Digital Integrated Circuit Design

CMPE 530/630

MOSFET Switch Model and Circuits

Dr. Cory Merkel

Learning Objectives

- Review basic functionality of metal oxide semiconductor field effect transistors (MOSFETs)
  - Differences between p-type MOSFETs (PMOS) and n-type MOSFETs (NMOS)
- Understand the tradeoffs between pass transistors, transmission gates, and complementary metal oxide semiconductor (CMOS) logic circuits
- Be able to convert a logic function into a CMOS, pass transistor, or transmission gate logic circuit and vice versa

Why Digital Logic?

- Advantages
  - Better noise tolerance
  - Higher precision (can be arbitrarily high!)
  - Easier detection/correction of errors
  - Simpler/more robust circuits

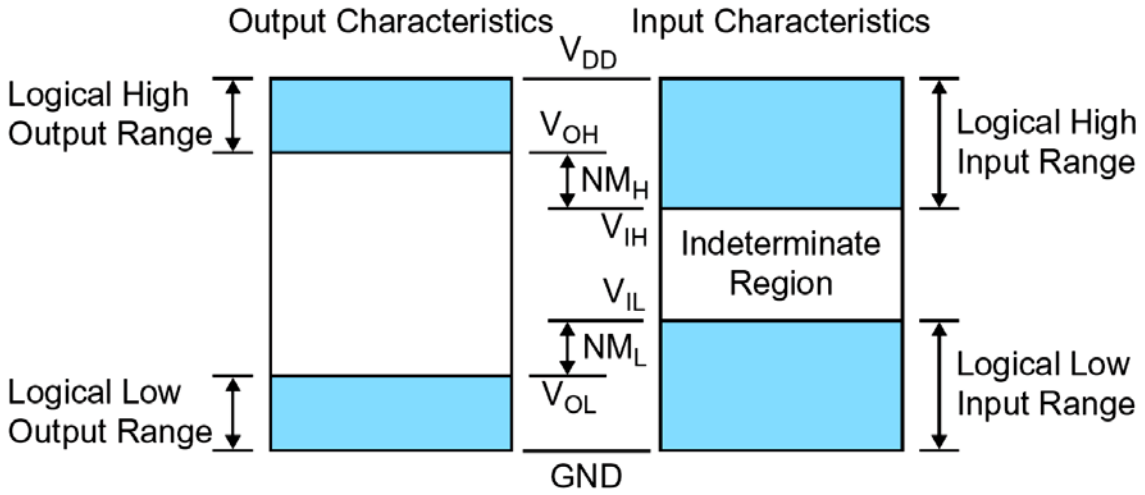
- Disadvanges
  - For some functions analog designs are much simpler and much more size, weight, and power (SWaP)-eficient

Representation of Digital Signals

- Typically, we represent a logical '1' with V_{DD} and a logical '0' with ground
- We need to give our circuits some "wiggle room", though:

