Lecture Outline

Reminders to self:

- ❑ Turn on lecture recording to Cloud
- ❑ Turn on Zoom microphone

- Last Lecture
  - Hazards in combinatorial logic
  - Started Multiplexers
- Today's Lecture
  - Continue Multiplexers
  - Brief look at CMOS gates at transistor level
  - Three-state buffers / Tri-state outputs

Handouts and Announcements

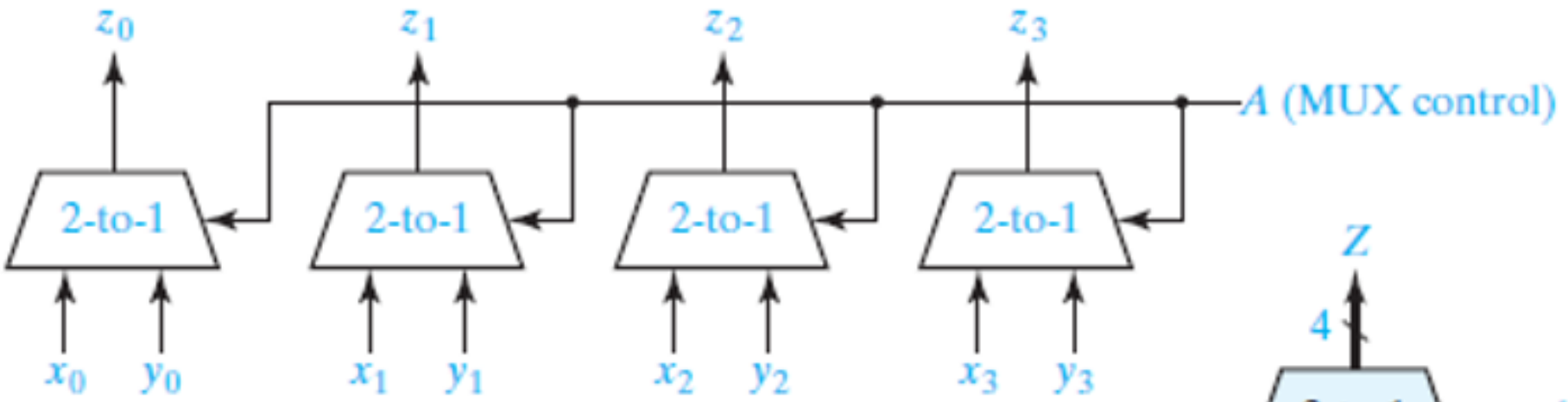
- Announcements
  - Homework: No new assignment today
  - Homework Reminder:
    - HW 8-1 and 8-2 Due: 11:59pm Thursday 2/16
  - Participation Quiz 7 available starting 12:25pm today
    - Due 12:25pm Thursday 2/16
    - Available until 12:25pm Friday 2/17 with late penalty
  - Read for Friday: pages 271-275

Handouts and Announcements

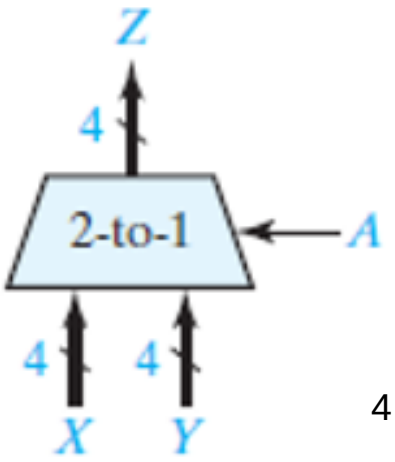
- Announcements
  - Mini-Exam 3 Reminder
    - Available 5pm Monday 2/20 through 5:00pm Tuesday 2/21
    - Due in Carmen PROMPTLY at 5:00pm on 2/21
    - Designed to be completed in ~36 min, but you may use more
    - When planning your schedule:
      - I recommend building in 10-15 min extra
      - To allow for downloading exam, signing and dating honor pledge, saving solution as pdf, and uploading to Carmen
    - I also recommend not procrastinating
  - Exam review topics available on Carmen
  - Sample Mini-Exams 3 and 4 from Au20 also available

Multiplexers

- Multiplexers often used to select data which is to be processed or stored in digital system design
- *N* identical MUX connected in parallel, with shared control, can be used to select between *N*-bit data words
- A 4-bit example:
  - (x_3, x_2, x_1, x_0) and (y_3, y_2, y_1, y_0) are the two 4-bit data words to select between
  - (z_3, z_2, z_1, z_0) is the output data word

