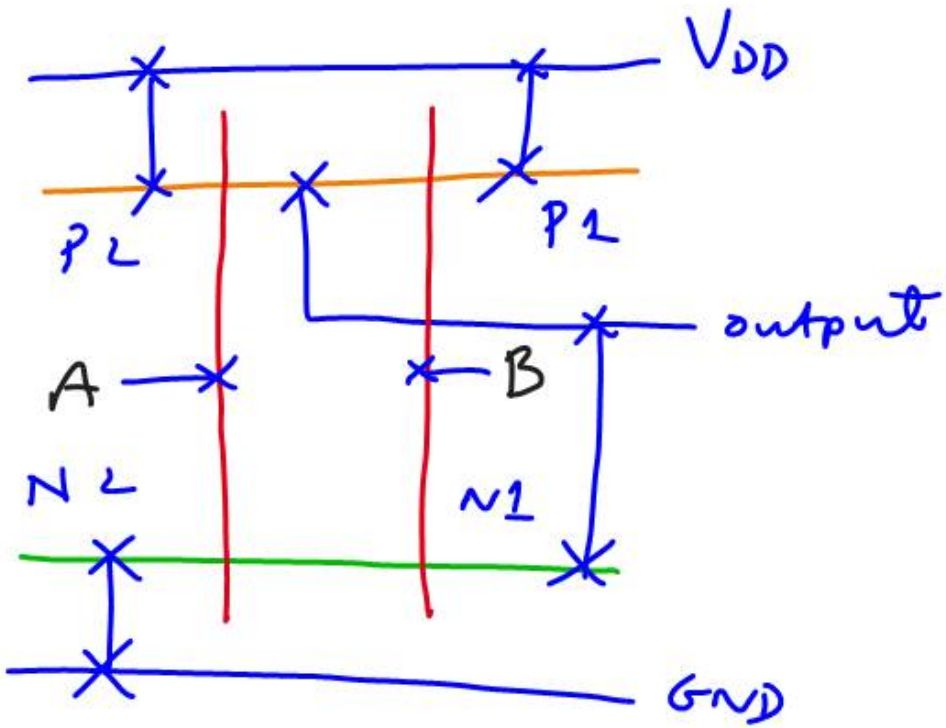


(CADENCE) Tool

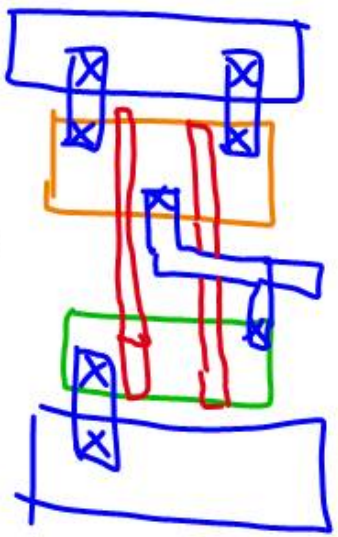
2 NAND



STICK DIAGRAM

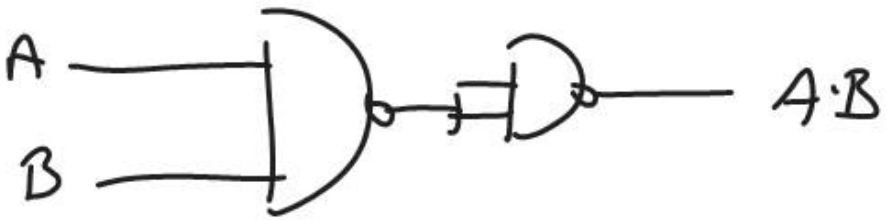


LAYOUT

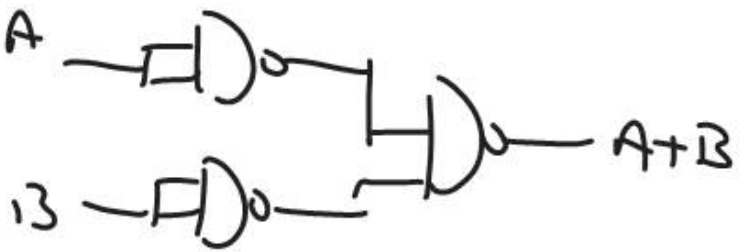


Since a logic function can be obtained by NAND GATES only, we can create the logic functions by copy-pasting NAND layouts and making the proper connections between them.

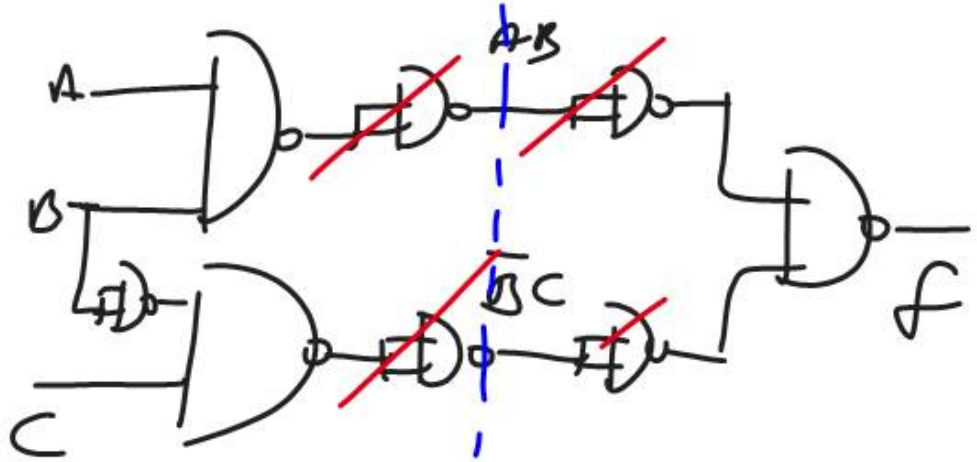
AND



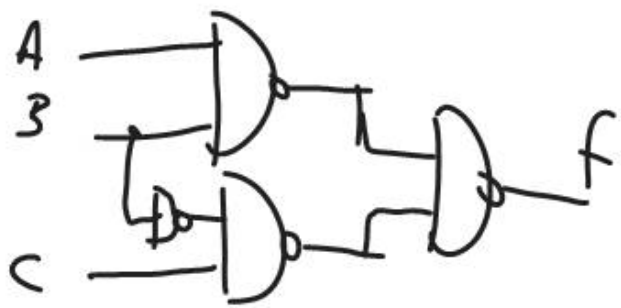
OR



ex $f = AB + \bar{B}C$



AND | OR



SOP form

HW #3

$f = AB + BCD + \bar{A}CD$

implement this function by using NOR gates only.

& draw the stick diagram

next week!