V_{OH}=High output voltage V_{OL}= Low output voltage NM_H= High noise margin

V_{IH}= High input voltage V_{IL}= Low output voltage NM_L = Low noise margin

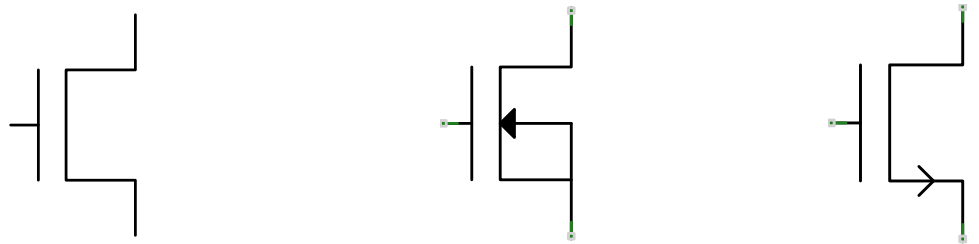


Our Friend, the MOSFET

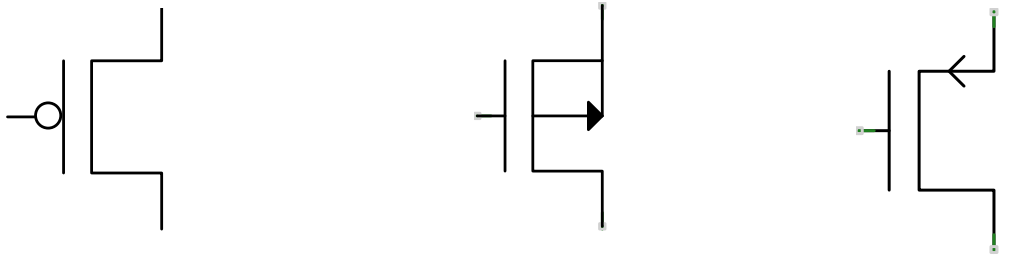
» Metal Oxide Semiconductor Field Effect Transistor
  - We will fully understand this acronym in a couple weeks

» MOSFET Symbols:

