Chapter 10

Analog details

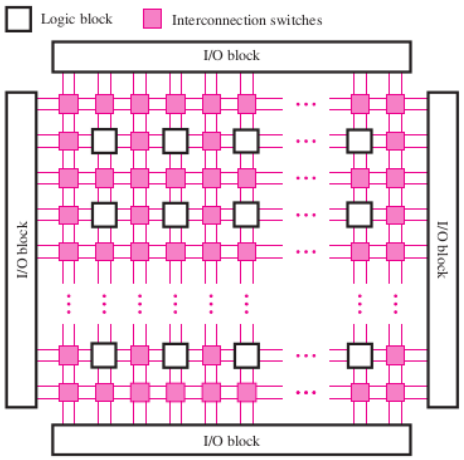
Some of the material is out of the textbook. Additional resources include Appendix B of Brown and Vranesic book, "Fundamentals of digital logic."

10.1 Objectives

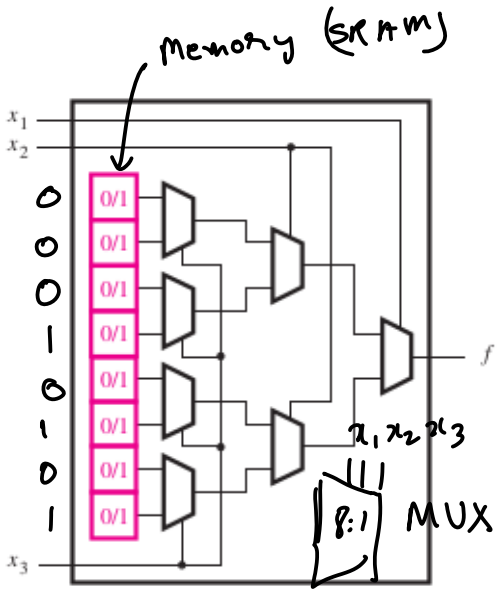
1. Describe how tri-state and open-collector outputs are different from totem-pole outputs
2. Compute noise margin of one device driving the same time

10.2 FPGA [3, Section B.6.5]

Verilog → ASIC



(a) General structure of an FPGA



x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Figure B.37 A three-input LUT ← Look up table

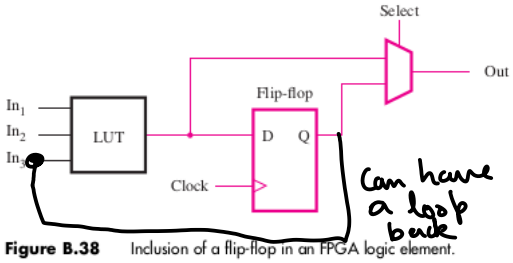


Figure B.38 Inclusion of a flip-flop in an FPGA logic element.

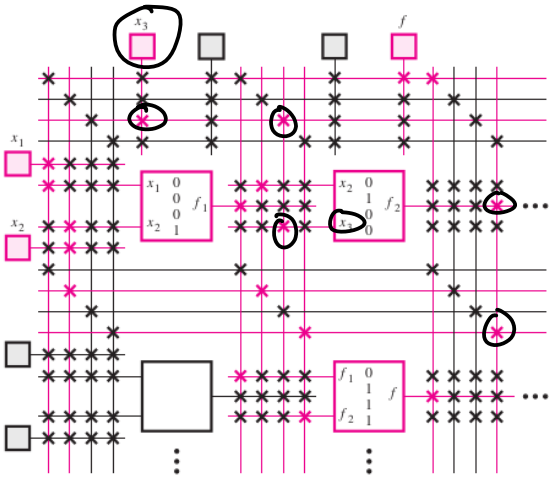


Figure B.39 A section of a programmed FPGA.

Definition 10.1 (Random Access Memory (RAM)). *Structure of a RAM is as follows:*

How can we arrange single bit memory cells into an addressable memory?

How many location can be encoded in a n-bit address bus?

2^n

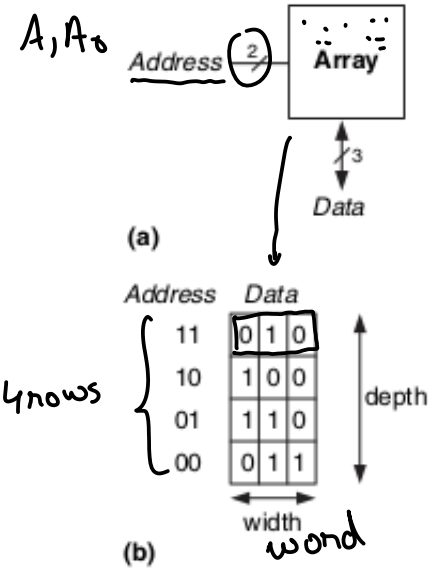


Figure 5.39 4 × 3 memory array: (a) symbol, (b) function

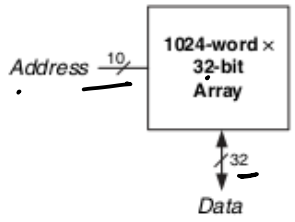


Figure 5.40 32 Kb array: depth = 2^{10} = 1024 words, width = 32 bits

$2^{10} = 1024 \simeq 10^3$

A_1	A_0
0	0
0	1
1	0
1	1

nmos switches

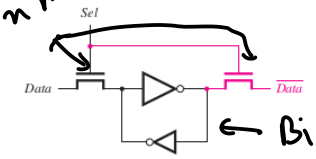


Figure B.64 An SRAM cell.