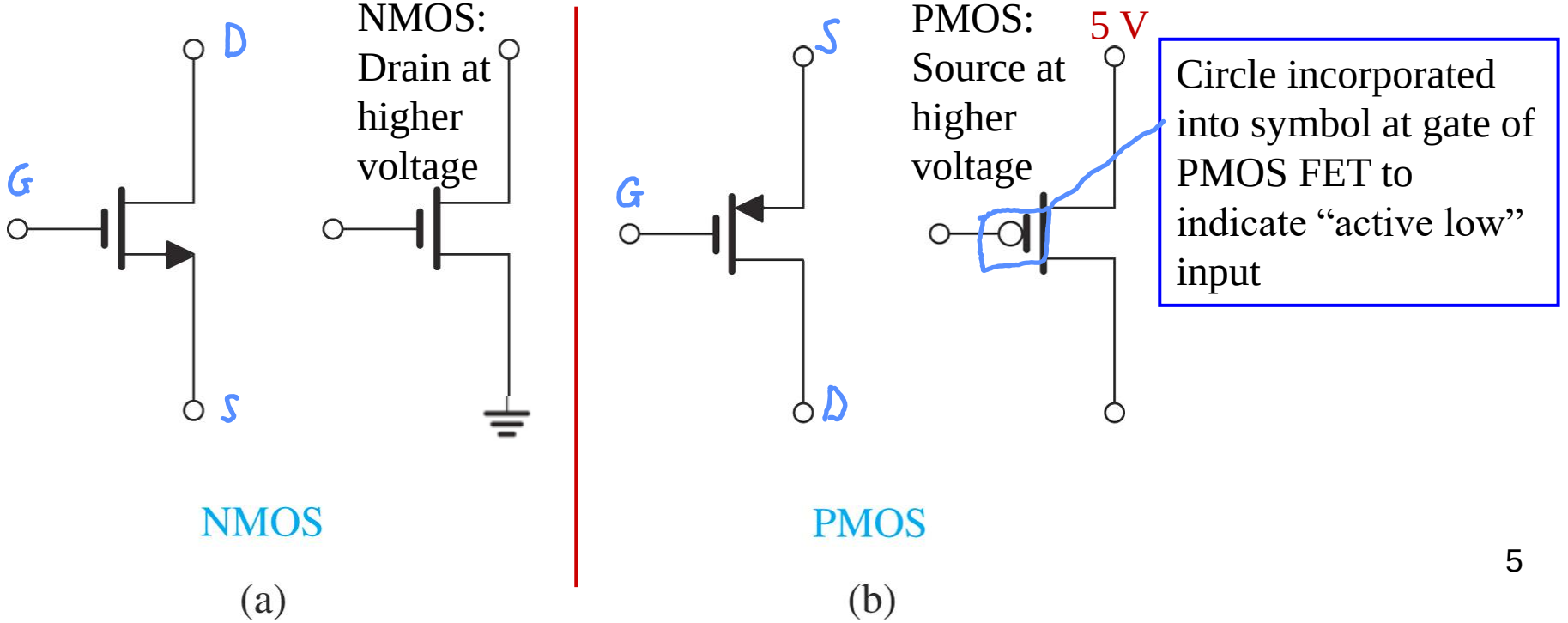


- Compact notation showing 4-bit buses at inputs and output ⇨
  - Thick line to represent bus
  - Diagonal mark with number designates number of bits on bus

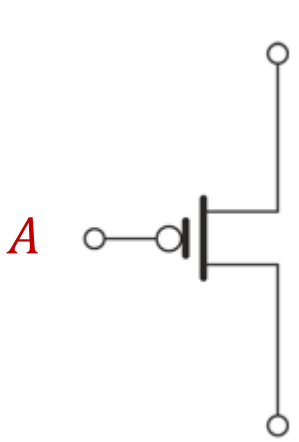


Digital MOSFET circuit symbols

- MOSFET is a type of transistor, dominant in modern digital ICs
- Two varieties: NMOS and PMOS
- Three terminal device: Gate, Drain, Source (arrow in symbol)
- Digital-circuit designers often use a different circuit symbol for the MOSFET when it is used for implementing logic
- Which terminal is source and which is drain inferred from circuit:



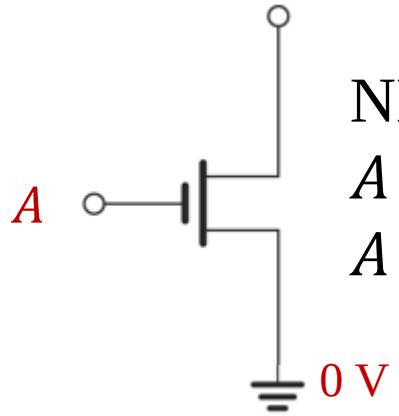
PMOS + NMOS = CMOS



PMOS

$A = 0$: 5 V gets through (Short)

$A = 1$: Nothing gets through (Open)



NMOS

$A = 1$: 0 V gets through (Short)

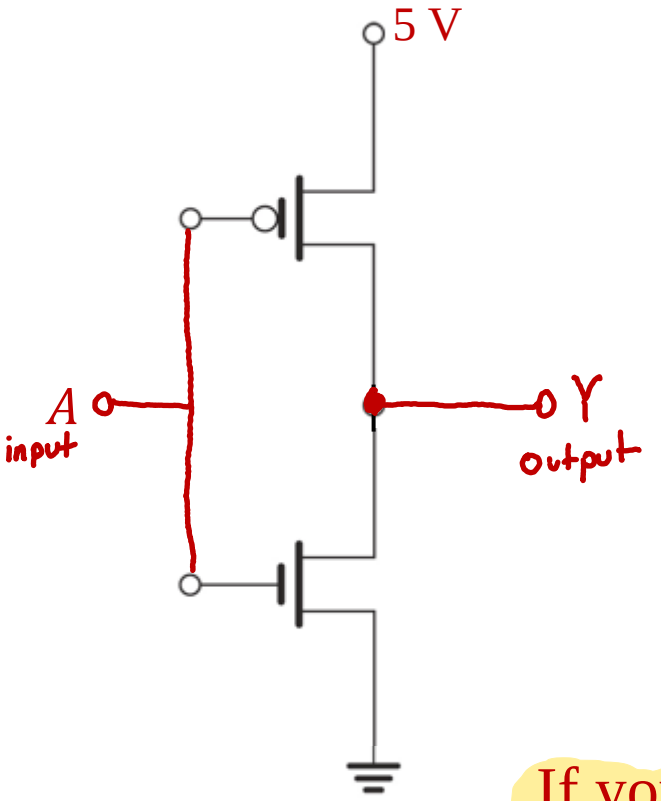
$A = 0$: Nothing gets through (Open)

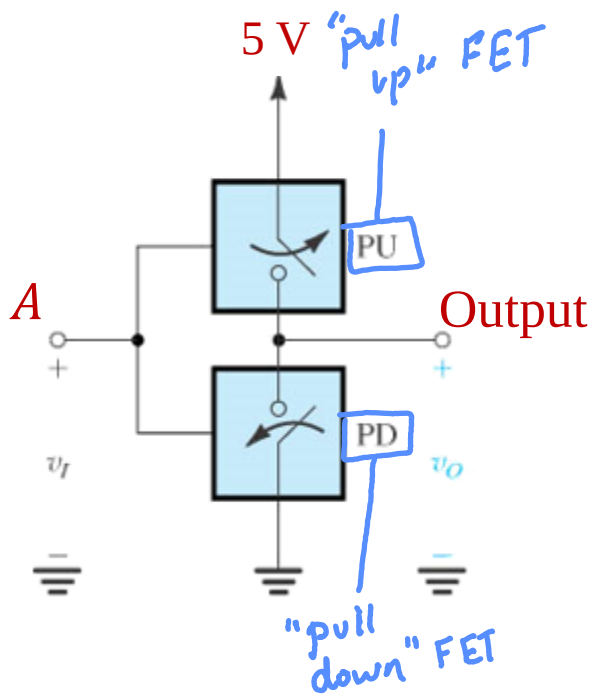
PMOS + NMOS = CMOS

C stands for "complimentary"

