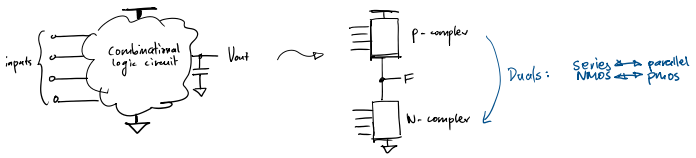
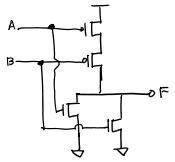
Static CMOS Combinational Logic

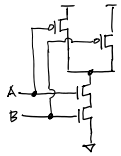
October 15, 2019 18:17



NOR



NAND



General steps for logic f(A,B,C....)

① Simplify function $\bar{f}(A,B,\dots)$
function could be + or ×

② Design NMOS (pull down)
+ is parallel
× is series

③ Design PMOS (pull up) — just make duals: switch series ↔ parallel

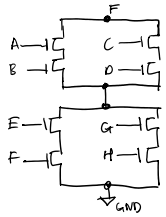
④ Put P complex & N complex on top of each other

⑤ Connect input & output

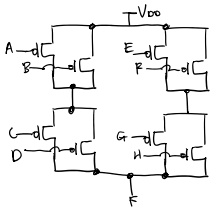
⑥ Size the MOS'es

ex. $(AB+CD)(EF+GH)\rightarrow=(A\times B+C\times D)\times(EF+GH)$

NMOS:



PMOS (Dual):

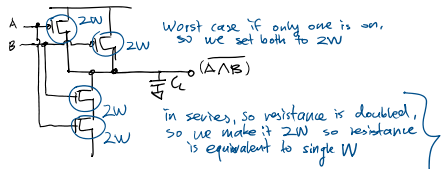


Gate Sizing depends on worst-case scenario (to match with the default inverter)

Default Inverter:

for parallel only one is one (max. resistance)

NAND:

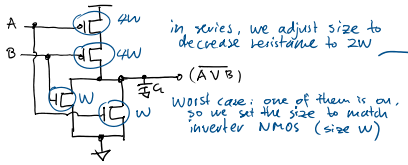


Worst case if only one is on, so we set both to 2W

in series, so resistance is doubled, so we make it 2W so resistance is equivalent to single W

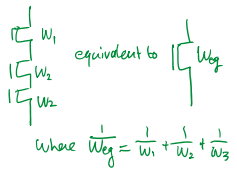
$\uparrow W \Rightarrow \downarrow R$

NOR:



in series, we adjust size to decrease resistance to 2W

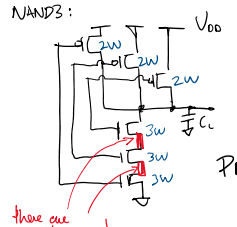
Worst case: one of them is on, so we set the size to match inverter NMOS (size W)



where $\frac{1}{W_{eq}}=\frac{1}{W_1}+\frac{1}{W_2}+\frac{1}{W_3}$

Diffusion Capacitances:

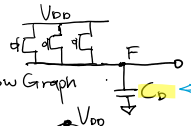
ex. NAND3:



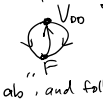
these are capacitances at nodes here

Worst case: C_{eff} PMOS $= C_{eff}(2W+2W)$ ← sharing source & drain
C_{eff} NMOS $= C_{eff}(3W+3W+3W)$

PMOS:



Signal Flow Graph



← we're after C_D

Draw the "slab", and follow SFG, draw gate for each transition

$C_{dpmos} = C_{eff}(2W+2W)$

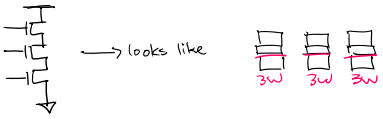
Self capacitances mostly dominated by drain/bulk cap.
Load capacitances mostly dominated by gate cap.

Draw the "slab", and follow SFG, draw gate for each transition

$C_{dpmos} = C_{eff}(2W+2W)$

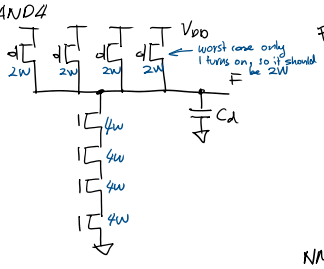
2 diffusion areas
$\therefore (2 \times 2W)$

NMOS:



$C_{dnmos} = C_{eff}(3W+3W+3W)$

ex. NAND4

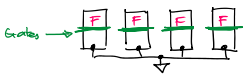


PMOS Diffusion capacitances:

Design:

$\therefore C_{dpmos} = C_{eff} \cdot (2 \times 2W)$

NMOS Diffusion capacitances:



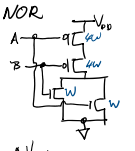
$\therefore C_{dnmos} = C_{eff} \cdot (4 \times 4W)$

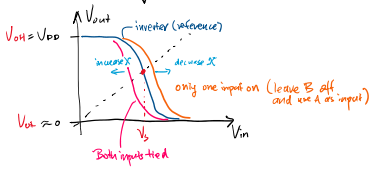
VTC & Noise Margin

- Consider two extreme cases
  - All input ON
  - only one input ON
- All characteristics are bracketed between.