NMOS



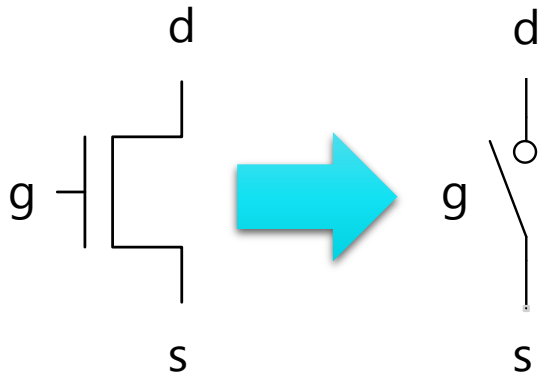
PMOS



Our Friend, the MOSFET

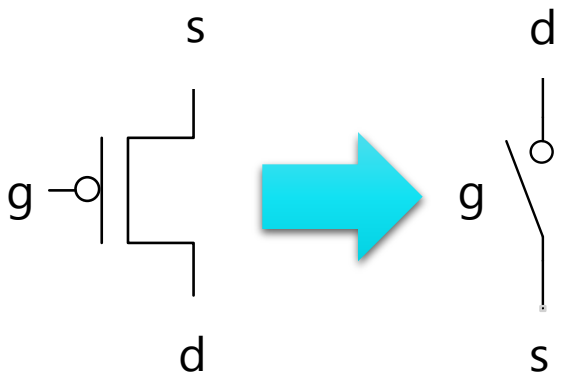
» For now, MOSFET = fancy switch

NMOS



g = 0: Switch OFF
g = 1: Switch ON

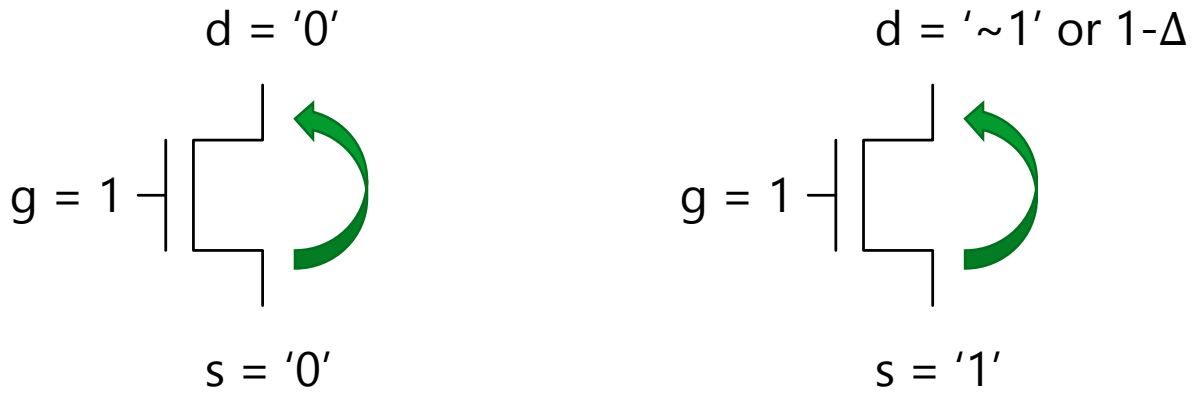
PMOS



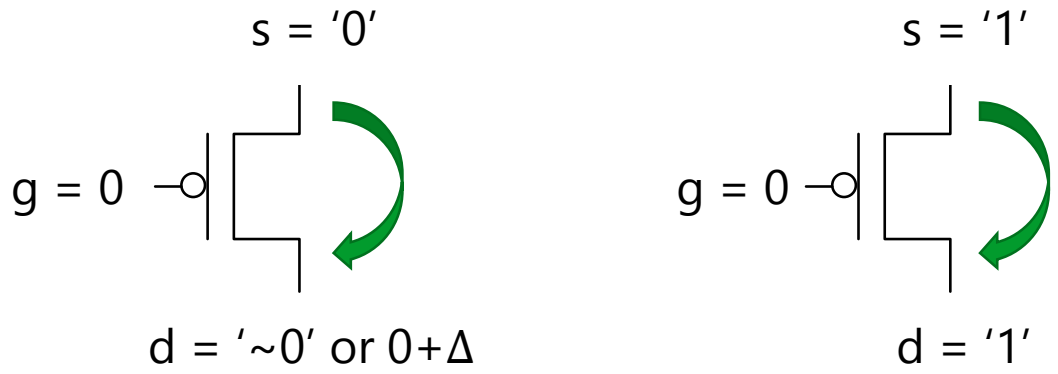
g = 0: Switch ON
g = 1: Switch OFF

Our Friend, the MOSFET

- NMOS passes a strong ‘0’ from source to drain and a weak ‘1’ from source to drain:



- PMOS passes a strong ‘1’ from source to drain and a weak ‘0’ from source to drain:

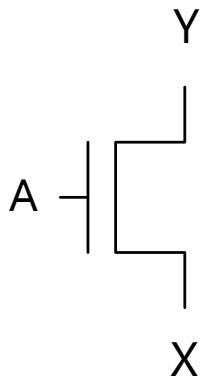


Δ is a small value less than 1. We will see exactly what it is later in the course.