CMOS Inverter

Sketch in wires for input and output





$v_A = 0$ V:
PMOS = On (short)
NMOS = Off (open)
Output = 5 V
"Pulled up" to 5 V

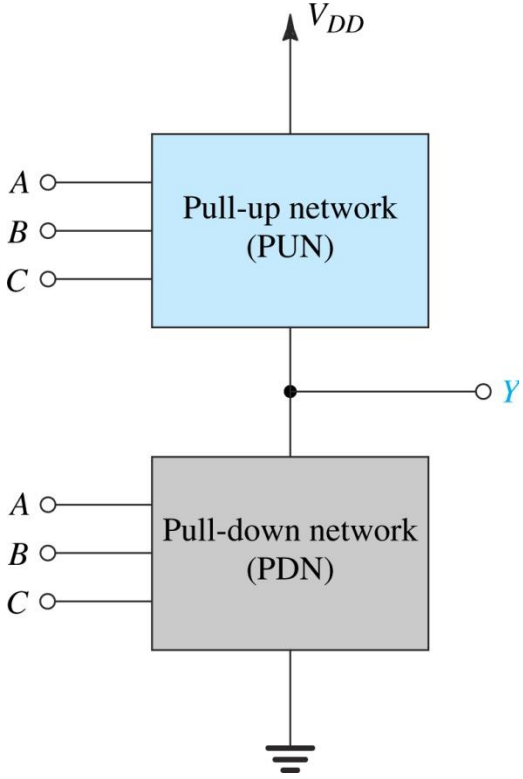
$v_A = 5$ V:
PMOS = Off (open)
NMOS = On (short)
Output = 0 V
"Pulled down" to ground

If you were to turn on both FETs

- Tries to short power supply to ground
- Large current flow and power dissipation
- Heat and damage

CMOS Logic-Gate Circuits

- Derived from CMOS inverter, but individual transistors replaced by networks of transistors
  - PU FET replaced by a Pull-Up Network (PUN) of PMOS FETs
  - PD FET replaced by a Pull-Down Network (PDN) of NMOS FETs
- Both networks operated by input variables, in complementary way similar to PMOS and NMOS



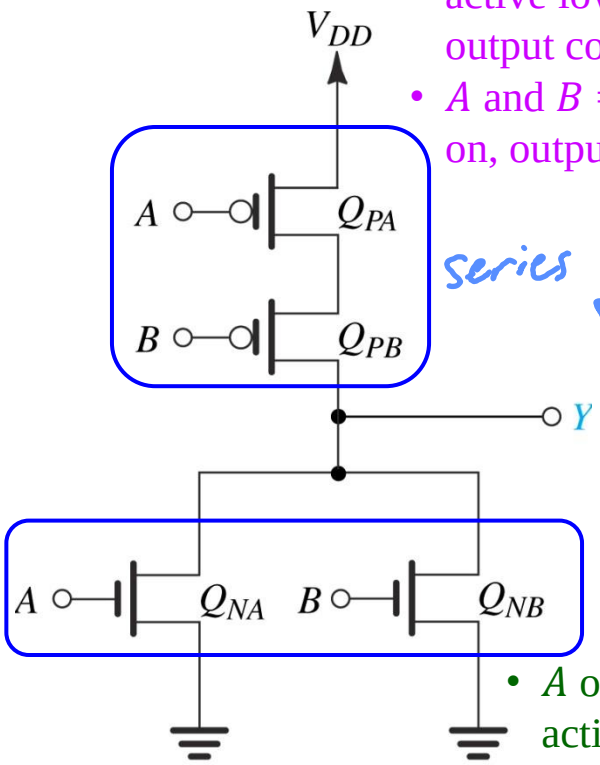
- PDN network establishes path to ground for input combinations that require low output ($Y=0$, $v_Y = 0$)
- At same time, PUN does **not** establish path to V_{DD}

- PUN network establishes path to V_{DD} for input combinations that require high output ($Y=1$, $v_Y = V_{DD}$)
- At same time, PDN does **not** establish path to ground

CMOS Logic-Gate Circuits

NOR and NAND

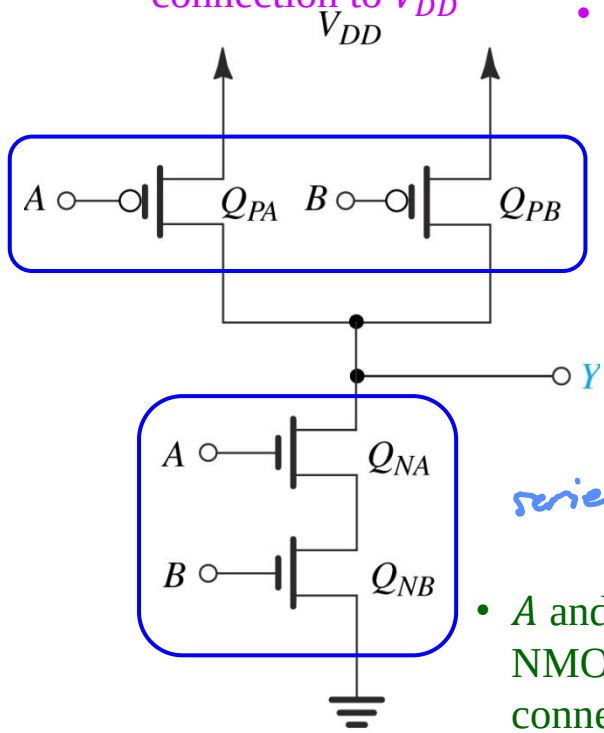
- A or B = 1: At least one active low PMOS off, no output connection to V_{DD}
- A and B = 0: Both PMOS on, output connected to V_{DD}



$Y = \overline{A + B}$

- A or B = 1: At least one active high NMOS on, output connected to ground
- A and B = 0: Both NMOS off, no connection to ground

- A and B = 1: Both PMOS off, no output connection to V_{DD}
- A or B = 0: At least one PMOS on, output connected to V_{DD}



$Y = \overline{AB}$

- A and B = 1: Both NMOS on, output connected to ground
- A or B = 0: At least one NMOS off, no connection to ground