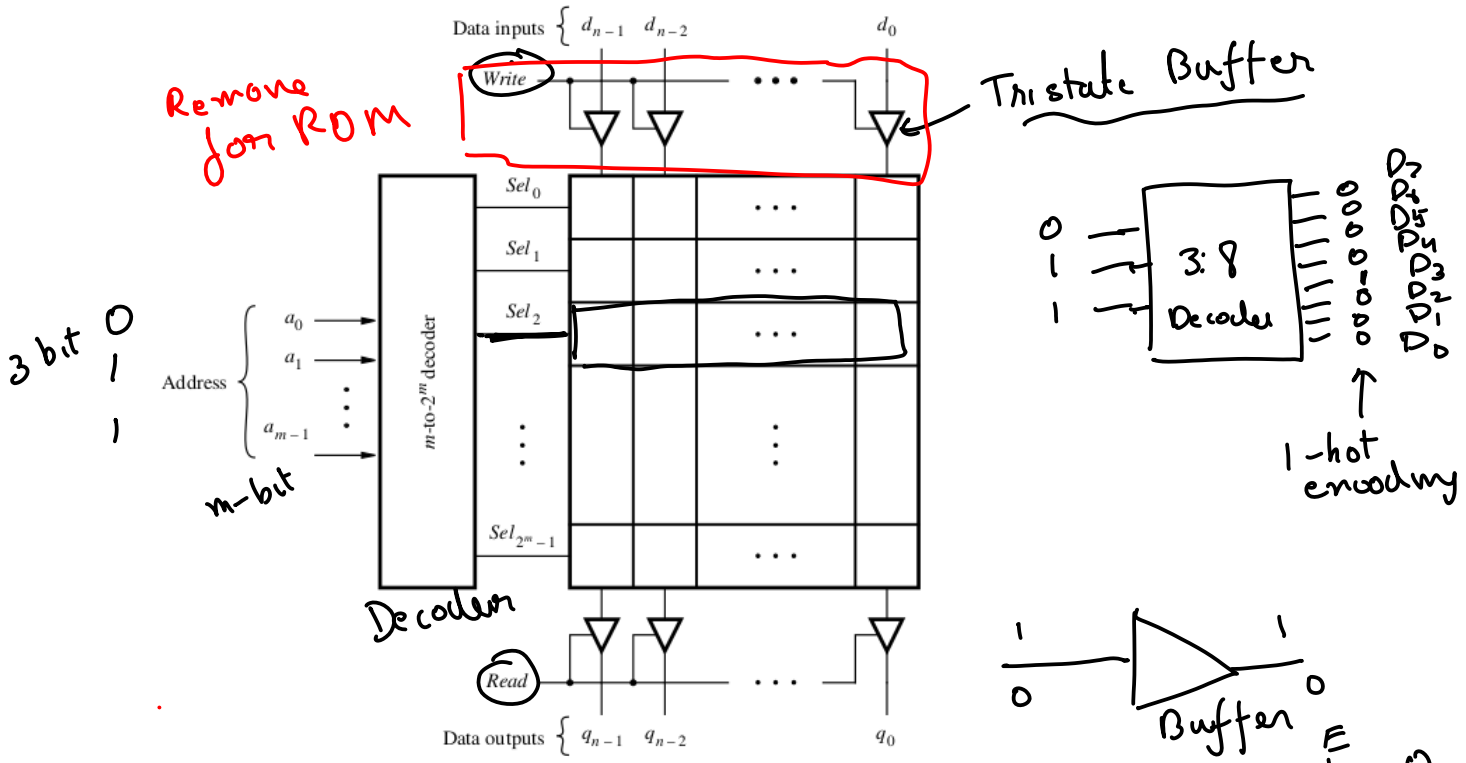
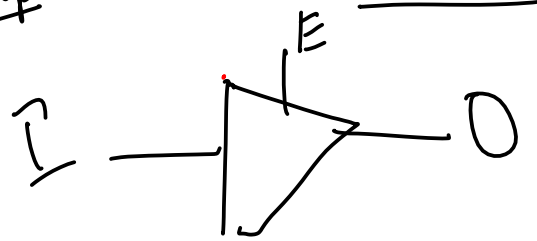


Figure B.66 A $2^m \times n$ SRAM block.

Skip

Definition 10.2 (Read Only Memory (ROM)). *Structure of a ROM is as follows:*



E	O
0	floating = Z
1	O = I

E	O
1	I
0	Z

Tristate floating

0/1/Z/X ← floating

↑ undefined

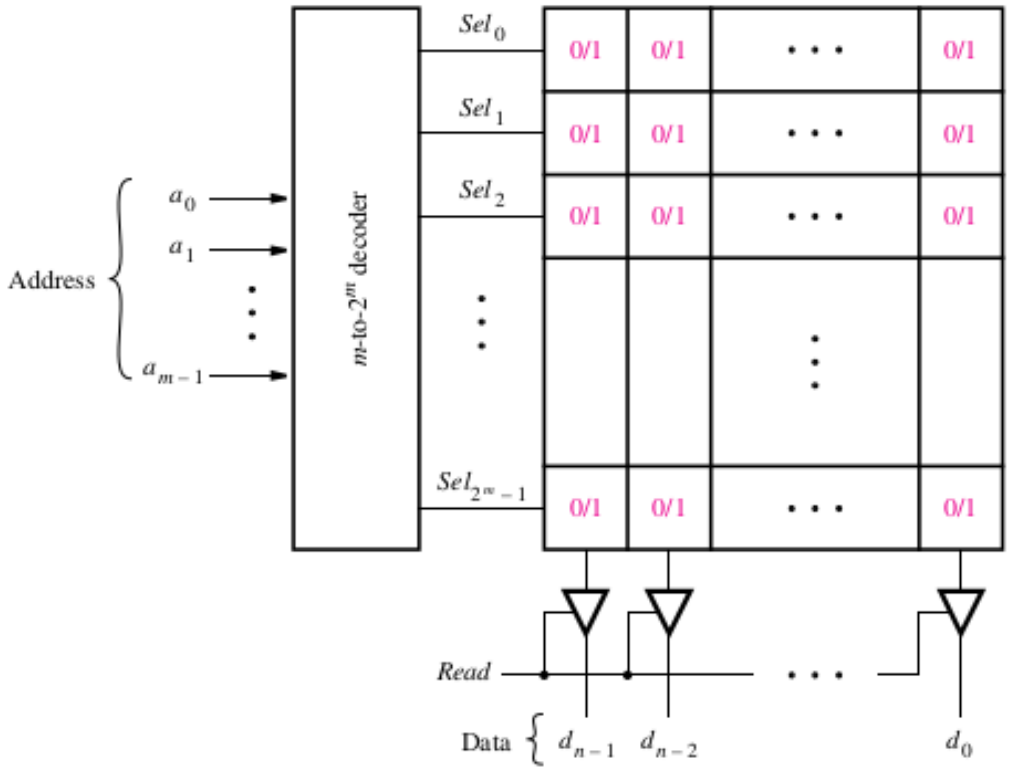
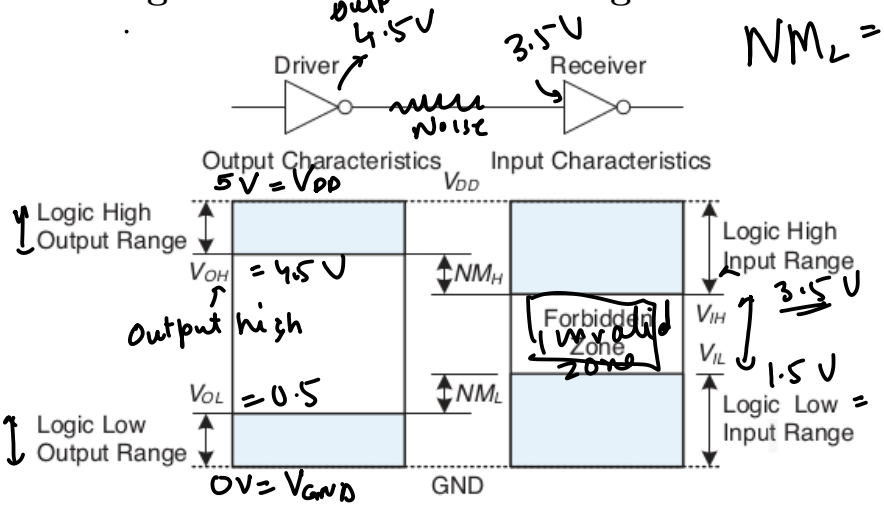


Figure B.72 A $2^m \times n$ read-only memory (ROM) block.

