Transistors, CMOS, Logic, and Design

CIS 2400 Recitation 2

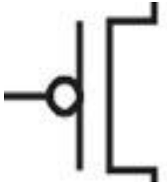
Recitation Outline

- Transistors and CMOS
- Practice, practice, and more practice
- Logic Gates and PLAs

Transistors and CMOS

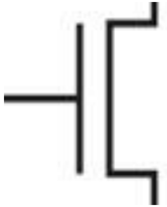
Transistors Review

- MOSFET
- Inputs (two relevant voltages)



pMOS

Current flows when input is 0

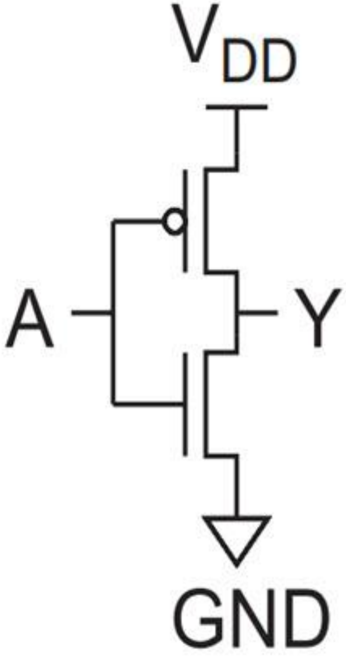


nMOS

Current flows when input is 1

CMOS Review

- CMOS = Complementary MOSFET
- Pull-up network (PUN)
  - Connects to power
  - Use pMOS
- Pull-down network (PDN)
  - Connects to ground
  - Use nMOS



CMOS Review

- Rules
  - Output should be connected to exactly one of 1 or 0
  - Transistors in series are a logical “and”
  - Transistors in parallel are a logical “or”

How to make a CMOS circuit

- Develop a truth table and/or Boolean expression
- Simply logic if you can
- Start with PDN, then get PUN
  - PDN generally easier, but you do you
- Remember that PUN is complement of PDN
  - Can be acquired by changing series → parallel and vice versa
  - This is due to De-Morgan’s law, which we talk about soon
- Test your circuit!

CMOS Practice Time!

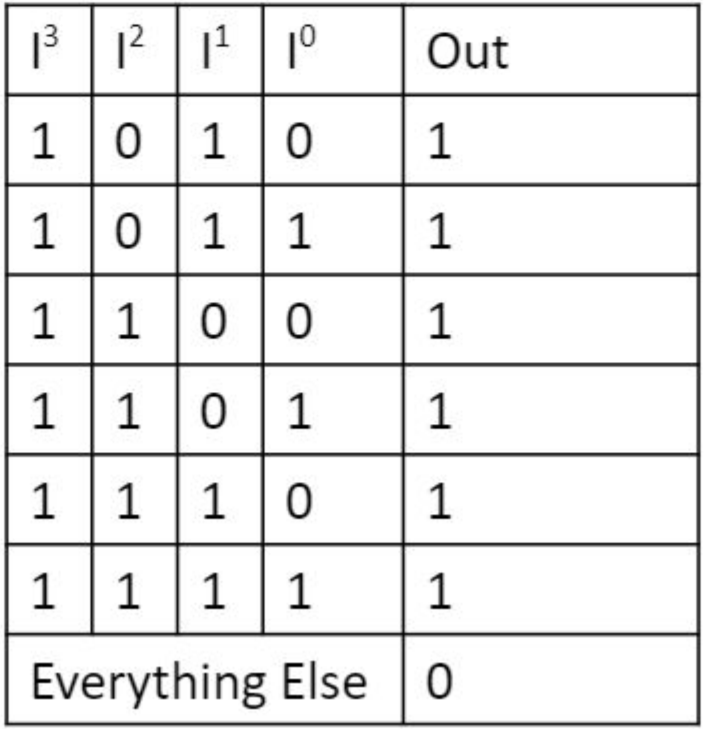
Question 1

- Given 4 bits that represent an unsigned value, design a circuit that outputs ‘1’ if the hexadecimal representation of the 4 bits is a hex letter (A-F), ‘0’ otherwise.