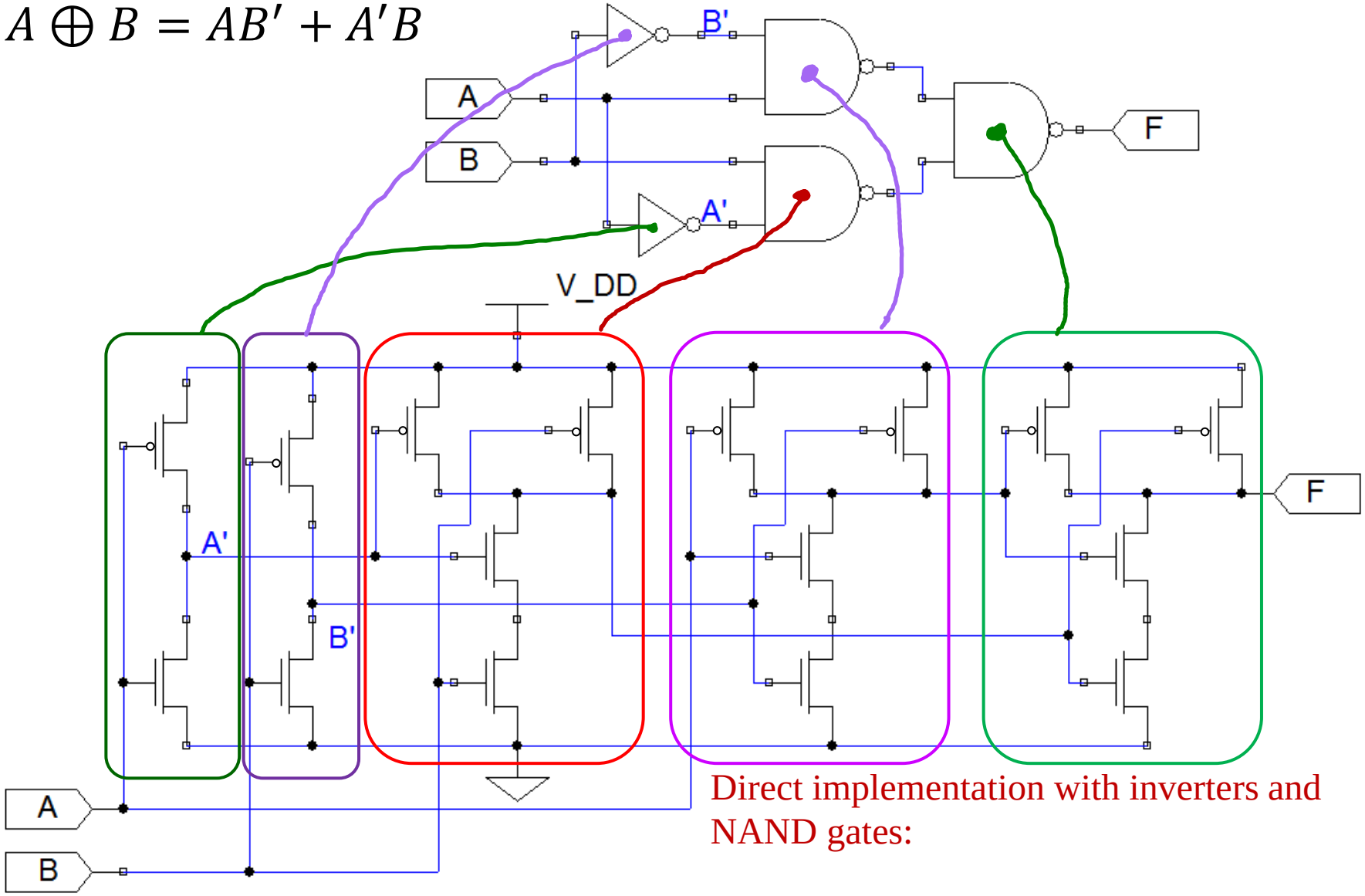
To make an OR, follow this with 2-FET inverter: 6 transistors

To make an AND, follow this with 2-FET inverter: 6 transistors

CMOS Logic-Gate Circuits

XOR

$$A \oplus B = AB' + A'B$$

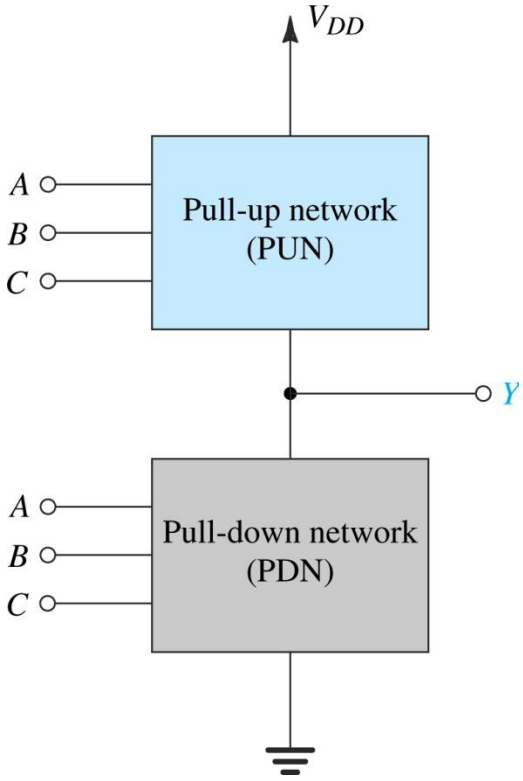


Direct implementation with inverters and NAND gates:

CMOS Logic-Gate Circuits

- But can do better by directly applying these PUN and PDN concepts
- Transistor-Level Design

Generic 3-input example



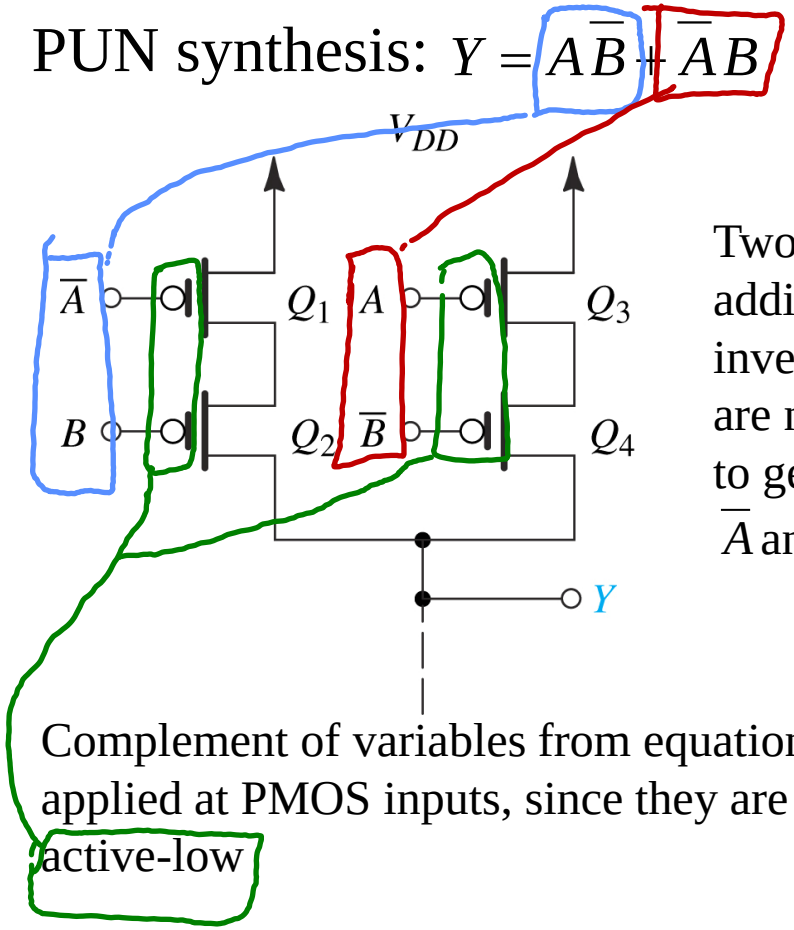
- PDN network establishes path to ground for input combinations that require low output ($Y = 0$, $v_Y = 0$)
- At same time, PUN does **not** establish path to V_{DD}