Example 10.1. *Draw a Multiplexer using sum of products form.* Already covered

10.3 Logic levels and Noise Margins

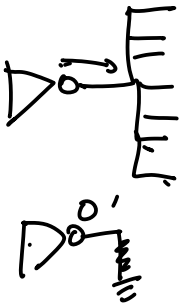


Definition 10.3 (Supply Voltage ($V_{DD}/V_{CC}/V_{DD}$)). *The highest DC voltage that drives a digital circuit. As chips have progressed to smaller transistors, V_{DD} has dropped from 5V to 1.2V or even lower to save power.*

Definition 10.4 (Ground Voltage (V_{GND})). *The lowest DC voltage that drives a digital circuit, typically 0V.*

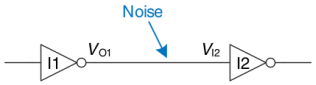
Definition 10.5 (Input high (V_{IH}) and Input Low (V_{IL}) of a gate). *V_{IH} is the voltage level, such that an input voltage to a gate between V_{DD} and V_{IH} is considered* HIGH. *Similarly, input voltage to a gate between V_{IL} and V_{GND} is considered* LOW.

Definition 10.6 (Output high (V_{OH}) and Output low (V_{OL}) of gate). *V_{OH} is the voltage level, such that an output voltage to a gate between V_{DD} and V_{OH} is considered* HIGH. *Similarly, output voltage to a gate between V_{OL} and V_{GND} is considered* LOW.



Definition 10.7 (Positive logic and Negative logic). *What we have considered so far is Positive logic where* HIGH *voltage is equated to Boolean logic* TRUE *or* 1 *and* LOW *is considered* FALSE *or* 0. *In negative logic these are reversed. Same physical circuit can represent different logical circuits in positive logic and negative logic.*

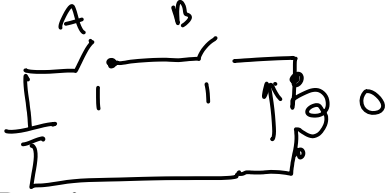
Definition 10.8 (Noise margins (NM_L and NM_H) of a channel). *The maximum amount of noise that can be added (or substracted) to a channel without exceeding the logic level specifications of a gate.* $NM_L = V_{IL} - V_{OL}$
$NM_H = V_{OH} - V_{IH}$



Example 10.2.
If $V_{DD} = 5V$, $V_{IL} = 1.35V$, $V_{IH} = 3.15V$, $V_{OL} = 0.33V$ and $V_{OH} = 3.84V$ for both the "inverters", then what are the low and high noise margins? Can the circuit tolerate 1V of noise at the channel?

$NM_L = 1.02 = 1.35 - 0.33 = V_{IL} - V_{OL}$

$NM_H = 0.69 = 3.84 - 3.15 = V_{OH} - V_{IH}$



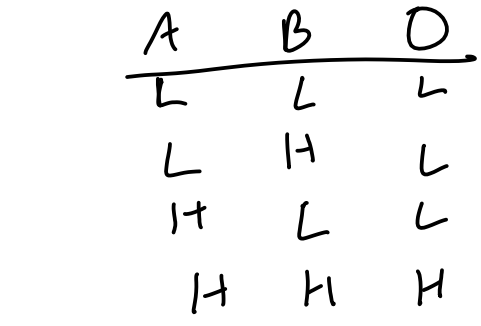
10.4 Semiconductors and Doping

Not in syllabus but good to know

Elements recognized as metalloids V • T • E

	13	14	15	16	17
2	B Boron	C Carbon	N Nitrogen	O Oxygen	F Fluorine
3	Al Aluminium	Si Silicon	P Phosphorus	S Sulfur	Cl Chlorine
4	Ga Gallium	Ge Germanium	As Arsenic	Se Selenium	Br Bromine
5	In Indium	Sn Tin	Sb Antimony	Te Tellurium	I Iodine
6	Tl Thallium	Pb Lead	Bi Bismuth	Po Polonium	At Astatine

Commonly recognized (86–99%): B, Si, Ge, As, Sb, Te
Irregularly recognized (40–49%): Po, At
Less commonly recognized (24%): Se
Rarely recognized (8–10%): C, Al
(All other elements cited in less than 6% of sources)
Arbitrary metal-nonmetal dividing line: between Be and B, Al and Si, Ge and As, Sb and Te, Po and At



