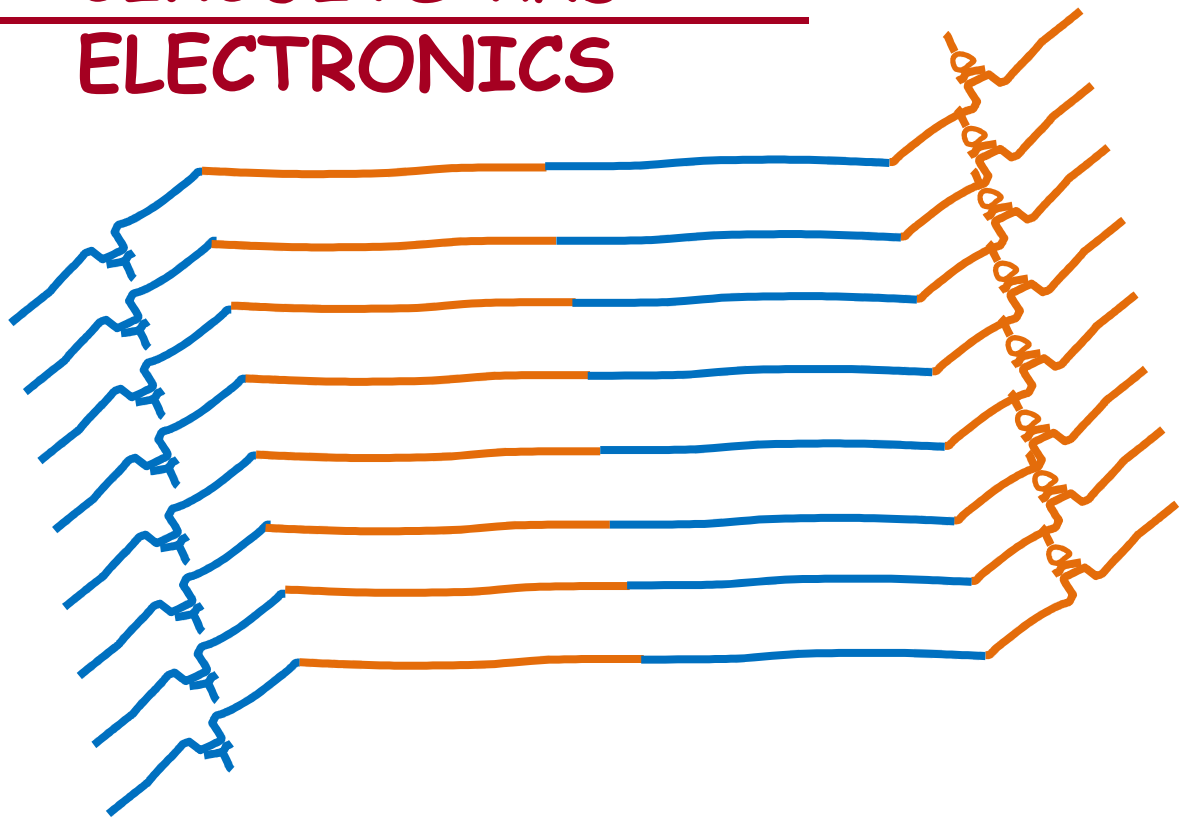
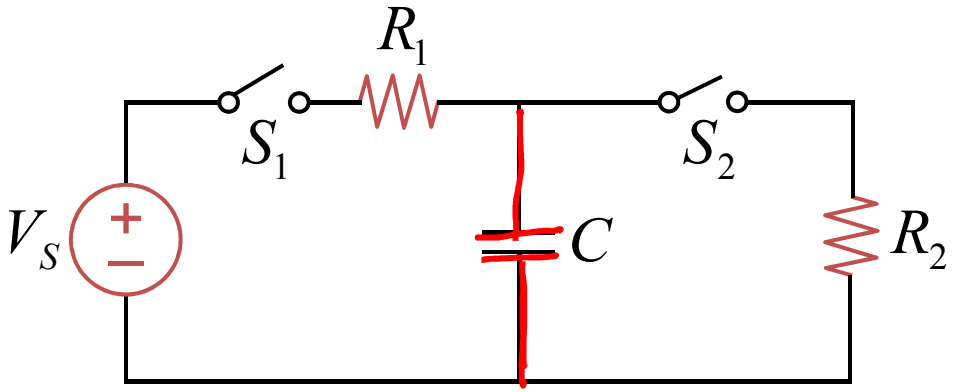
6.002x CIRCUITS AND ELECTRONICS