- PUN network establishes path to V_{DD} for input combinations that require high output ($Y = 1$, $v_Y = V_{DD}$)
- At same time, PDN does **not** establish path to ground

CMOS Logic-Gate Circuits

XOR – by direct synthesis of PUN and PDN

PUN synthesis: $Y = A\overline{B} + \overline{A}B$



Two additional inverters are needed, to generate $\overline{A}$ and $\overline{B}$

Complement of variables from equation applied at PMOS inputs, since they are active-low

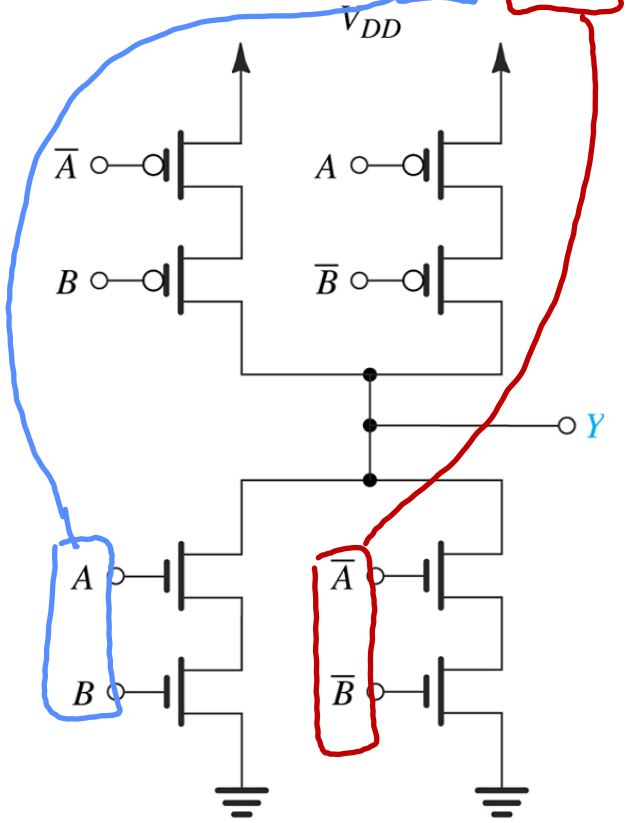
Including the two inverters: 12 transistors

Inverter delay + 1 gate-delay

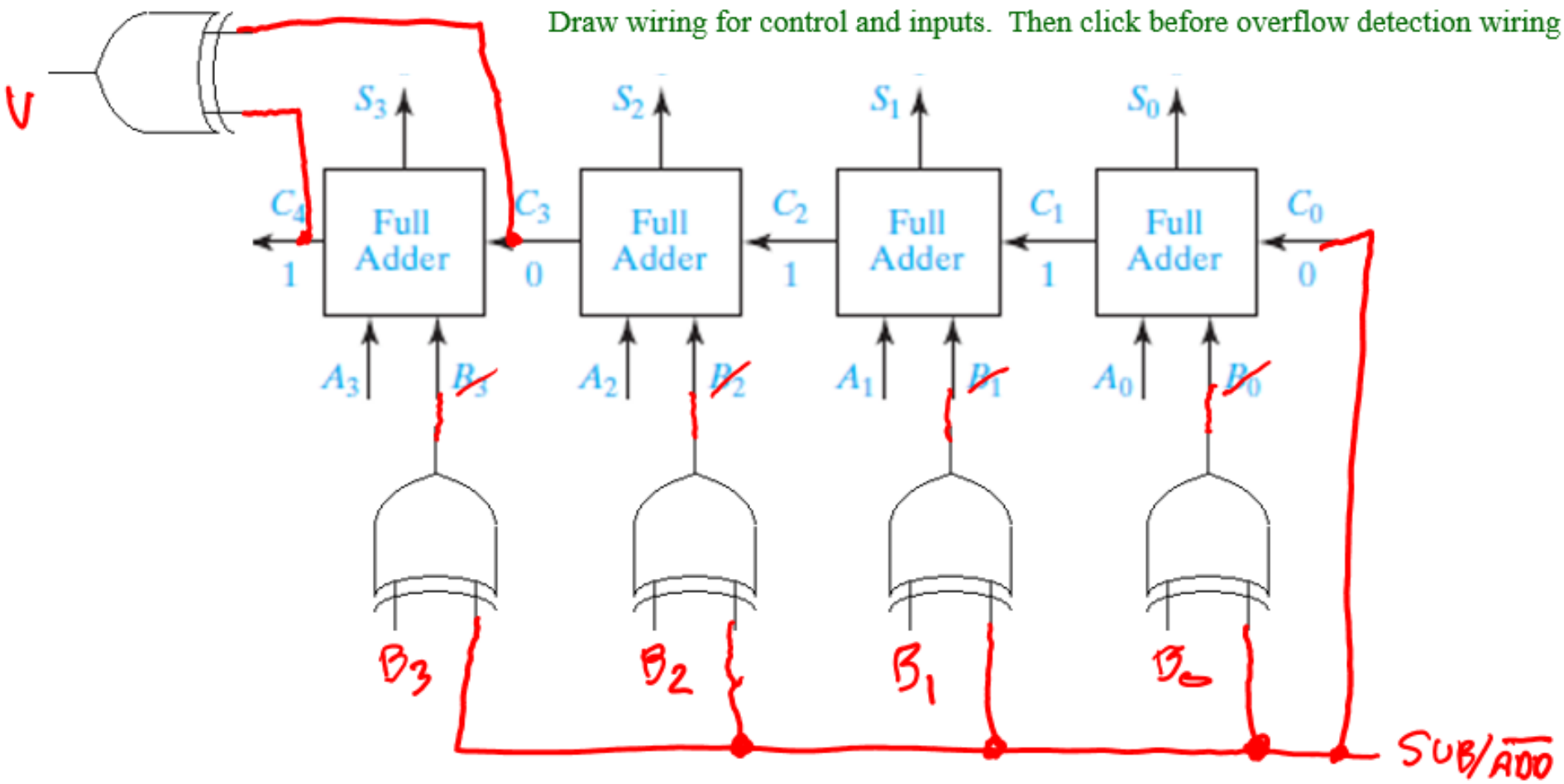
PDN synthesis: $\overline{Y} = \overline{A\overline{B} + \overline{A}B} = \overline{A\overline{B}}\,\overline{\overline{A}B}$