L=0 H=1 Logic +ve

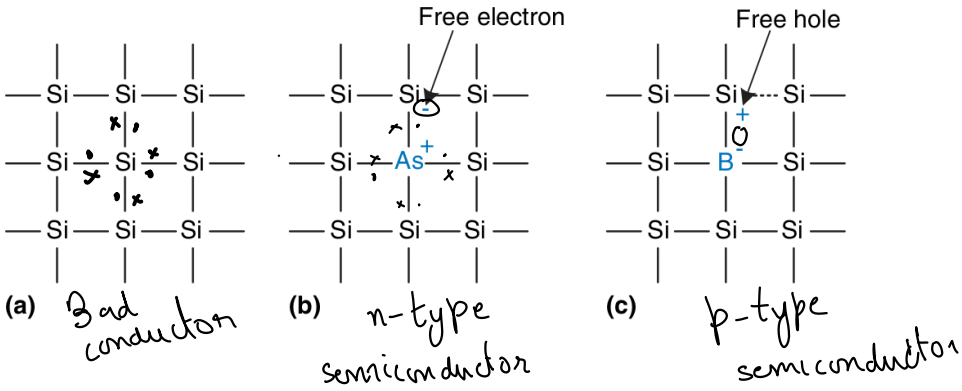
-ve logic L=1 H=0

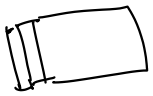
AND gate

A	B	D
0	0	0
0	1	0
1	0	0
1	1	1

A	B	D
1	1	1
1	0	1
0	1	1
0	0	0

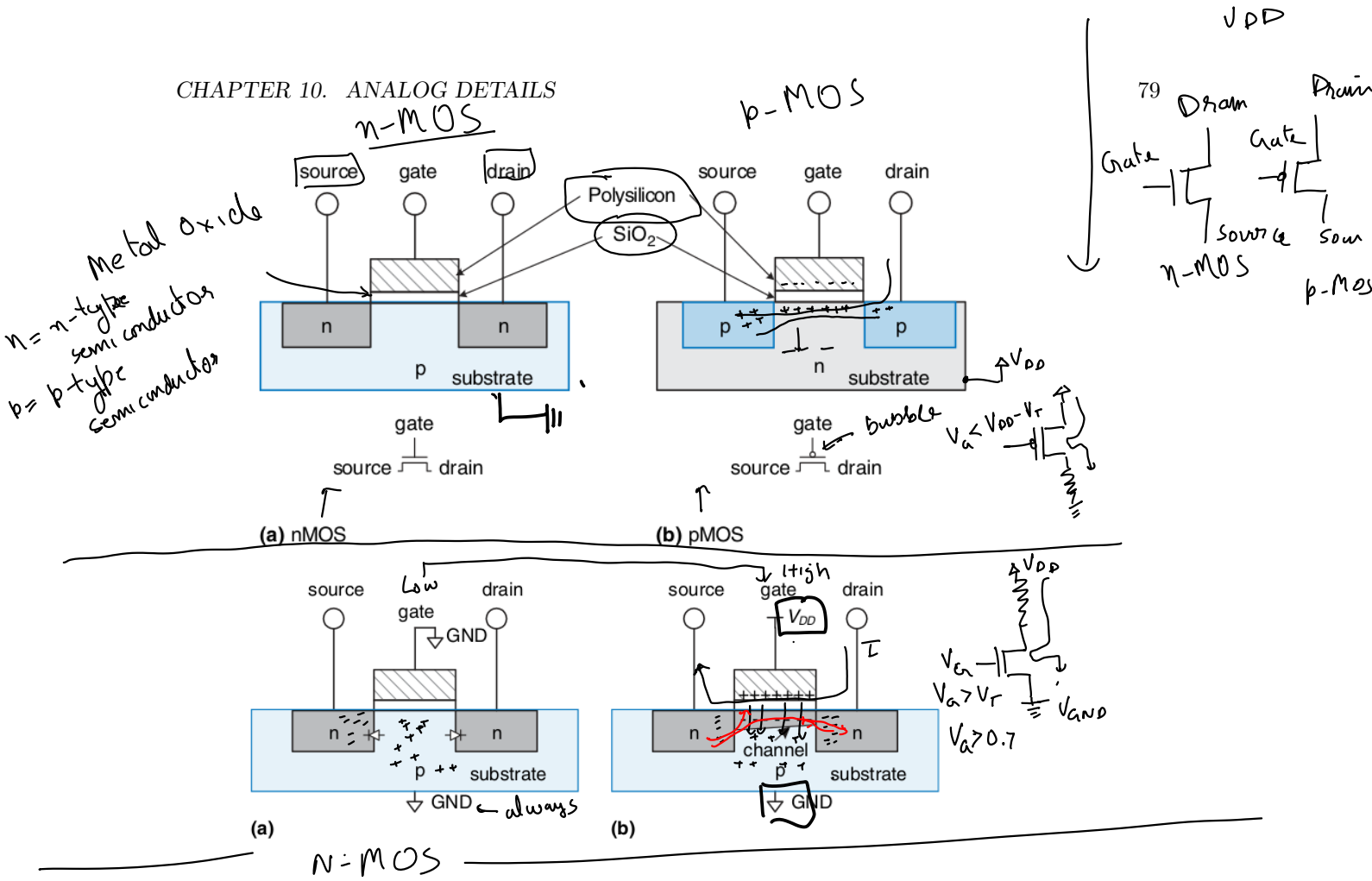
OR gate



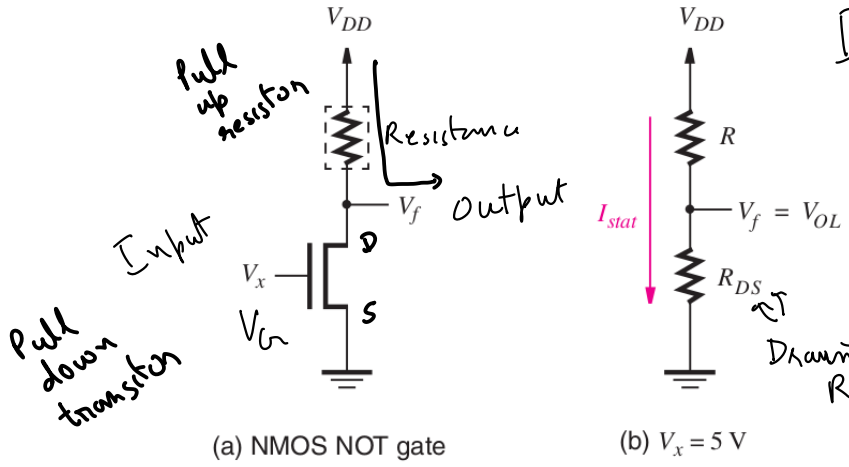


10.5 MOSFET: Metal Oxide Field Effect Transistors

Not in syllabus but good to know



10.6 DC Transfer characteristic

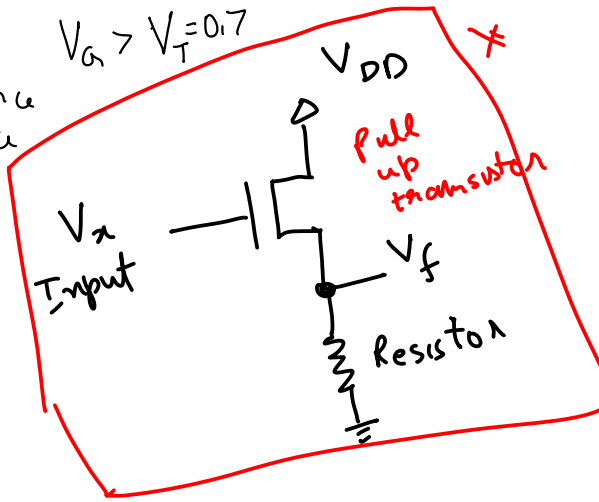


$$I_{stat} = \frac{V_{DD}}{R + R_{DS}}$$

$$V_f = \left(\frac{R_{DS}}{R + R_{DS}}\right) V_{DD}$$

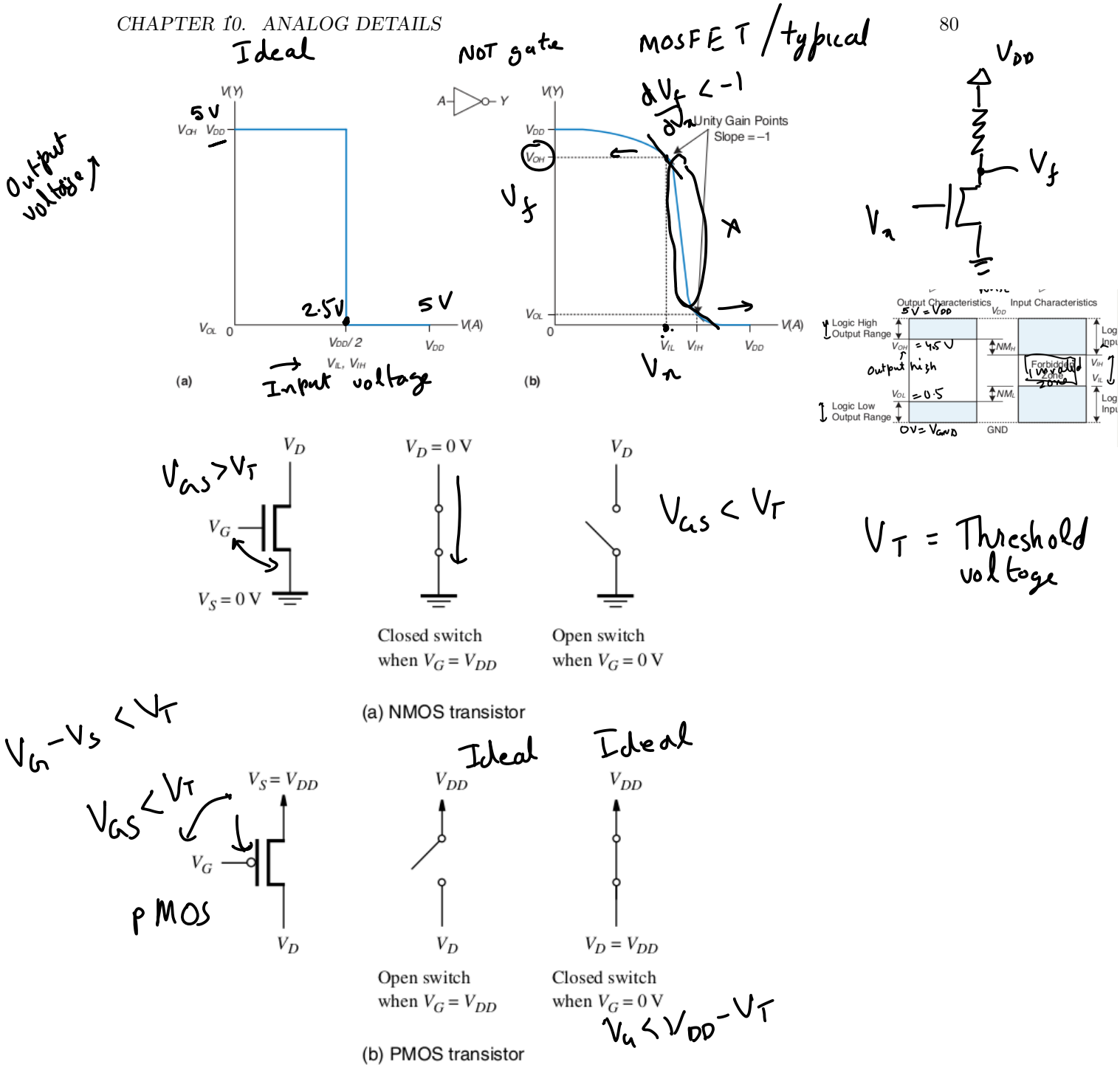
$V_G > V_T = 0.7$

Drain to Source Resistance

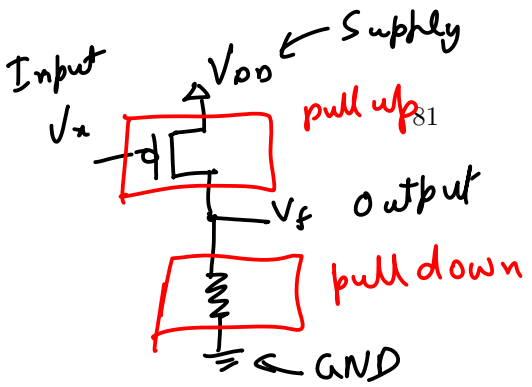


n-MOS should not be used as a pull up transistor only used it as a pull down transistor