Series and Parallel MOSFETs

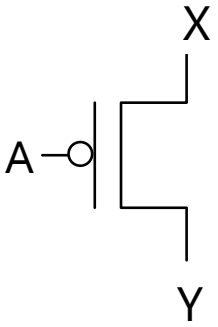
Single

NMOS



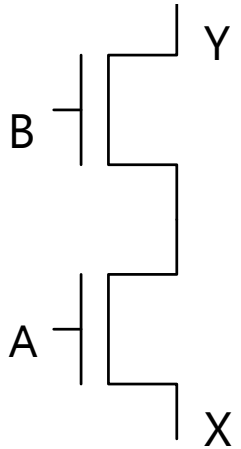
A	X	Y
0	0	Z
0	1	Z
1	0	0
1	1	~1

PMOS

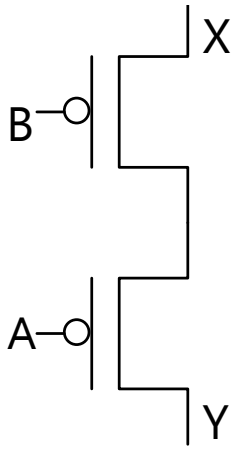


A	X	Y
0	0	~0
0	1	1
1	0	Z
1	1	Z

Series

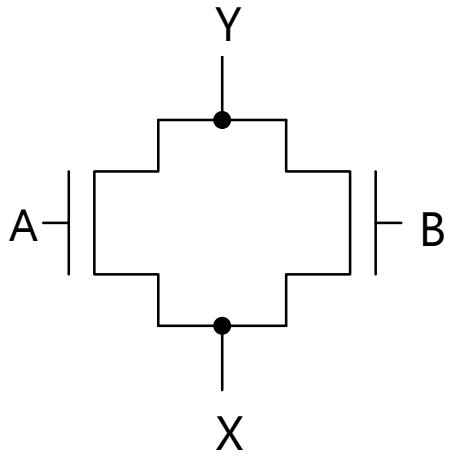


A	B	X	Y
0	0	0	Z
0	0	1	Z
0	1	0	Z
0	1	1	Z
1	0	0	Z
1	0	1	Z
1	1	0	0
1	1	1	~1

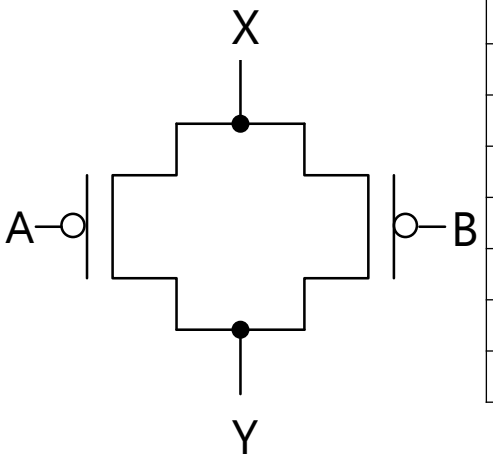


A	B	X	Y
0	0	0	~0
0	0	1	1
0	1	0	Z
0	1	1	Z
1	0	0	Z
1	0	1	Z
1	1	0	Z
1	1	1	Z

Parallel



A	B	X	Y
0	0	0	Z
0	0	1	Z
0	1	0	0
0	1	1	~1
1	0	0	0
1	0	1	~1
1	1	0	0
1	1	1	~1



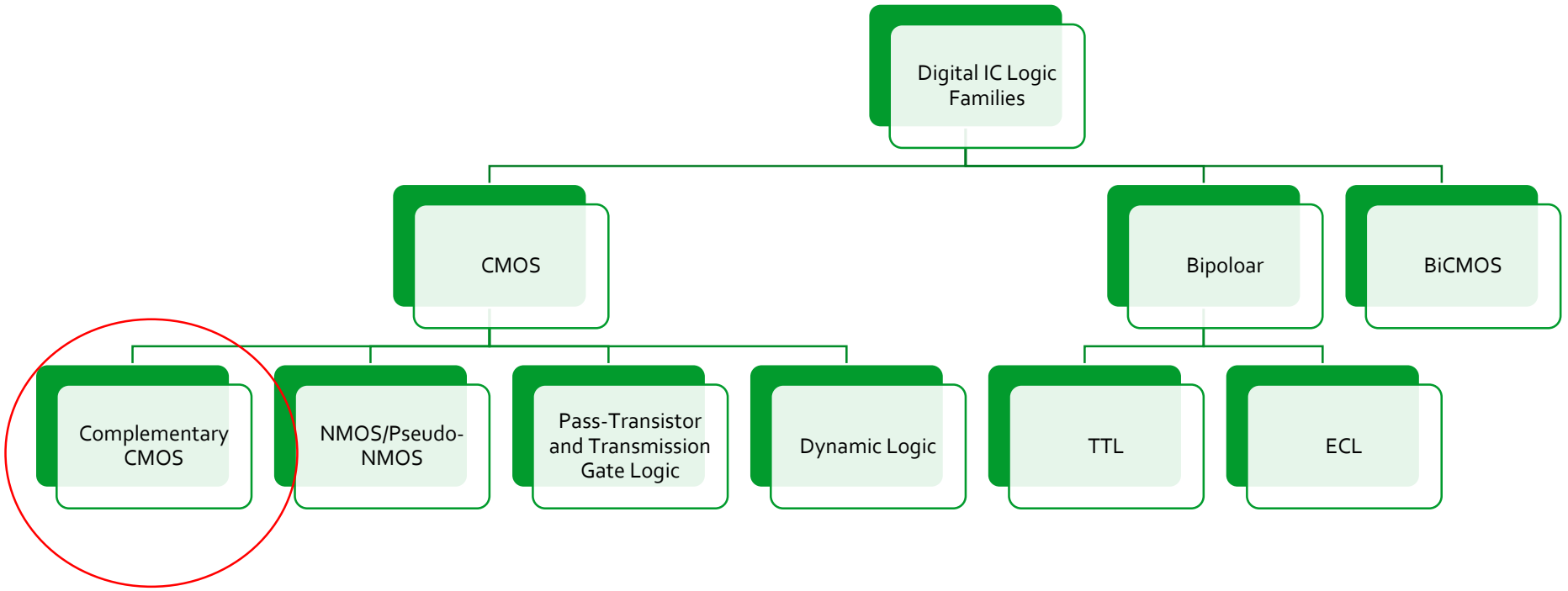
A	B	X	Y
0	0	0	~0
0	0	1	1
0	1	0	~0
0	1	1	1
1	0	0	~0
1	0	1	1
1	1	0	Z
1	1	1	Z