Energy, CMOS

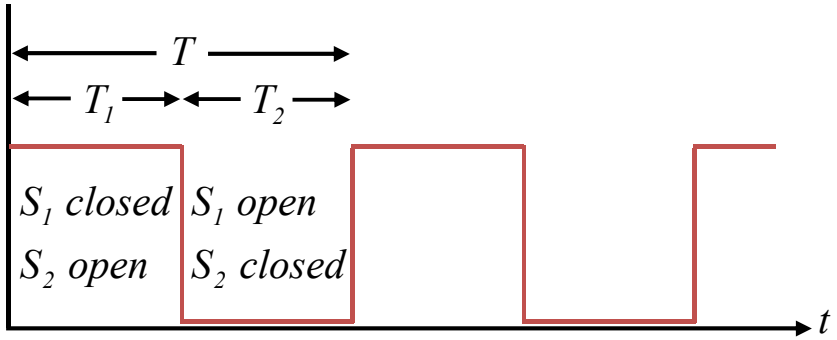


Reading Section 11.4, 11.5 of A&L

Review Hypothetical example circuit

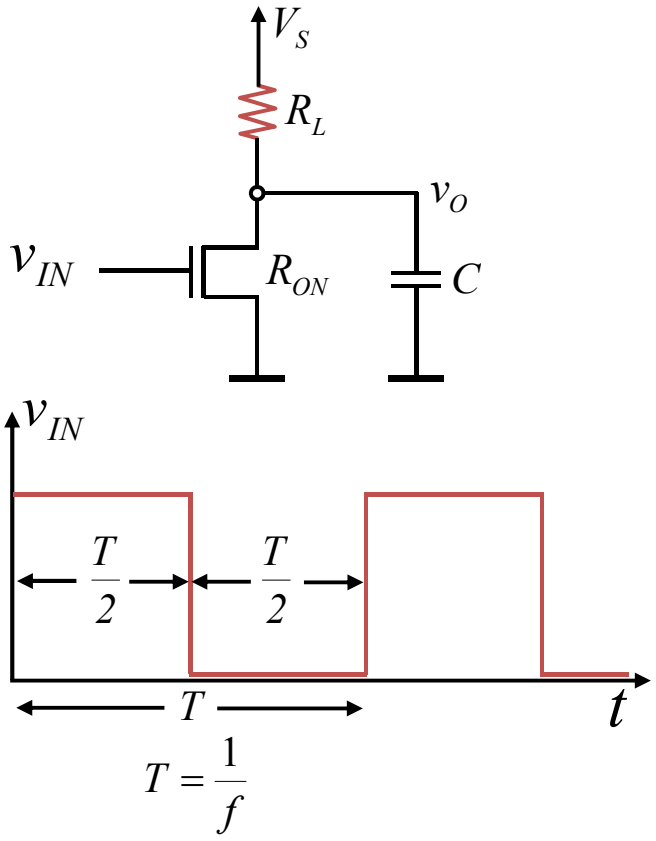


$$\overline{P} = CV_S^{\,2} f$$ ★★

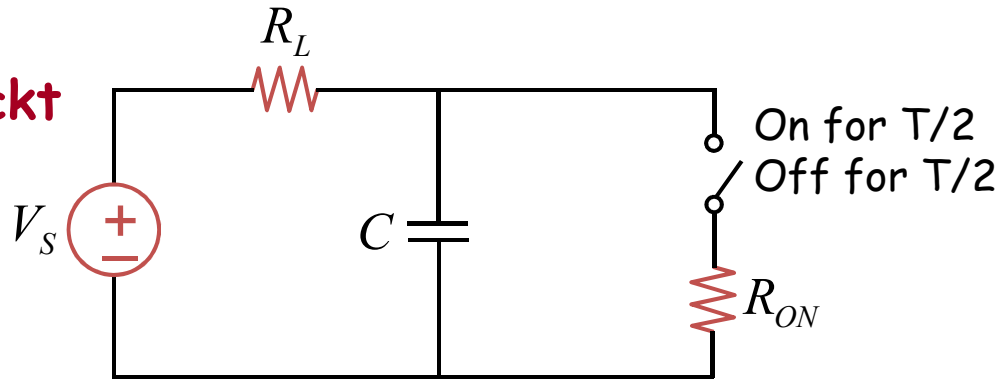


$$T = T_1 + T_2 = \frac{1}{f}$$

Review Inverter



Eqvt. ckt



$$\overline{P} = \frac{V_S^{\,2}}{2R_L} + CV_S^{\,2} f$$

$\overline{P}_{STATIC}$ $\overline{P}_{DYNAMIC}$

Review Inverter – numbers

One gate: $\overline{P} = \frac{V_S^2}{2R_L} + CV_S^2 f$

$C = 1f\,\mathrm{F}$

$R_L = 10\,\mathrm{k}\Omega$

$f = 3\times10^9$

$V_S = 5\,\mathrm{V}$

$\#\,\mathrm{gates} = 5\times10^6$

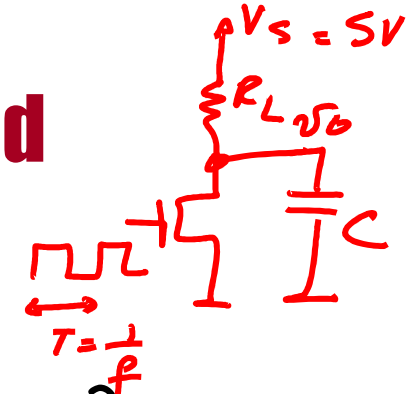
$$\overline{P} = 5\times10^6\left[\frac{25}{2\times10^4} + 10^{-15}\times25\times3\times10^9\right]$$

($\overline{P}_{STATIC}$ and $\overline{P}_{DYNAMIC}$)