Vice-versa for p-MOS. p-MOS: pull up transistor. Why?



Example 10.3. *Draw a NOT gate using nMOS transistors.*



Example 10.4. *Draw a NOT gate using pMOS transistors.*

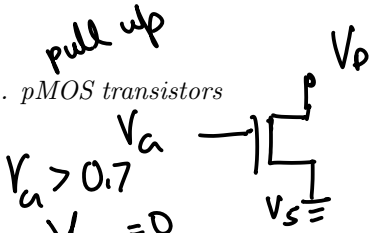
V_x	V_f
L	H
H	L

pull down

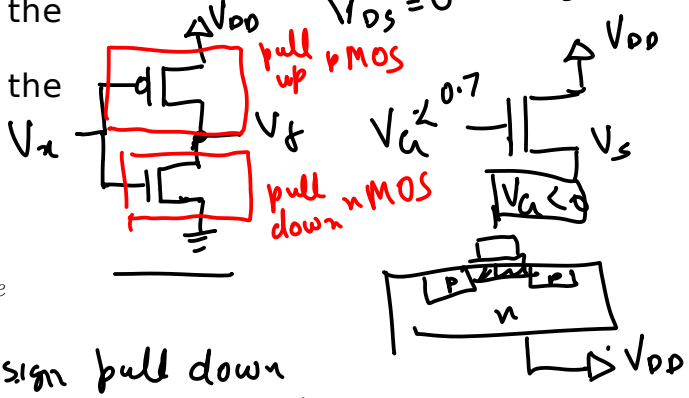
Remark 10.1. *nMOS transistors pass 0's well (output between 0 and $V_{DD} - V_t$). pMOS transistors pass 1's well (output between V_t and V_{DD}).*

pull up

Example 10.5. *Draw CMOS NOT Gate.* complimentary MOS

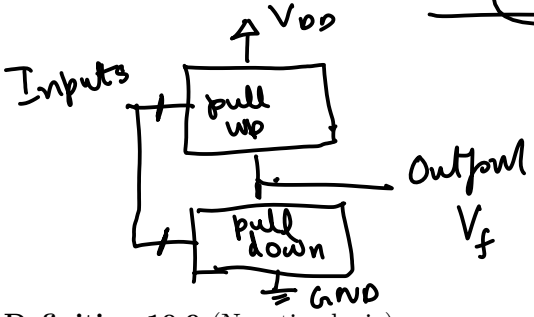


n-MOS behaves well when gate voltage is opposite to the substrate voltage (V_GND).
p-MOS behaves well when gate voltage is opposite to the substrate voltage (V_DD/supply)