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

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



Adder/Subtractor



In a previous lecture saw an adder/subtractor circuit in which XOR gates were used to

- invert the bits of B when $SUB = 1$ (subtraction), and
- not invert them when $SUB = 0$ (addition)

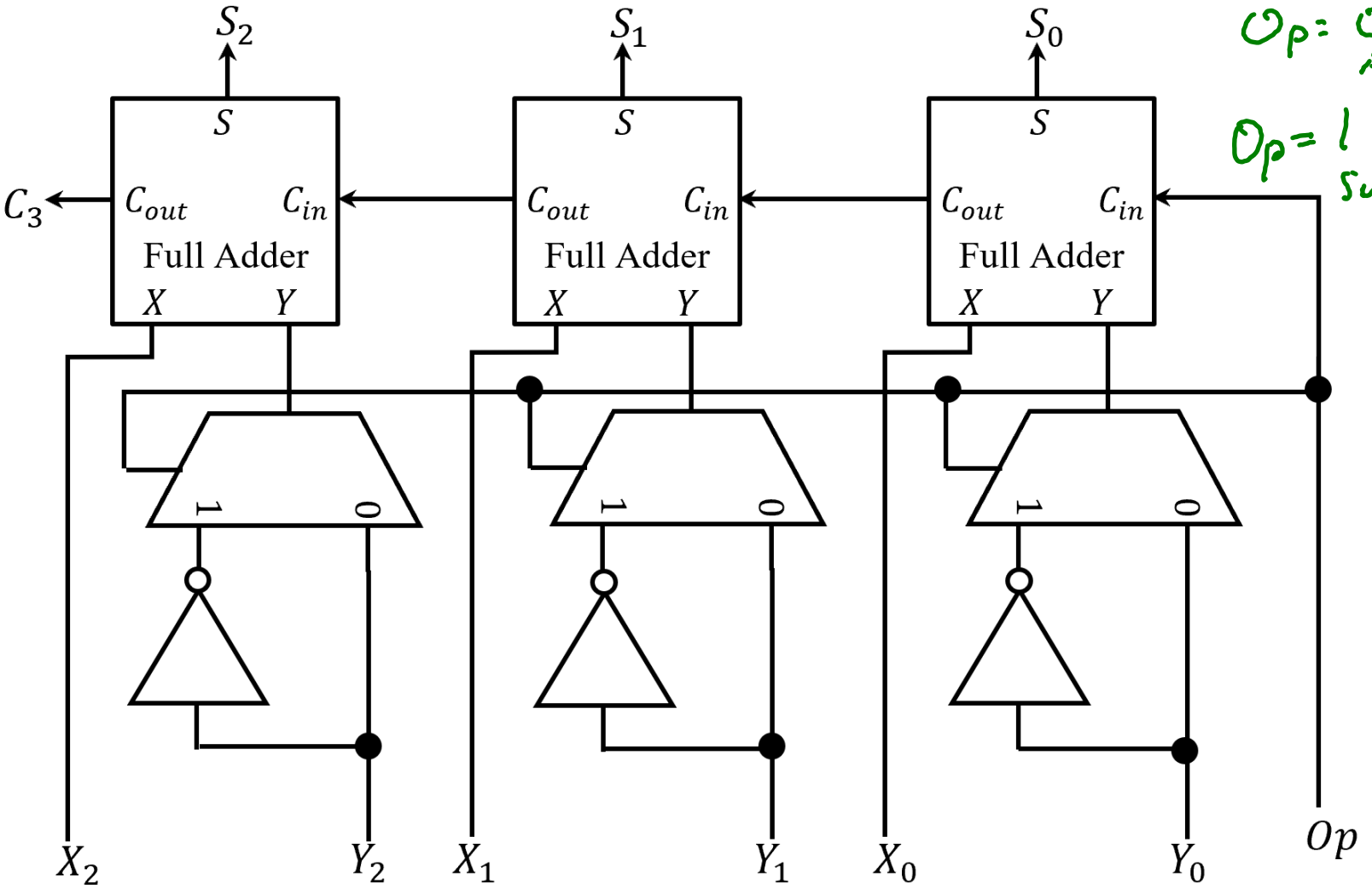
Adder/Subtractor

Here is a 3-bit version, implemented using multiplexers

Quiz 7

Op: 0 addition

Op: 1 subtraction

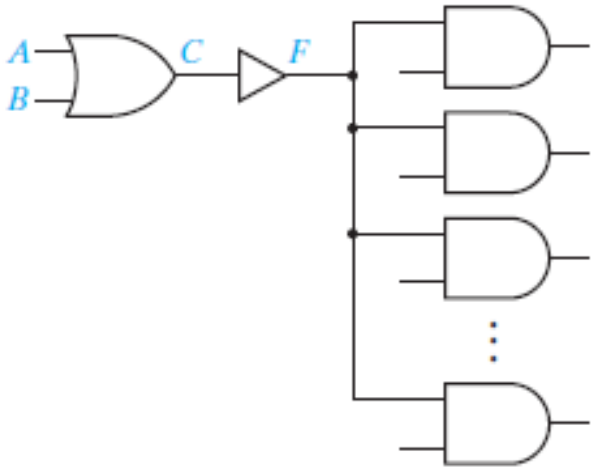


Three-State Buffers

Buffers

- A gate output can only be connected to a limited number of other device inputs without degrading the digital system's performance (Fan-out)
- A simple buffer may be used to increase the driving capability of a gate output
- Figure 9-10 shows a buffer connected between a gate output and several gate inputs

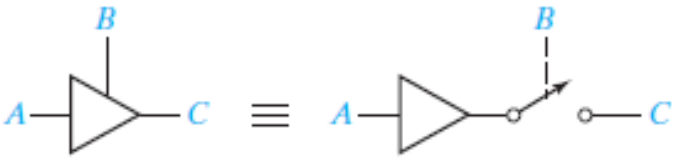
FIGURE 9-10
Gate Circuit with Added Buffer



Three-State Buffers

- Tying the outputs of two gates together is **BAD**
- Tying gate outputs ≠ OR
- Gates FORCE a wire to 1 or 0
- Gates FORCE the voltage on a wire to High (e.g. 3 V) or Low (e.g. 0 V)
- Three-state logic permits two or more outputs to be wired together
- In the third state the output of the gate is "high impedance"
  - Hi-*Z*
  - It is effectively disconnected from the logic circuit inside the gate
- Also known as "Tri-state" Buffers