$$= 5\times10^6\,[1.25\ \text{milliwatts} + 75\ \text{microwatts}]$$

6.25KW! — **disaster!**

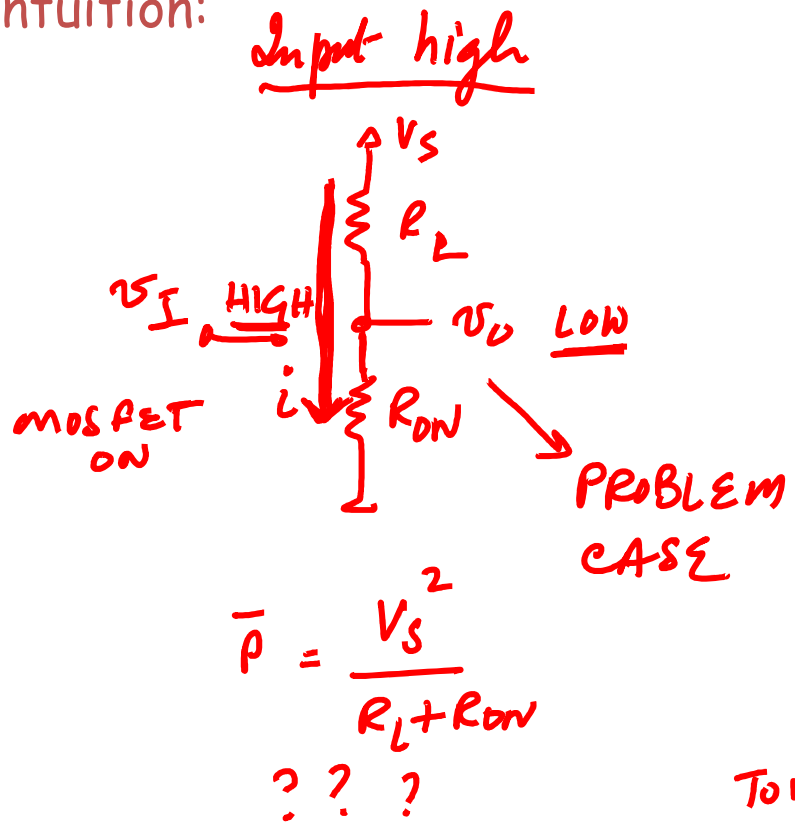
375W — **bad**

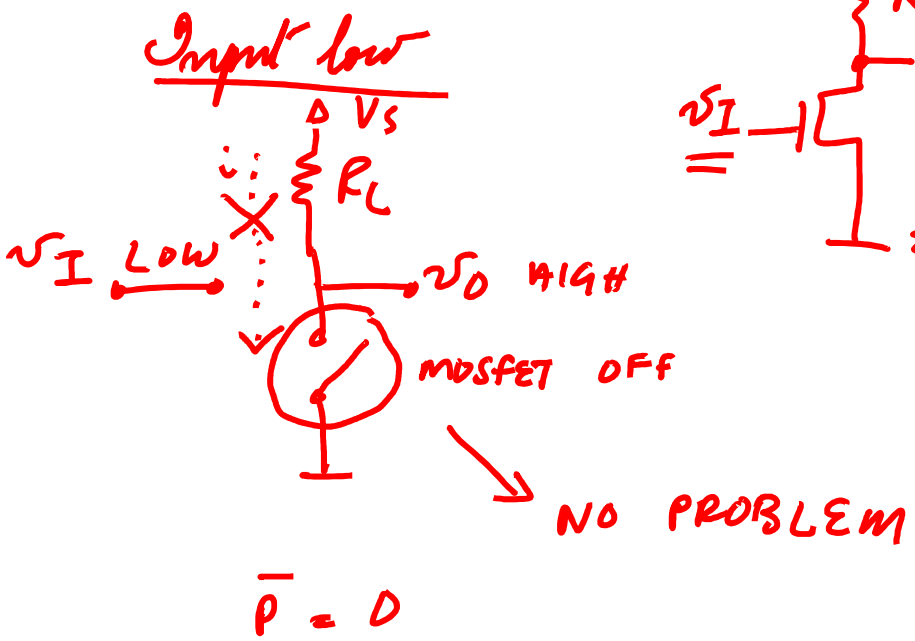


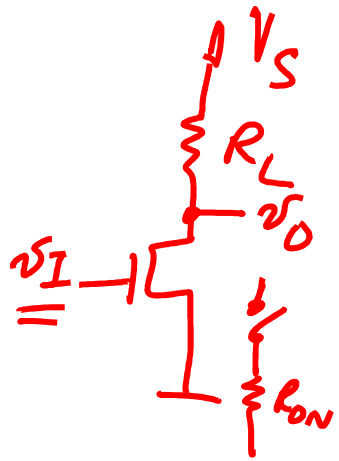
How do we get rid of static power?

How to get rid of static power

Intuition:



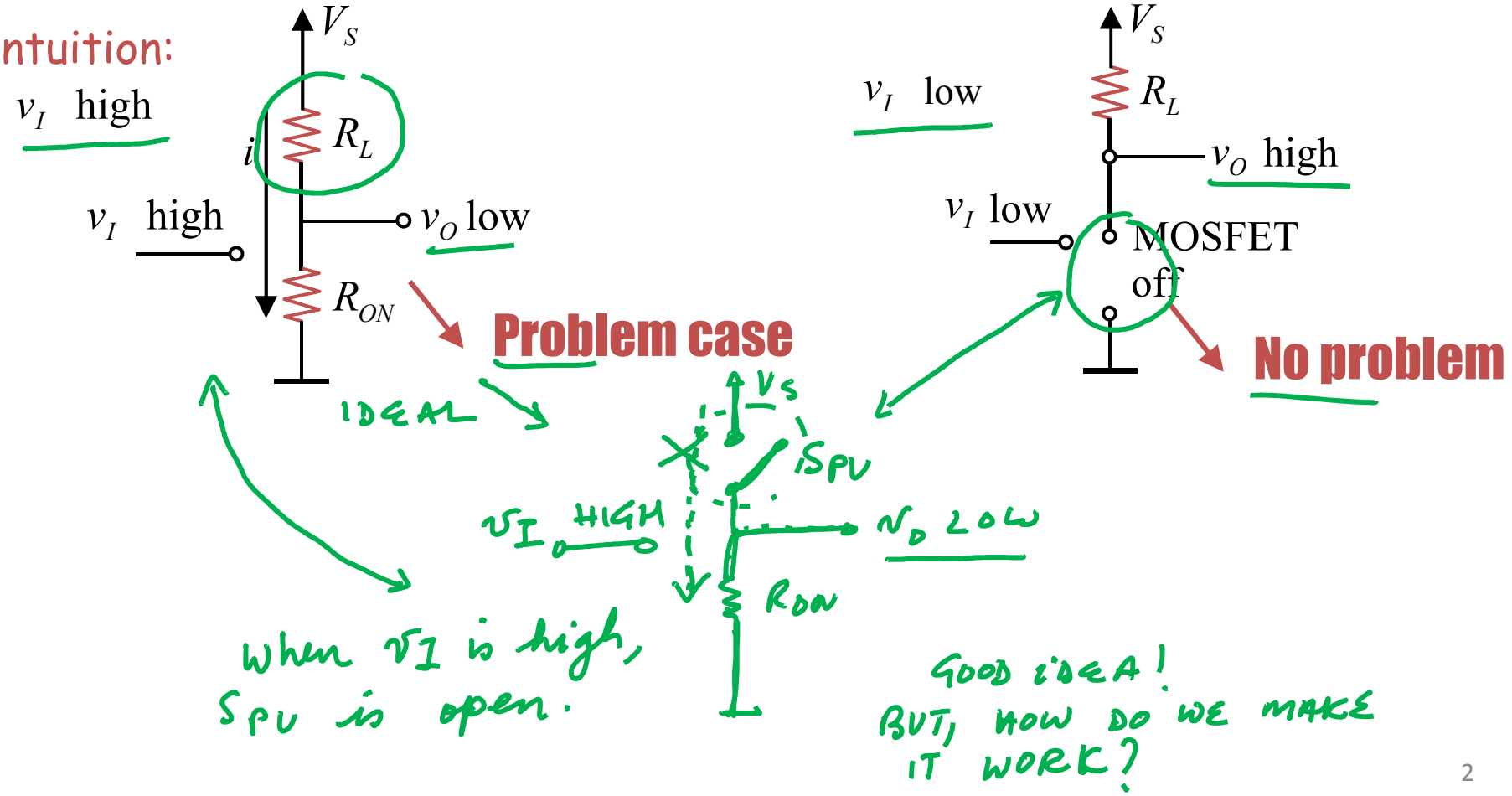




TOWARDS THE NEXT AHA MOMENT!

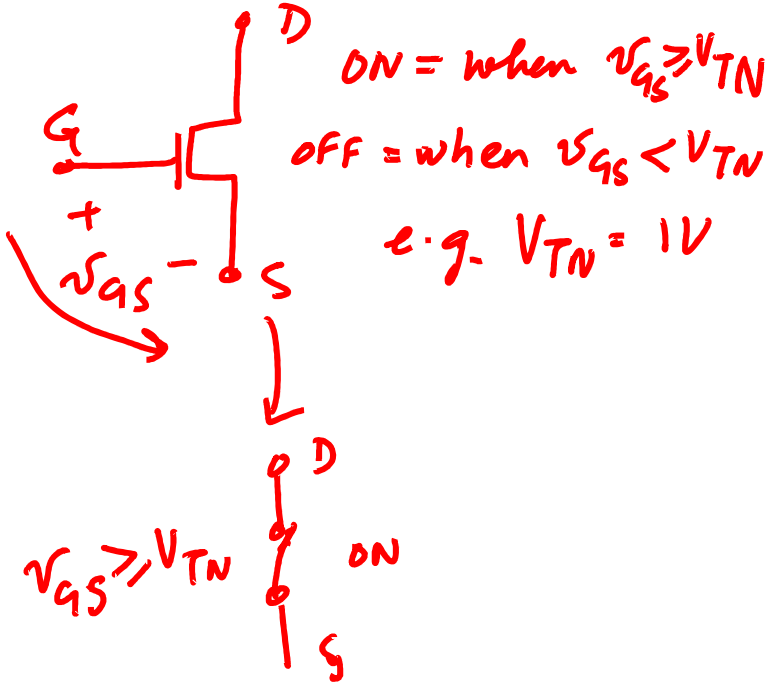
How to get rid of static power when input is high

Intuition:

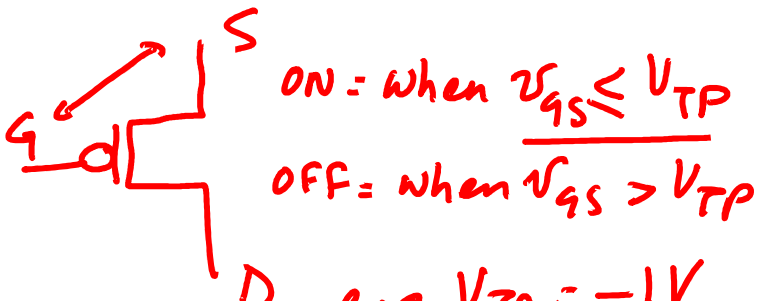


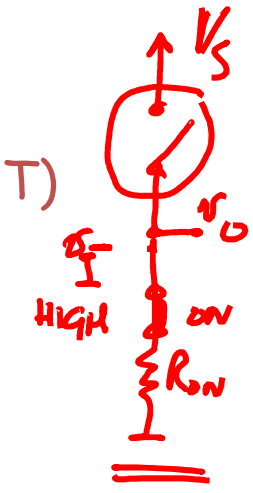
New Device PFET

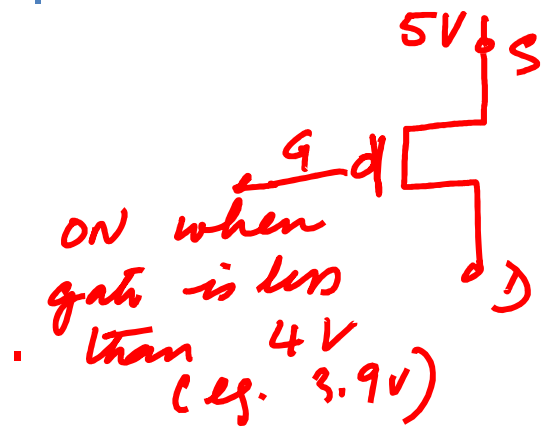
N-channel MOSFET (NFET)



P-channel MOSFET (PFET)