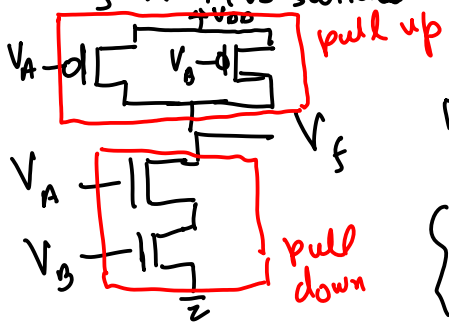


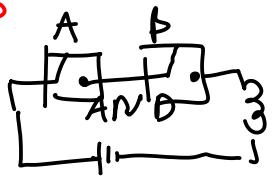
Example 10.6. *Draw a two input CMOS NAND Gate*

A	B	f
0	0	1
0	1	1
1	0	1
1	1	0



① Design pull down using n-MOS switches





{ AND ≡ Series
OR ≡ parallel

Definition 10.9 (Negative logic)**.**

$f = \overline{A \cdot B} = \overline{A} + \overline{B}$

② Design pull up side as DeMorgan inverse of pull down side

Example 10.7. *Analyze the above circuit under negative logic.*

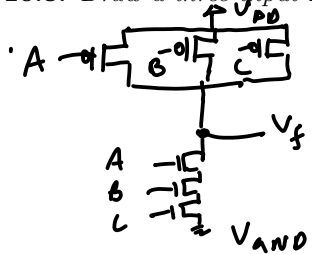
V_A	V_B	V_f
L	L	H
L	H	H
H	L	H
H	H	L

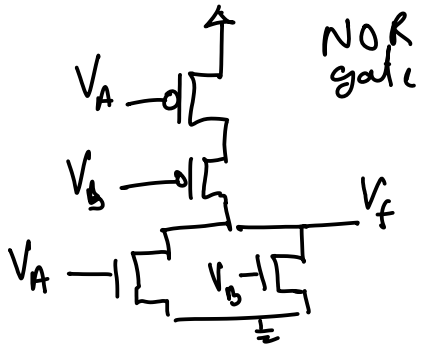
−ve logic
L = 1
H = 0

A	B	f
1	1	0
1	0	0
0	1	0
0	0	1

NOR gate $f = \overline{A + B}$

Example 10.8. *Draw a three input NAND using CMOS.*



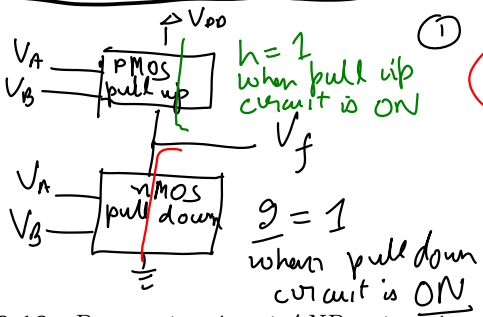


two input XOR gate using CMOS

Complex CMOS gates

Example 10.9. *Draw a three input NOR using CMOS.*

A	B	f
0	0	1
0	1	0
1	0	0
1	1	0



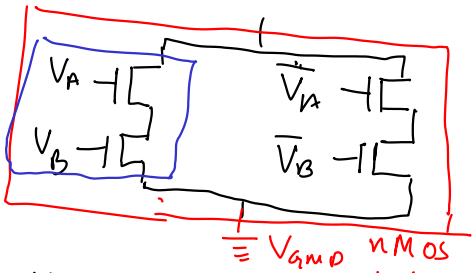
① Design nMOS pull down

$g = \bar{A}\cdot\bar{B} + A\cdot B$

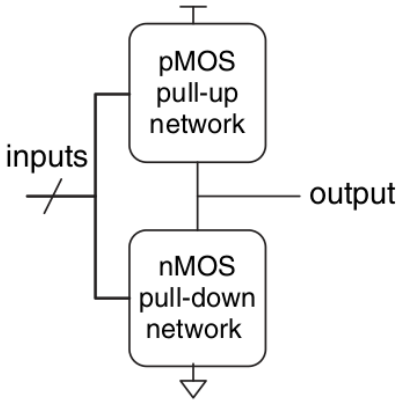
• ≡ Series

+ ≡ Parallel

A	B	$g = \bar{f}$
0	0	1
0	1	0
1	0	0
1	1	1



Example 10.10. *Draw a two input AND gate using CMOS.*



② Design pMOS pull up

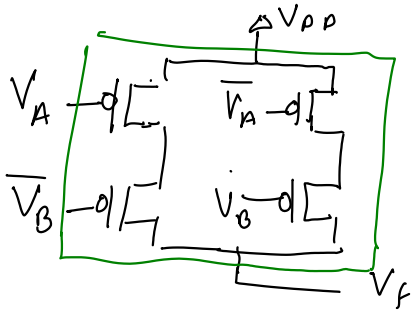
$A' = \bar{A}$	$B' = \bar{B}$	h
1	1	0
1	0	1
0	1	1
0	0	0

A	B	h = f
0	0	0
0	1	1
1	0	1
1	1	0

$h = A'B' + \bar{A}\bar{B}$ (SOP)

• ≡ Series

+ ≡ Parallel



pMOS pull up (SOP)

1. Use AND gate == Switches in series and OR gate == Switches in parallel to design the nMOS pull-down network and the pMOS pull-up network

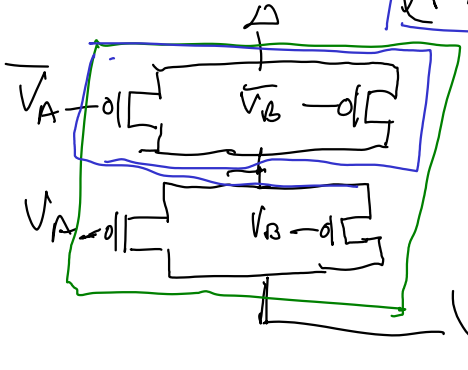
2. When nMOS network is ON, then the output is low. So you need to design nMOS network for the inverted function.