Series and Parallel MOSFETs

- When X=0
  - Series NMOS behave like NAND
  - Parallel NMOS behave like NOR

- When X=1
  - Series PMOS behave like NOR
  - Parallel PMOS behave like NAND

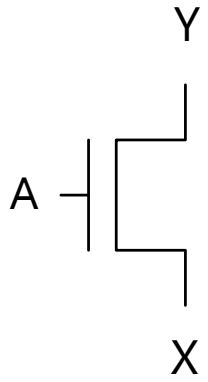
Logic Styles



Primary focus in this course.

Design an Inverter Using an NMOS Transistor

» Recall:

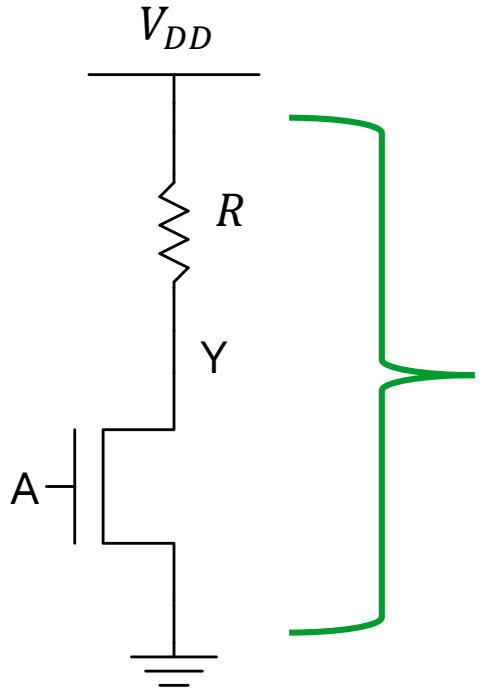


A	X	Y
0	0	Z
0	1	Z
1	0	0
1	1	~1

We want:

A	Y
0	1
1	0

» Could do this:



When A=0, there is no path to ground, so Y≈ V_{DD} (i.e. '1').

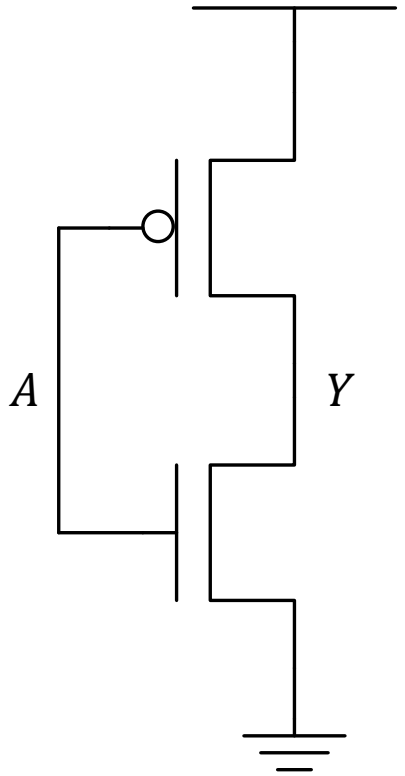
When A=1, the switch is closed, so Y≈ Ground (i.e. 'o')

Several issues with this circuit:

- Resistors are large
- High power consumption
- Reduced noise margins
- Slow if R is large