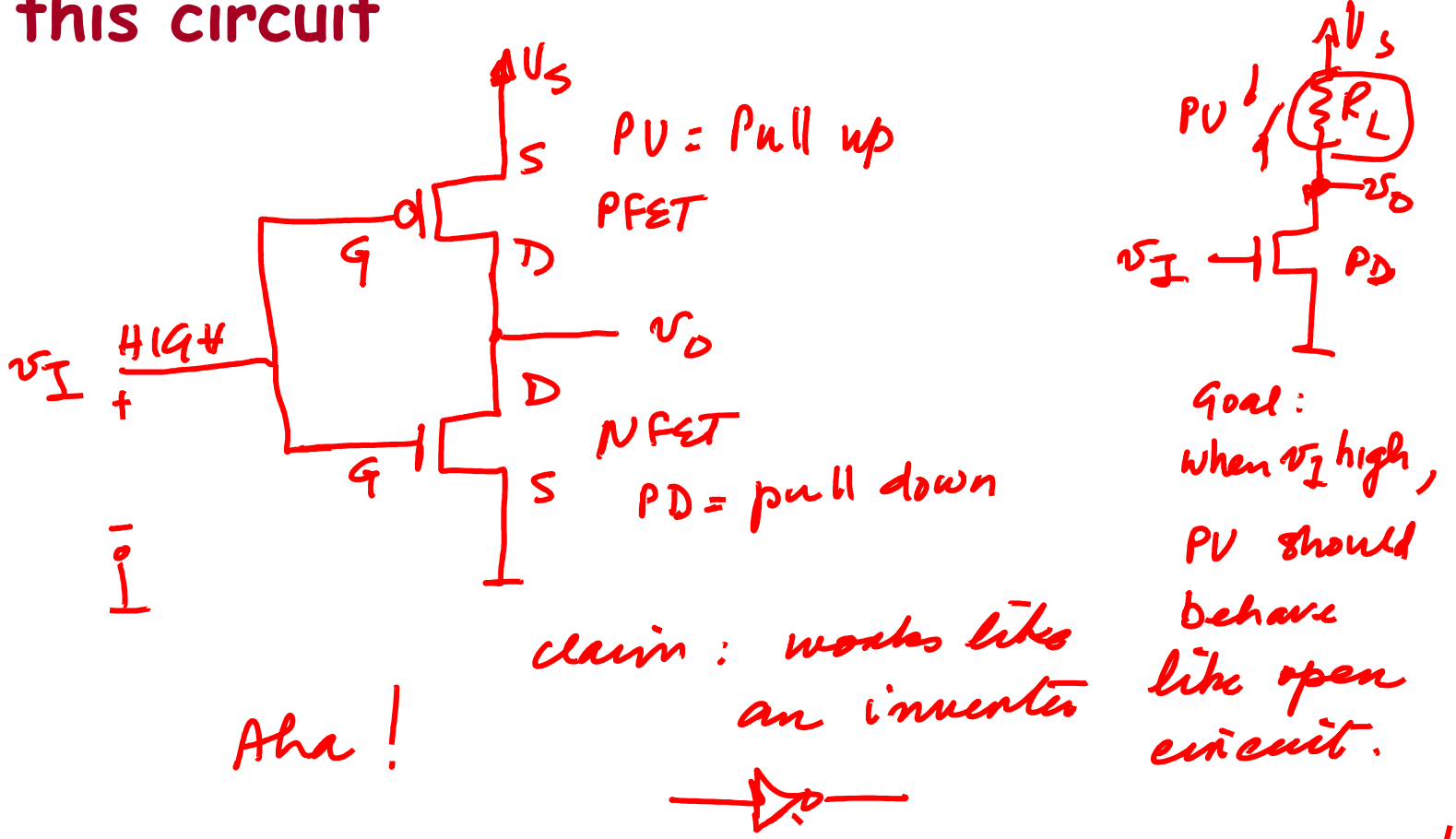
$v_{GS} = -1.1V$

$V_{TP} = -1V$

$v_{GS} < V_{TP}$

$-1.1V \quad -1V$

Consider this circuit



Behavior of the circuit

$v_I = 5V$ (input high)

$V_S = 5V$

$V_S = 5V$

S

PFET

G

D

v_O

$= 0V$

D

R_{ON_N}

S NFET

$v_I = 5V$

$v_I = 0V$ (input low)

V_S

R_{ON_P} PFET

$v_O = 5V$

NFET

v_I	v_O
0V	5V
5V	0V

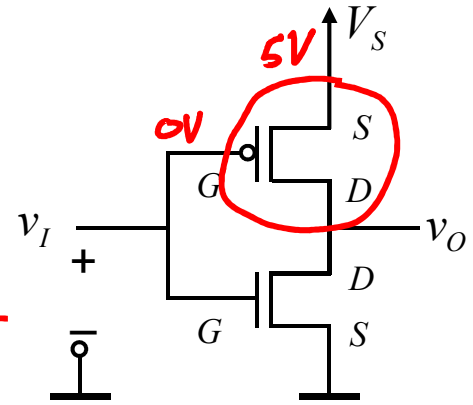
assume $V_{TP} = -1V$

Never has a path from V_S to GND!

Called Complementary MOS, "CMOS" logic

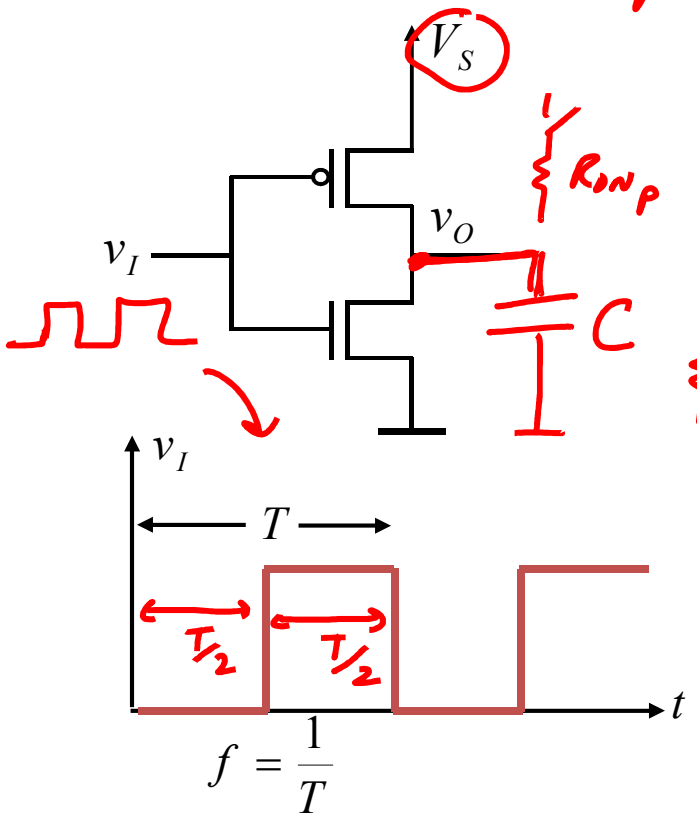
(our previous logic: NMOS)