3. You can take DeMorgan's inverse of nMOS network to create the pMOS network.

$g = \bar{f} \Rightarrow h = f = \bar{g}$

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

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

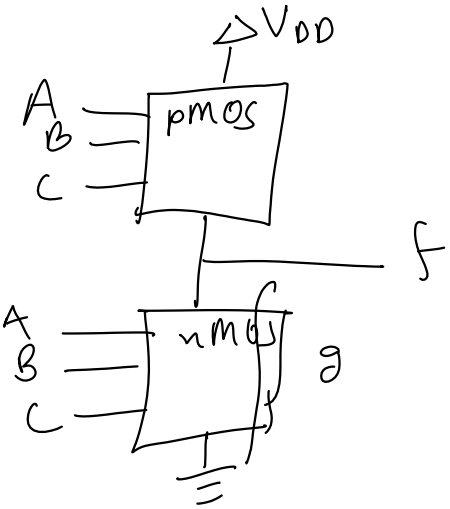


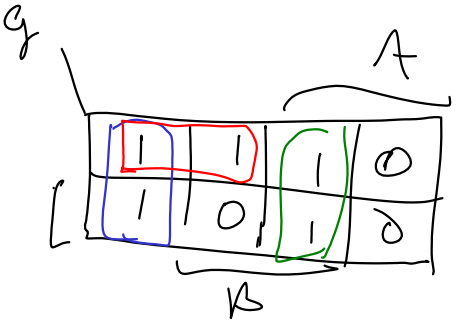
pMOS pull up (POS)

April 18, 2025

- Design a complex CMOS gate for:

A	B	C	f	$g = \bar{f}$
0	0	0	0	1
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	1	0
1	0	1	1	0
1	1	0	0	1
1	1	1	0	1

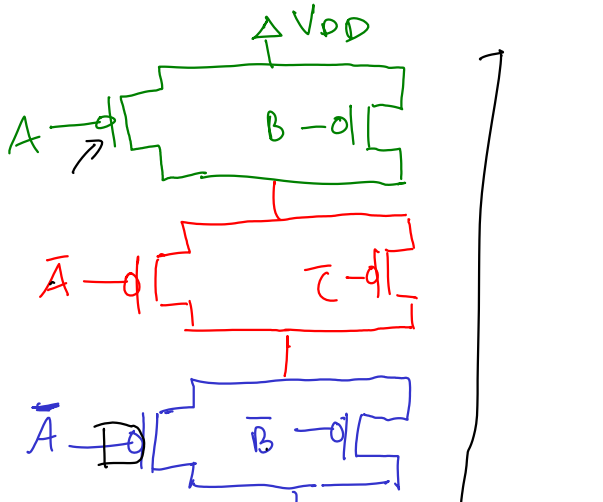




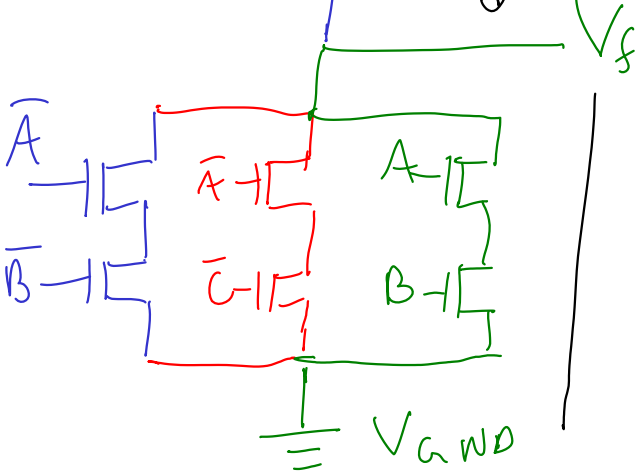
$$g = \bar{A}\bar{B} + \bar{A}\bar{C} + AB$$

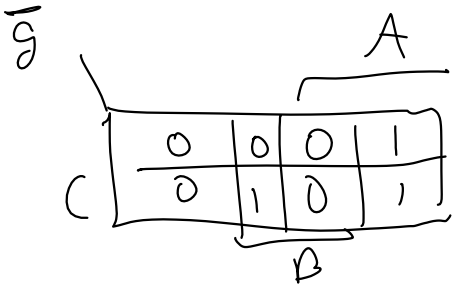
$$h = \bar{g} = (A+B)\cdot(A+C)\cdot(\bar{A}+\bar{B})$$

pMOS pull up side



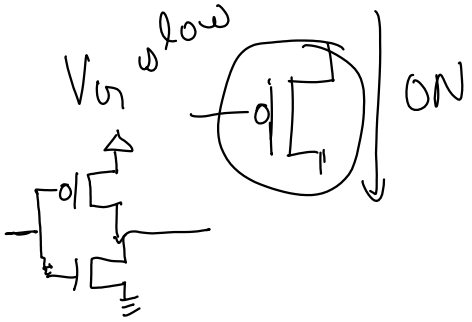
nMOS pull down

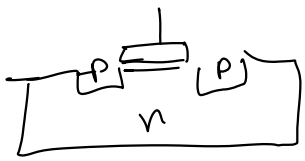




$$\bar{g} = A\bar{B} + \bar{A}BC$$

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

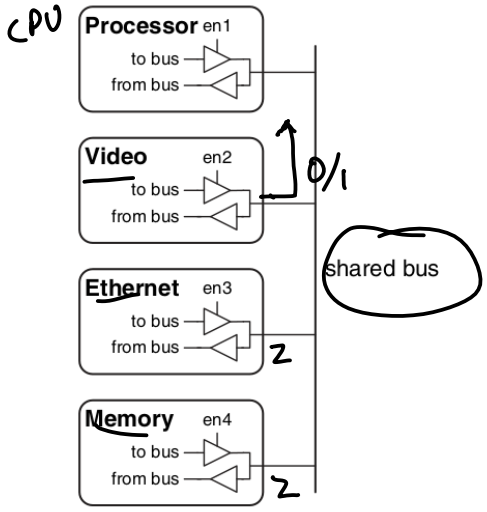




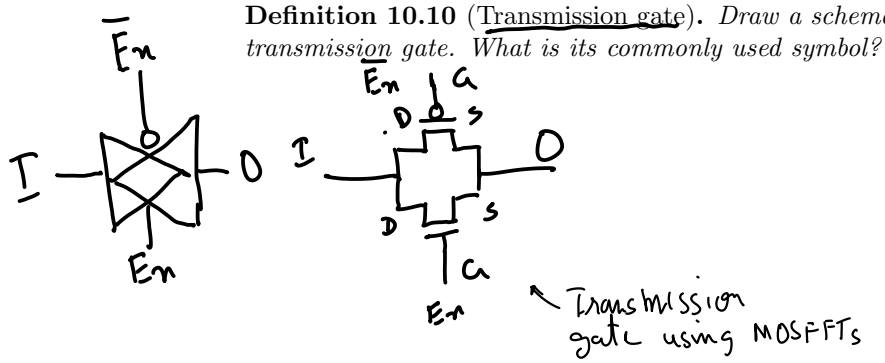
We cannot expect the inverted inputs to come for free on a low-level CMOS design. So we count the number of transistors needed for inverting inputs.