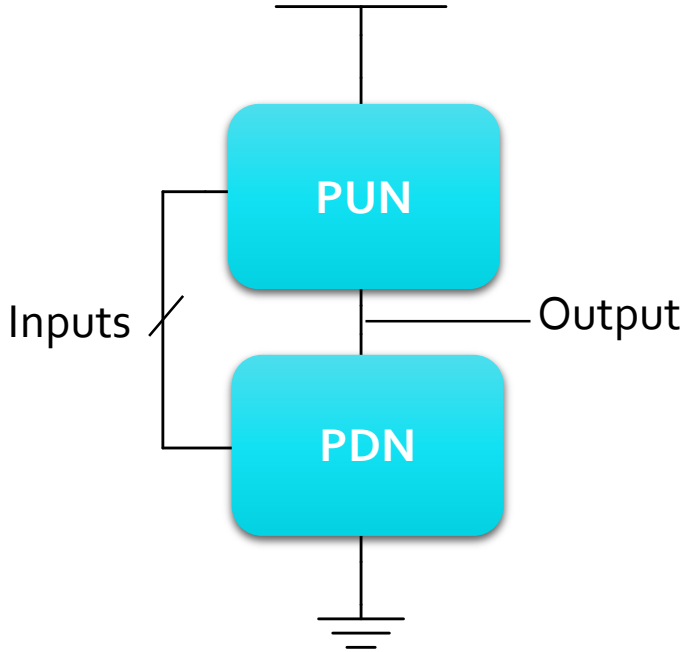
Now Try Using an NMOS and a PMOS Transistor

- When A=0, the PMOS is ON and NMOS is OFF
  - Y is "pulled up" to V_{DD}
- When A=1, the NMOS is ON and the PMOS is OFF
  - Y is "pulled down" to ground
- Power consumption is reduced
  - Why?
- This is an example of a complementary or static CMOS logic gate



CMOS Logic

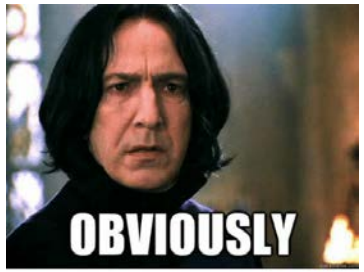
» In general, we want:
- Output connected to V_{DD} through PMOS transistors – Pull-Up Network (**PUN**)
- Output connected to ground through NMOS – Pull-Down Network (**PDN**)
- Only one supply path (PUN or PDN) on at a time



CMOS Logic

- How do we make sure only one supply path (PUN or PDN) is on at a time?
- Easy! Given some logic function Y, we will write it in complemented form (e.g. with a bar over it):

$$Y = \overline{\overline{Y}}$$

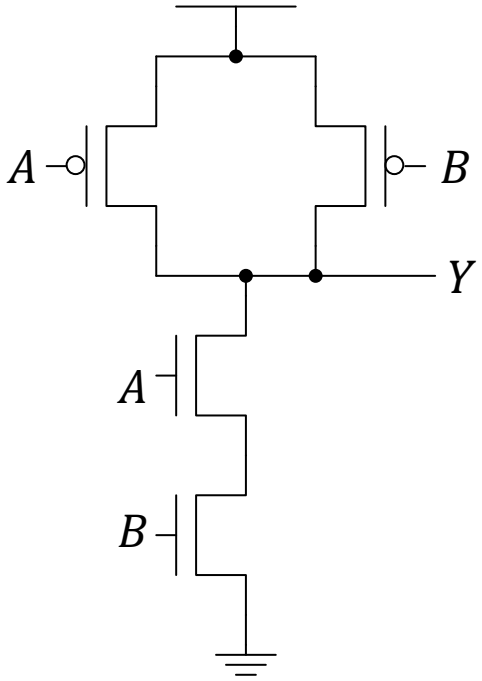


- For every **AND** in $\overline{Y}$, there is a **series NMOS** and **parallel PMOS**
- For every **OR** in $\overline{Y}$ there is a **parallel NMOS** and **series PMOS**

Example – NAND Gate

» $Y = \overline{AB}$ → Already in complemented form!