Behaves like an inverter

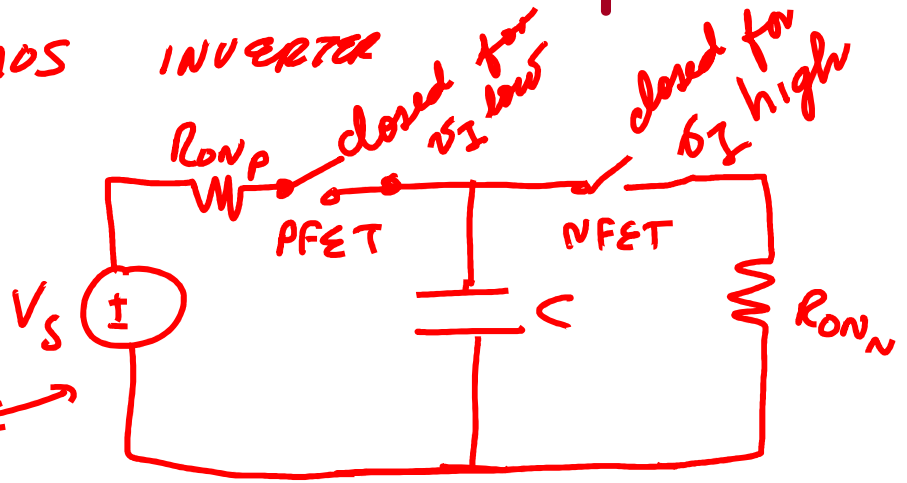


Key: no path from V_S to *GND*! no static power!

Let's compute $\overline{P}_{DYNAMIC}$



for CMOS INVERTER



WE SAW THIS BEFORE! IN REVIEW!

FROM $\overline{P} = C V_S^2 f$

Using numbers from our previous example for CMOS

$$\overline{P} = CV_S^2 f$$

$C = 1f\,\mathrm{F}$ (C = 1 pF)

$R_L = 10\,\mathrm{k}\Omega$

$f = 3\times10^9$ (f = 3 × 10⁹ Hz)

$V_S = 5\,\mathrm{V}$ (V_S = 5 V)

$\#\,\mathrm{gates} = 5\times10^6$ (# gates = 5 × 10⁶)

for 1 gate: $\overline{P} = CV_S^2 f$ — No, static power!

$= 10^{-15} \times 5^2 \times 3\times10^9$

$= 75\,\mu\mathrm{W}$ per gate

For 5×10^6 gate chip

$\overline{P} = 5\times10^6 \times 75\times10^{-6}\,\mathrm{W}$

$= 375\,\mathrm{W}$ → dynamic power

Not as bad as the 6.25 kW.

But 375 W is still bad and worsens in the future

Scaling up

Increase frequency and number of gates, but if everything else stayed the same

$$\overline{P} = CV_S^2 f$$

$C = 1f\,\mathrm{F}$

~~$R_L = 10\mathrm{k}\Omega$~~

~~$f = 3\times10^9$~~

$V_S = 5\,\mathrm{V}$

Gates	f	$\overline{P}$
10^6	100 MHz	~2.5 watts
2×10^6	300 MHz	~15 watts
2×10^6	600 MHz	~30 watts
5×10^6	3 GHz	~375 watts
25×10^6	3 GHz	~1875 watts

approx. same # of gates as this processor

Pentium

Pentium II

change

change ~~# gates~~

Pentium IV

MIT RAW processor

How to reduce power

BIG RESEARCH AREA!

$$\overline{P} = CV_S^2 f$$

$C = 1fF$

$R_L = 10k\Omega$

$f = 3\times10^9$

$V_S = 5V$

- V_S 5V → 3V → 1.8V → 1.5V → 1V → 0.7V

  1875W → 75W @ 1V (cannot run ckt as fast through)

- C shrinks as the process shrinks (transistors get smaller)

  700nm technology (~ transistor gate length) in 1993

  32nm in 2012!

  100M → Gates → 1B

  $i = \frac{k}{2}(V_{GS} - V_T)^2$

- Turn off clock when sub-circuit not in use — clock gating

- Change V_S depending on need.

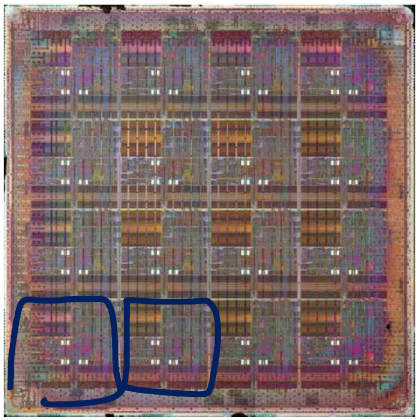
  0.7V and 1.5GHz most of the time.