NAND:

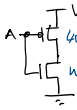
NOR





Only one input test:

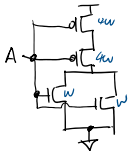
B is off: — equivalent circuit:



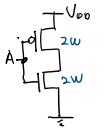
- PMOS much stronger NMOS
- K smaller
- VTC right shift

All input on test:

Both A & B tied equivalent circuit:



≡

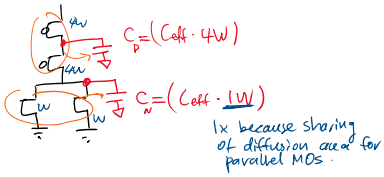


- NMOS much stronger than PMOS
- K bigger
- VTC left shift

We size the MOS to match Resistance,

But what about capacitance? ← Because load capacitances should be much larger, so the extra capacitances shouldn't matter too much.

Total capacitances:

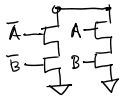


Ex. XOR Gate.

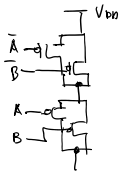
Function of XOR: $f(A,B) = (\bar{A}B + A\bar{B})$ $\quad \bar{f}(A,B) = \bar{A}\bar{B} + AB$

A	B	out
0	0	0
0	1	1
1	0	1
1	1	0

PULL DOWN:

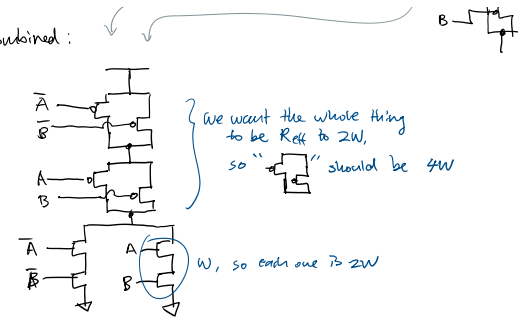


PULL UP (Dual)

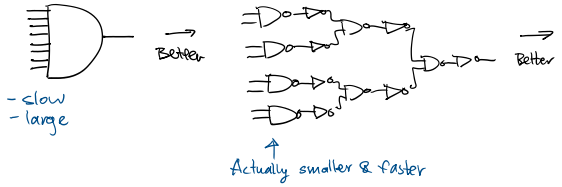


Combined:

Combined:



AND8



Actually smaller & faster

TAKE AWAY: KEEP EACH COMPONENT OF STATIC COMB. LOGIC SHOULD BE LIMITED TO 4 TRANSISTORS

But there's a better way w/ Pseudo-NMOS

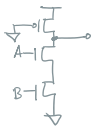
↳ A single PMOS can pull up to V_{DD}

↳ ⊖ but V_{OL} is no longer ≈ 0

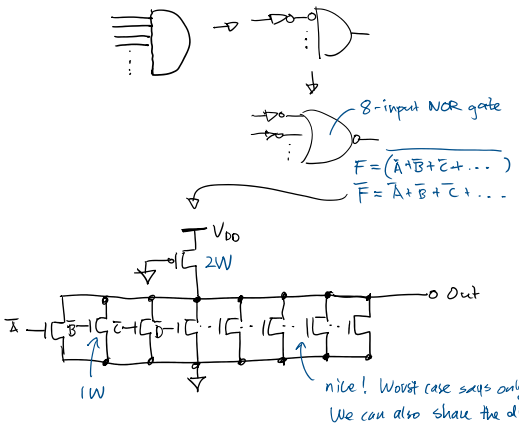
↳ ⊖ t_{pLH} slow

But if we try to make PMOS bigger, then V_{OL} is further from 0.

Pseudo-NMOS NOR:

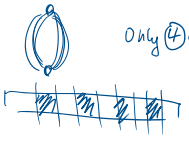


Pseudo NMOS



nice! Worst case says only 1xW is required

We can also share the diffusion areas.



Only (4) diffusion area: $C = C_{eff}(4 \times W)$