» $\overline{Y} = AB$, so we have a series of NMOS and parallel PMOS:



Example – Compound Logic Gate

» Try these on your own:

- $Y = AB + C$
- $Y = \overline{(A + B)C}$

Symmetric Logic Functions

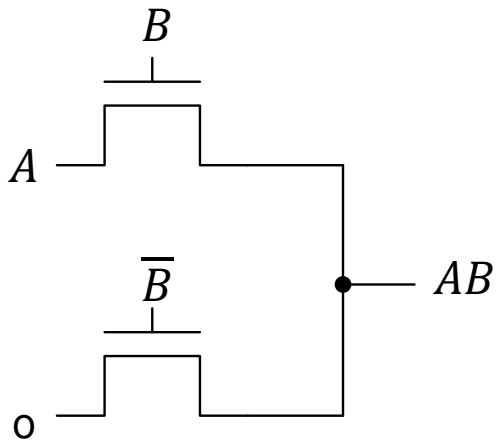
- When a logic function is symmetric, complementing its inputs complements its outputs
  - Complementing the inputs leads to complemented outputs

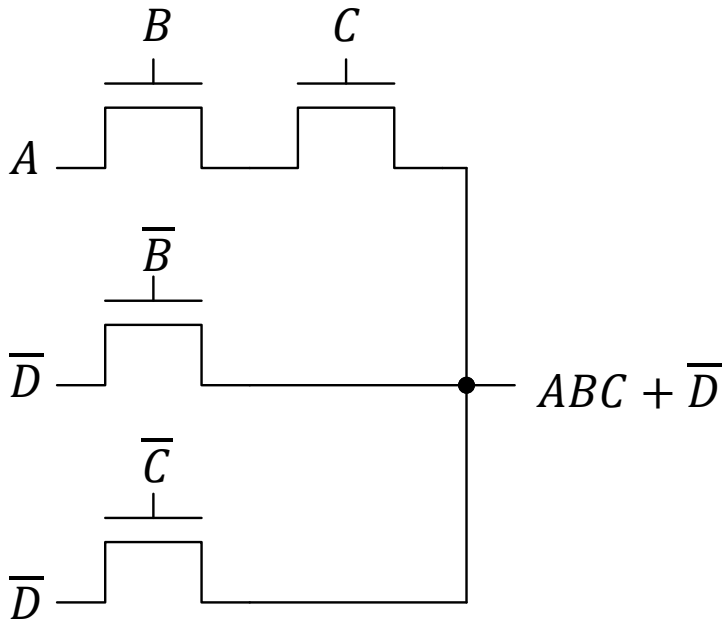
- In this case, the PUN and PDN are also symmetric or "mirrored"
  - Makes mask design simpler

Pass Transistor Logic

» Pass transistor: Used to pass an *input*, rather than a supply value from input to output

» Examples:





» Important: Need a low-impedance path to supply for all possible input combinations!

Differential Pass Transistor Logic

- Accepts true and complemented inputs
- Produces true and complemented outputs
- Eliminates need for inverters

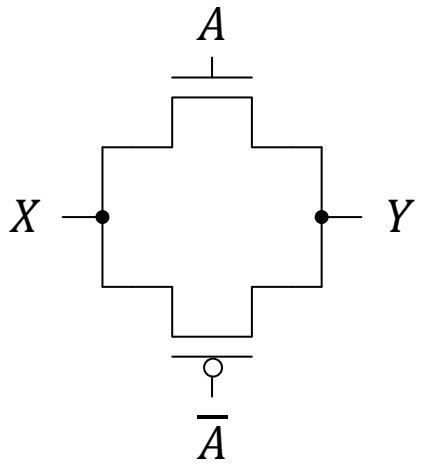
Pass Transistor Logic

» Advantages:
- Reduces number of components compared to complementary CMOS
- Potentially faster due to less capacitance

» Disadvantages
- NMOS passes a weak '1' and PMOS passes a weak '0'
- Can't drive other pass transistor logic circuits
- Noise

Transmission Gates (TG)

» Overcome the weak 0/1 issue by using both NMOS and PMOS in parallel:



» When A=1, a strong 1/0 is passed to Y

» Disadvantage: More transistors

Example: TG MUX

» Try on your own:
- $Y = \overline{S}D_0 + SD_1$

Tri-State Inverter

- Used when multiple circuits drive a single output
  - Example: Shared bus