This design needs 18 (6 for inversions / 12 for design transistors

10.6.1 Gates with floating output



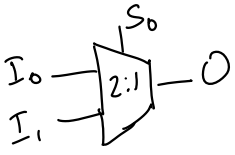
Definition 10.10 (Transmission gate)**.** *Draw a schematic of transmission gate and truth table for transmission gate. What is its commonly used symbol?*



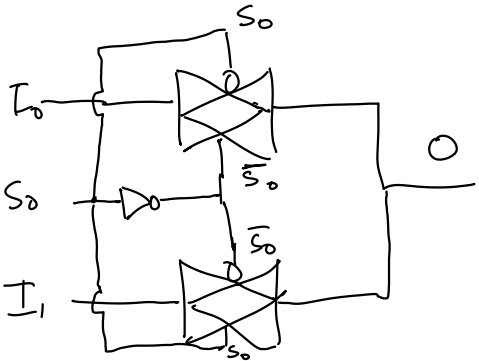
En	I	O
0	0	Z
0	1	Z
1	0	0
1	1	1

Definition 10.11 (Tristate buffer)**.** *What is tristate buffer? Draw it's symbol and truth table? Where is it used?*

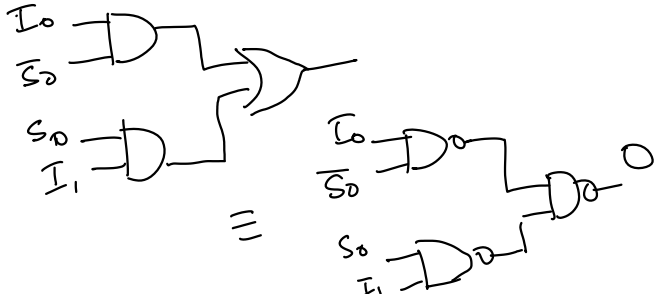
Example 10.11. *Draw a Multiplexer using transmission gates.*



$$O = \begin{cases} I_1 & \text{if } S_0 = 1 \\ I_0 & \text{if } S_0 = 0 \end{cases}$$



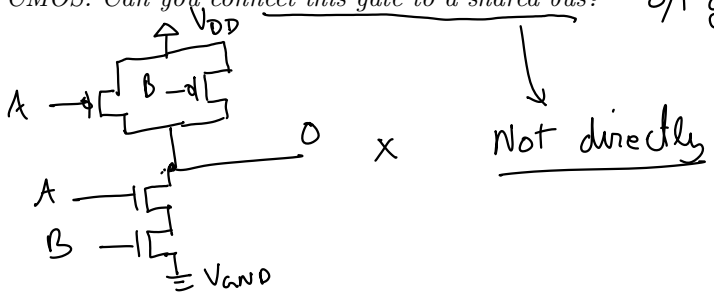
$$O = \overline{S_0} I_0 + S_0 I_1$$



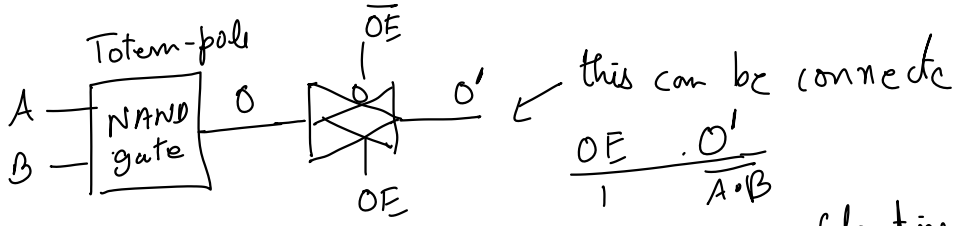
Example 10.12. *Draw a Multiplexer using tristate buffers.*
Do it yourself !

open collector / Tristate

Definition 10.12 (Totem-pole). *Draw a Push-pull (or Totem-pole) output NAND gate using CMOS. Can you connect this gate to a shared bus?* o/p gate

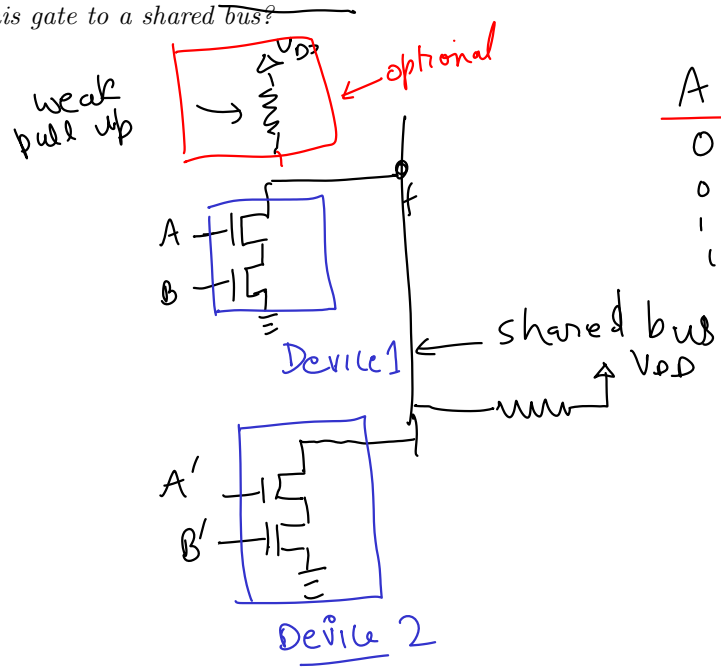


Definition 10.13 (Tristate). *Draw a Tristate output NAND gate using CMOS with an output enable (OE) input. Can you connect this gate to a shared bus?* at least some output must be floating



OE	O'
1	A·B
0	z ← floating

Definition 10.14 (Open-collector). *Draw a open-collector output NAND gate. Can you connect this gate to a shared bus?*



A	B	f
0	0	z
0	1	z
1	0	z
1	1	0

wire A = z;

10.7 Verilog truth tables

Table 11-11—Bitwise binary AND operator

&	0	1	x	z
0	0	0	0	0
1	0	1	x	x
x	0	x	x	x
z	0	x	x	x

A→ floating
B ↓
undefined →
undef

f = A&B
f=0 if either A or B is 0
f=1 if both A and B are 1

Table 11-12—Bitwise binary OR operator

|	0	1	x	z
0	0	1	x	x
1	1	1	1	1
x	x	1	x	x
z	x	1	x	x

undef

f = A|B
f=1 if either A or B are 1
f=0 if both A and B are 0

Bibliography

[1] Sarah L Harris and David Harris. *Digital design and computer architecture*. Morgan Kaufmann, 2022.

[2] Randy Katz and Gaetano Barriello. *Contemporary Logic Design*. Prentice Hall, 2004.

[3] Brown Stephen and Vranesic Zvonko. *Fundamentals of digital Logic with Verilog design*. McGraw Hill, 2022.