Question 1 - Strategy

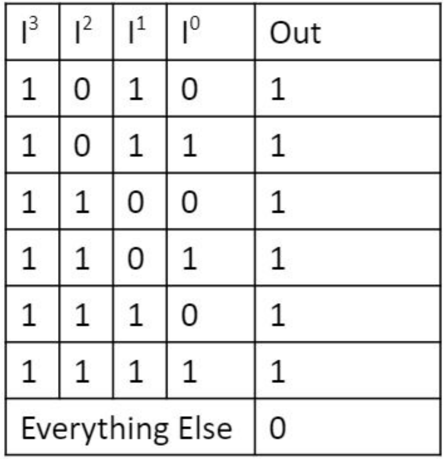
1. Convert requirements to a truth table
2. Extract primitive boolean expression from the truth table
   a. → Which inputs output 1?
3. Simplify the expression
4. Negate the original expression to find PDN expression
5. Design PDN
6. Negate the PDN expression to find PUN expression
7. Design PUN
8. Connect them to output

Step 1: Truth Table



I^3	I^2	I^1	I^0	Out
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1
Everything Else				0

Step 2: Primitive Expression



I^3	I^2	I^1	I^0	Out
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1
Everything Else				0

$$Output = I^3\overline{I^2}I^1\overline{I^0} + I^3\overline{I^2}I^1I^0 + I^3I^2\overline{I^1}\,\overline{I^0} + I^3I^2\overline{I^1}I^0 + I^3I^2I^1\overline{I^0} + I^3I^2I^1I^0$$

Step 3: Simplify Expression

$$Output = I^3\overline{I^2}I^1\overline{I^0} + I^3\overline{I^2}I^1I^0 + I^3I^2\overline{I^1}\,\overline{I^0} + I^3I^2\overline{I^1}I^0 + I^3I^2I^1\overline{I^0} + I^3I^2I^1I^0$$

$I^3\overline{I^2}I^1(\overline{I^0} + I^0) + I^3I^2\overline{I^1}(\overline{I^0} + I^0) + I^3I^2I^1(\overline{I^0} + I^0)$ - Distributive