  But increase to 1V and 2GHz when we want to boost performance

Real numbers

$$\overline{P} = CV_S^2 f$$

Transistors	f	$\overline{P}$	
2.5x10⁶	66 MHz	15W	Intel Pentium 700nm, 1993
7.5x10⁶	400 MHz	35W	Intel PII 350nm, 1997
44x10⁶	1 GHz	15W	Intel PIII 180nm, 1999
120x10⁶	0.43 GHz	18W	MIT Raw processor 16-core 180nm, 2002
120x10⁶	3 GHz	75W	Intel PIV 130nm, 2001
615x10⁶	0.7 GHz	20W	Tilera Tile64Pro 64-core 90nm, 2007
2300x10⁶	3.6 GHz	75W	Intel Nehalem 8-core 45nm, 2011



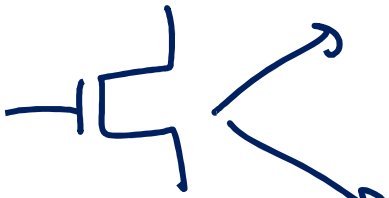
http://groups.csail.mit.edu/cag/raw

Scaling up – leakage arises, causes static power dissipation

Leakage is a new form of static power

↳ temperature

ON : R_{ON} very low



OFF : ideally ∞

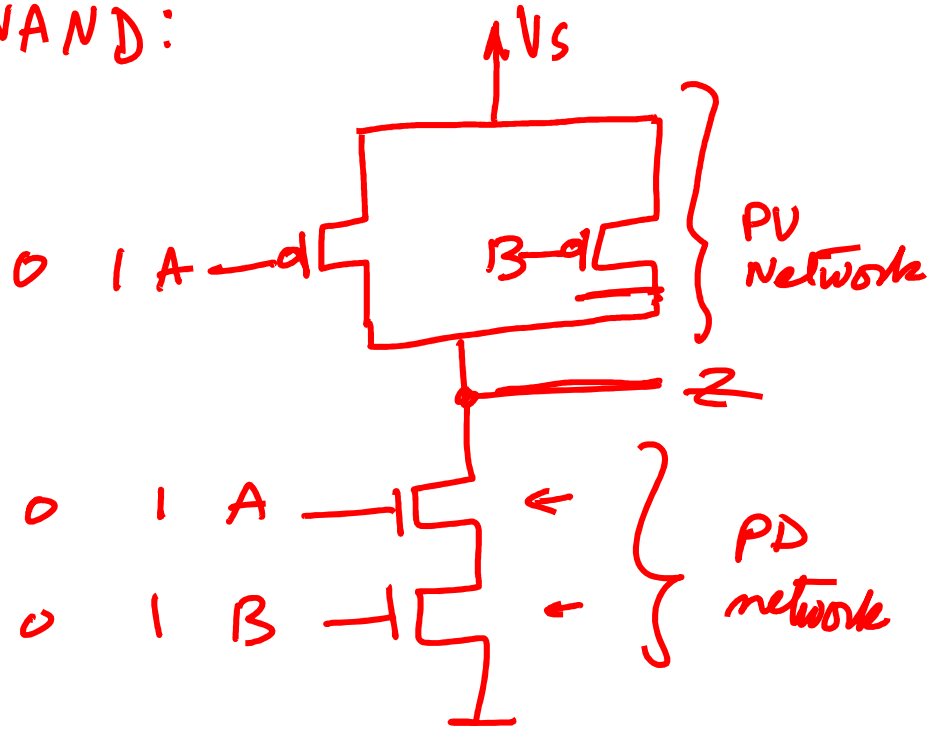
Practically, R_{OFF}

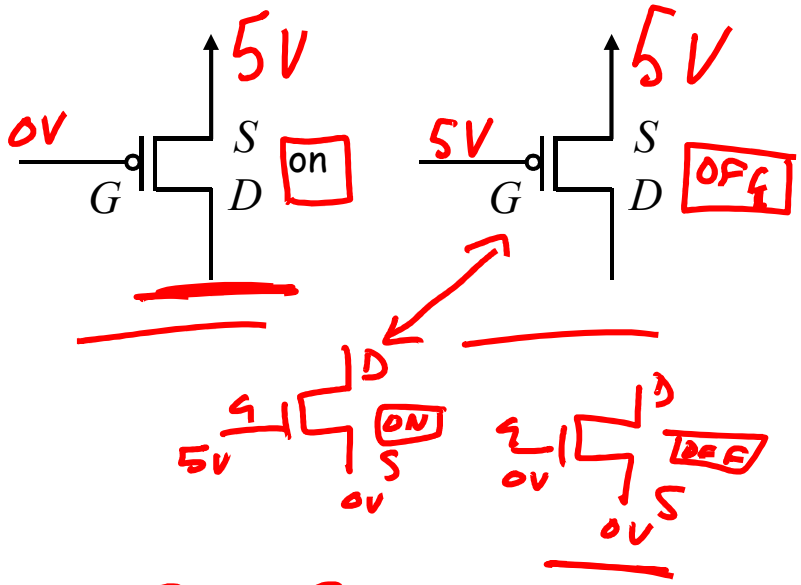
High !