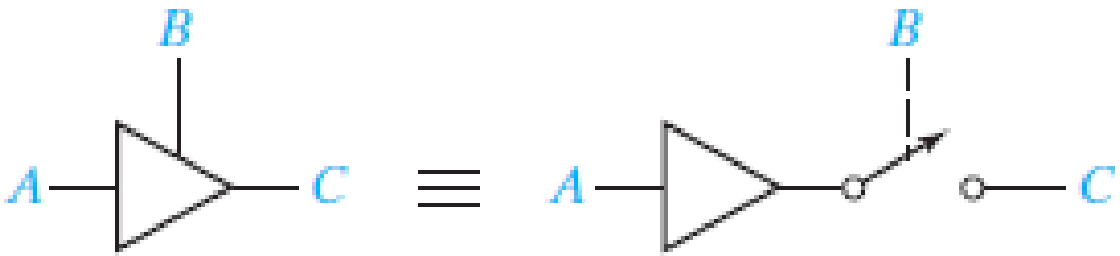
FIGURE 9-11
Three-State Buffer



Three-State Buffers

- Other reasons for adding a buffer (even if not tri-state)
  - Add a delay (for example: to keep two-level logic in time-step with three-level logic)
  - Increase output current driving capability (fan-out)
- Three-state buffer logic symbol & logical equivalent:

FIGURE 9-11
Three-State Buffer

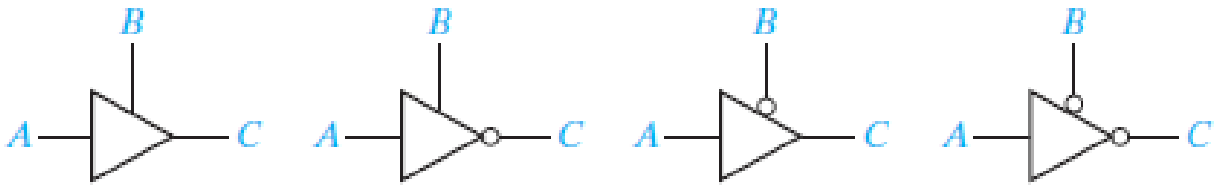


- Four types of three-state buffer
- All have same three output states:
  -
  -
  - "High Impedance"

Three-State Buffers

- Four types of three-state buffer

FIGURE 9-12
Four Kinds of Three-State Buffers



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

(a)

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

(b)

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

(c)

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

(d)

B is an "enable" control bit

not enabled

C=A

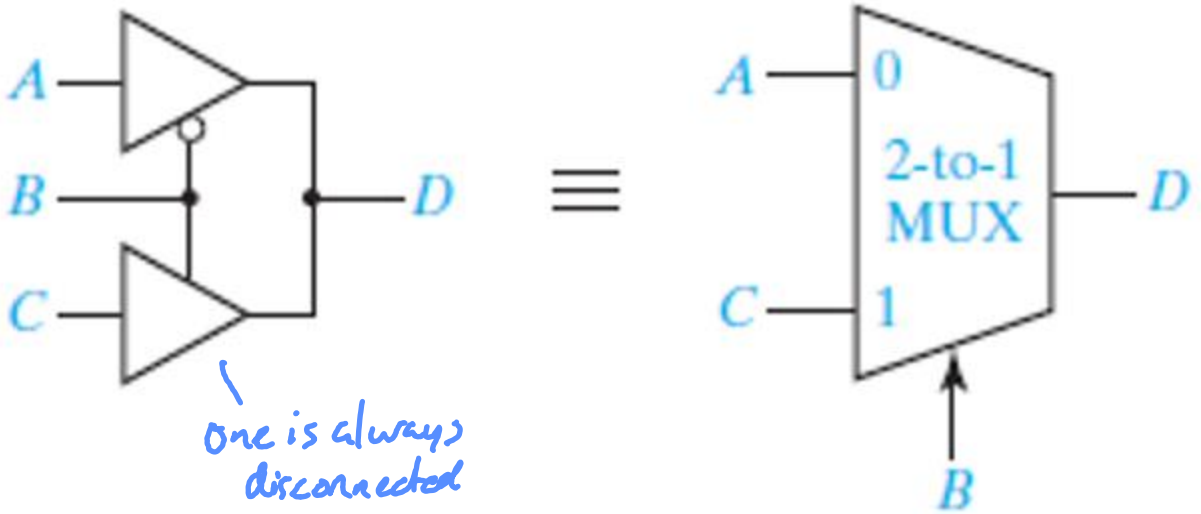
Inverter w/ tri-state output

Non-inverting and inverting tri-state buffers with active-low enable

Three-State Buffers

- Data selection using three-state buffers

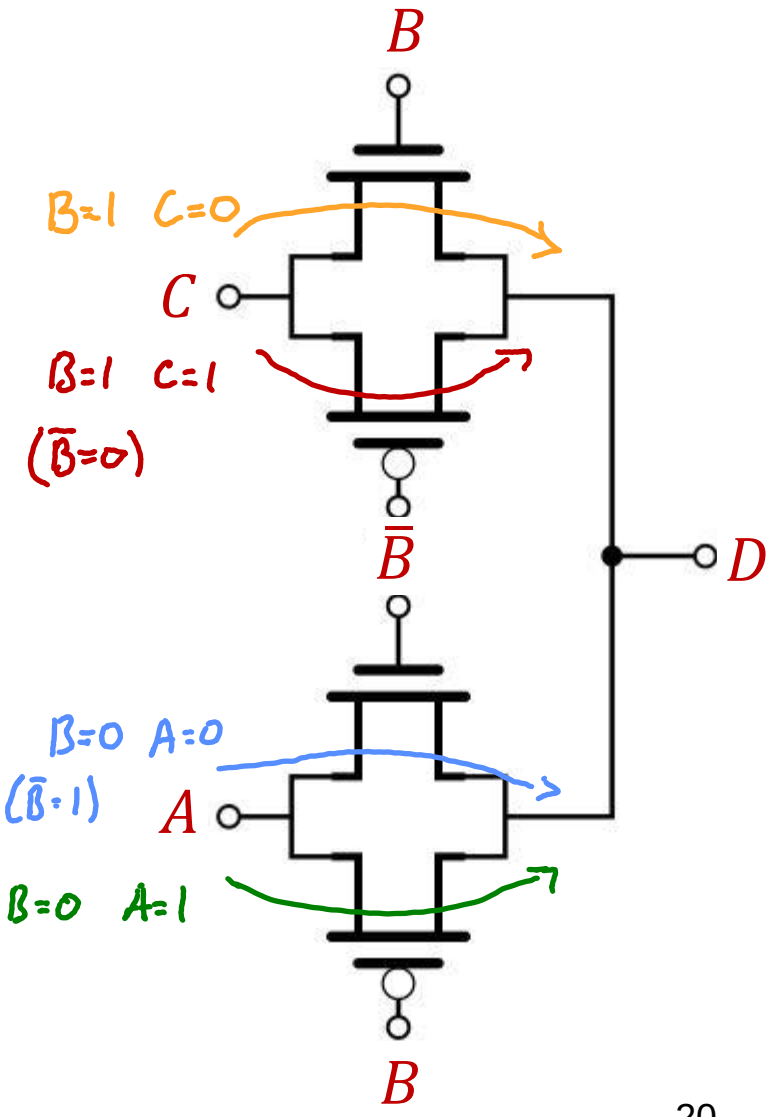
FIGURE 9-13
Data Selection Using Three-State Buffers



- Logically equivalent to 2-to-1 MUX introduced last lecture:
  - $B = 0 \rightarrow D =$ A, lower buffer in high-Z state
  - $B = 1 \rightarrow D =$ C, upper buffer in high-Z state
- Can be implemented with 6 transistors, vs 14 before
  - Inverter (2) + two "pass-transistor" PMOS/NMOS pairs (2 x 2)
  - Inverter (2) + three NAND gates (3 x 4 /gate) = 14

Three-State Buffers

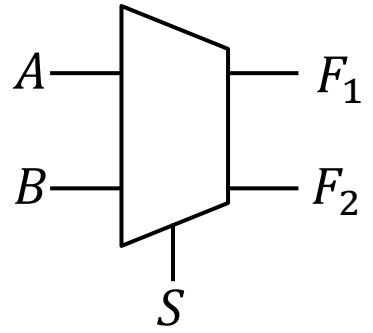
- Data selection using 3-state buffers
  - CMOS multiplexer
  - Note NMOS and PMOS in "pass-transistor" pairs
    - NMOS active-high
    - PMOS active-low
  - Need both to confidently pass signal for both high and low states of A or C
    - NMOS does good job of passing logic 0
    - PMOS does good job of passing logic 1
  - Inverter also needed, from B

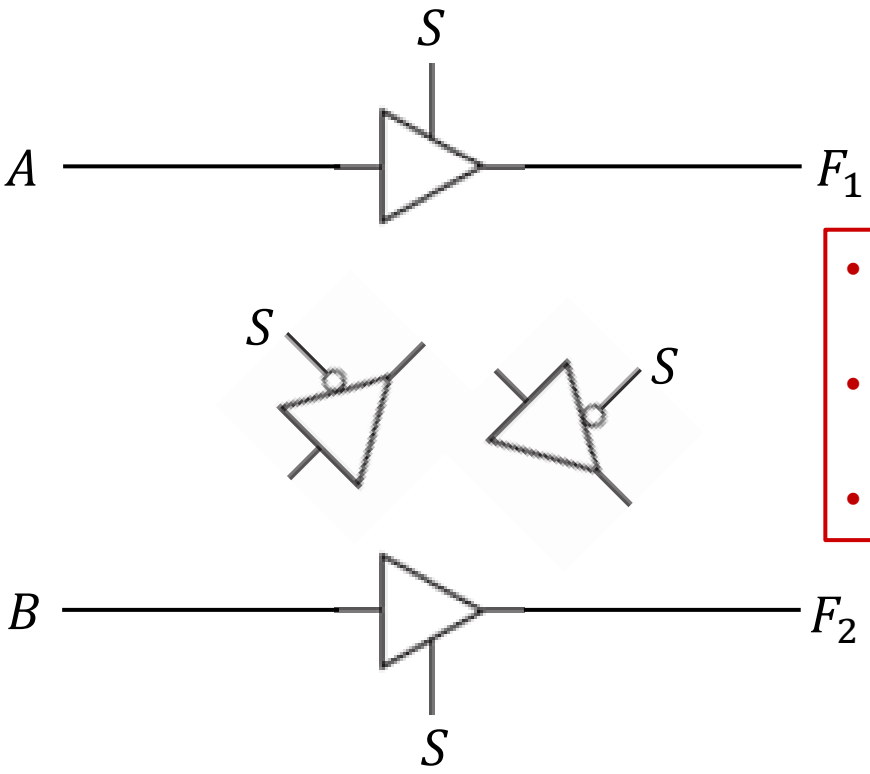


https://www.allaboutcircuits.com/technical-articles/implementing-multiplexers-with-pass-transistor-logic/

Three-State Buffers

- Straight-through/Cross-through MUX



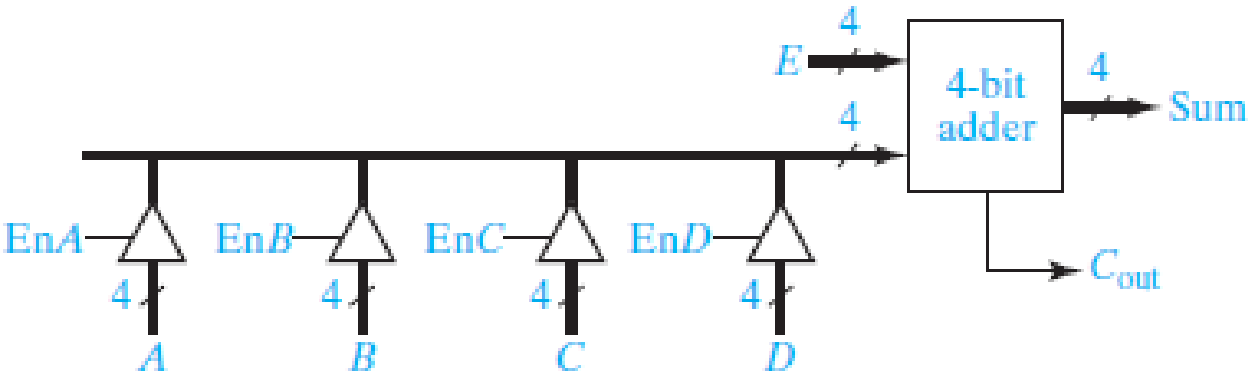


- Two outputs tied together
- But only one active at a time
- Other in high-Z state

Three-State Buffers

- Three-state bus as alternative to MUX data selector
- Example: 4-bit adder with one operand sourced from four different locations

FIGURE 9-15
4-Bit Adder with Four Sources for One Operand



- A bus controller is needed to ensure that only one of *EnA*, *EnB*, *EnC*, and *EnD* are true at any time
- Outputs of the other buffers must be in
- Device known as can be used for bus control