But not open circuit

CMOS Logic

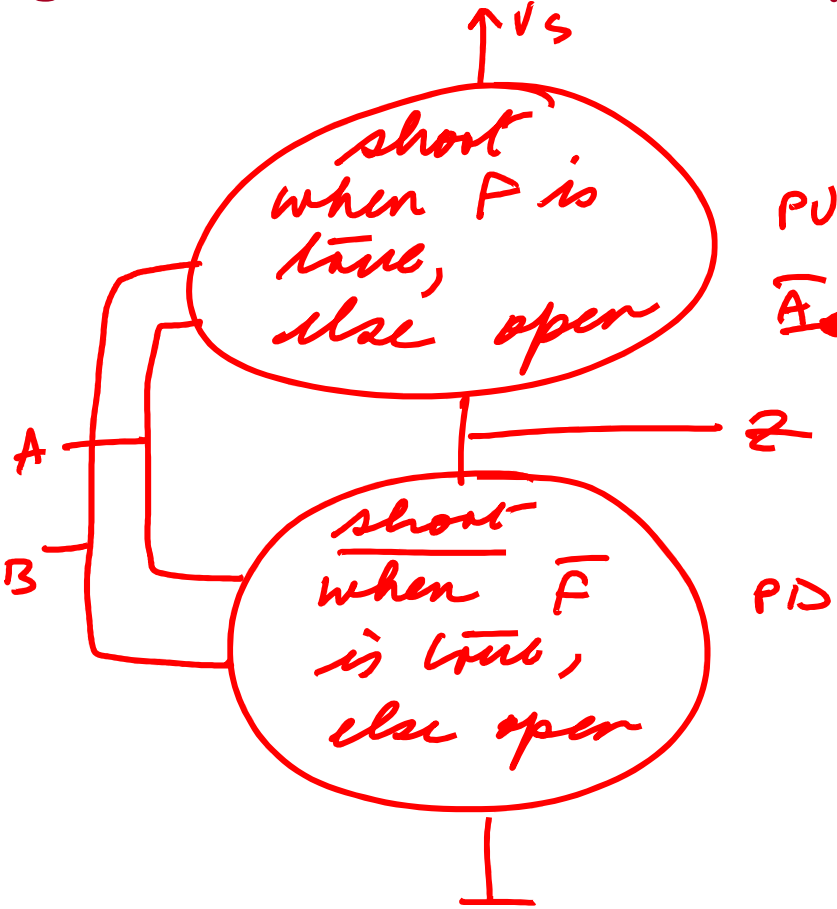
NAND:





A	B	Z
0	0	1
0	1	1
1	0	1
1	1	0

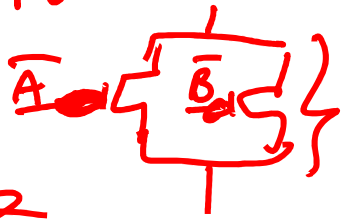
In general, if we want to implement F



e.g. $F = \overline{A \cdot B}$ or $\bar{F} = A \cdot B$

inversion inherent in pFets

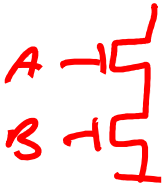
PU



use: $F = \overline{A \cdot B} = \bar{A} + \bar{B}$

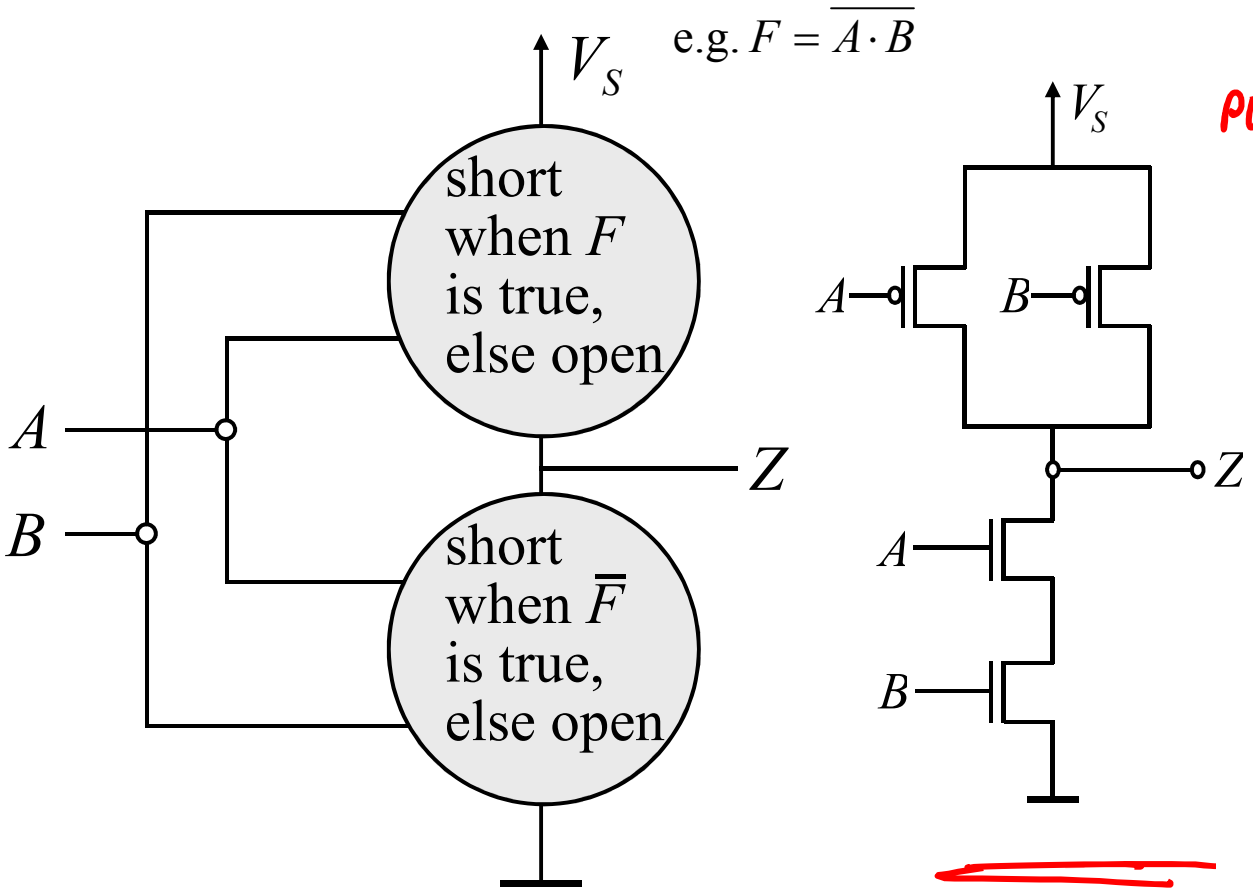
De Morgan's law

on page 263 of AFL



short when $A \cdot B$ is true, else open

If we want to implement F



use $F = \overline{A \cdot B} = \bar{A} + \bar{B}$

PU: short $\overline{A \cdot B} = \bar{A} + \bar{B}$ is true. note the OR style connection.

PD: Short when $A \cdot B$ is true. note the AND connection.