$I^3\overline{I^2}I^1 + I^3I^2\overline{I^1} + I^3I^2I^1$ - Identity

$I^3(\overline{I^2}I^1 + I^2\overline{I^1} + I^2I^1)$ - Distributive

= $I^3(I^2 + I^1)$

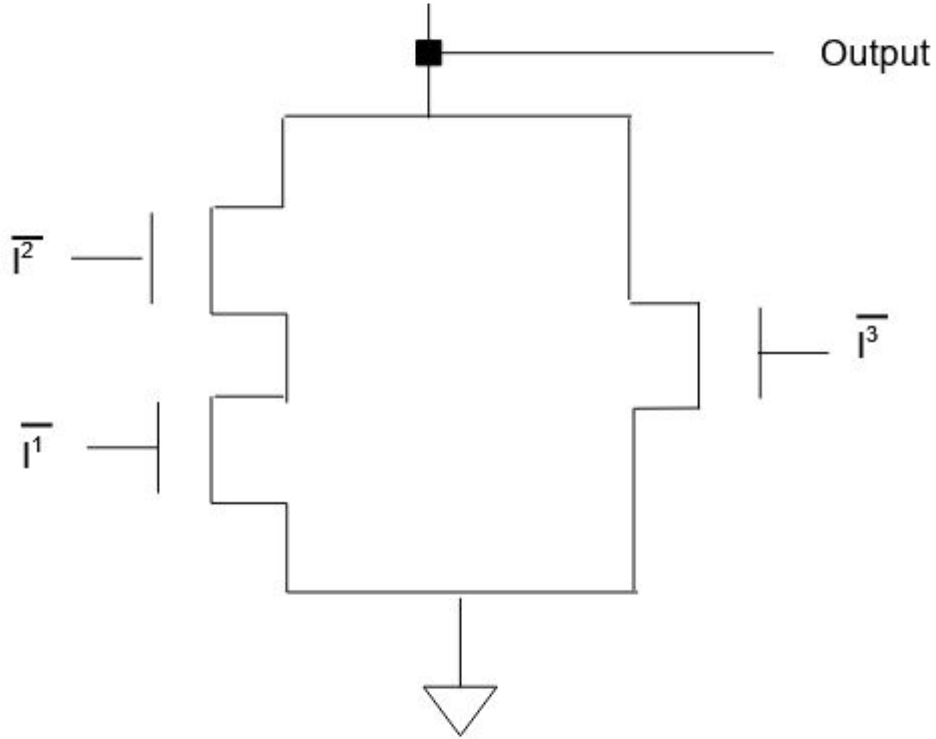
Step 4: Negate Expression for PDN

$$\overline{Output} = \overline{I^3(I^2 + I^1)}$$

$$\overline{I^3} + \overline{(I^2 + I^1)}$$

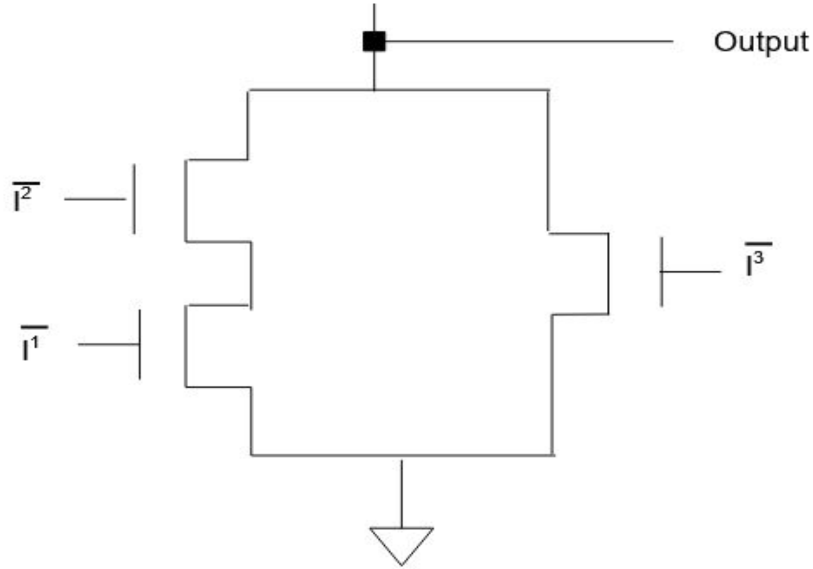
$$\overline{I^3} + (\overline{I^2}\,\overline{I^1})$$

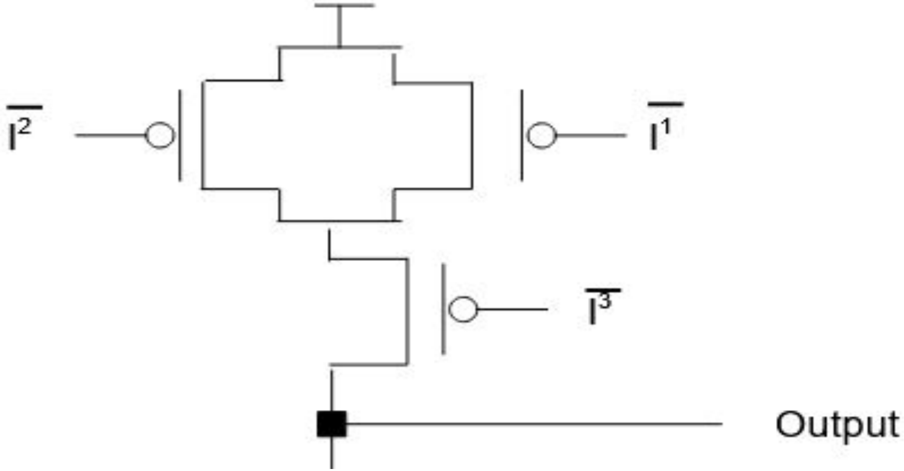
Step 5: Design PDN



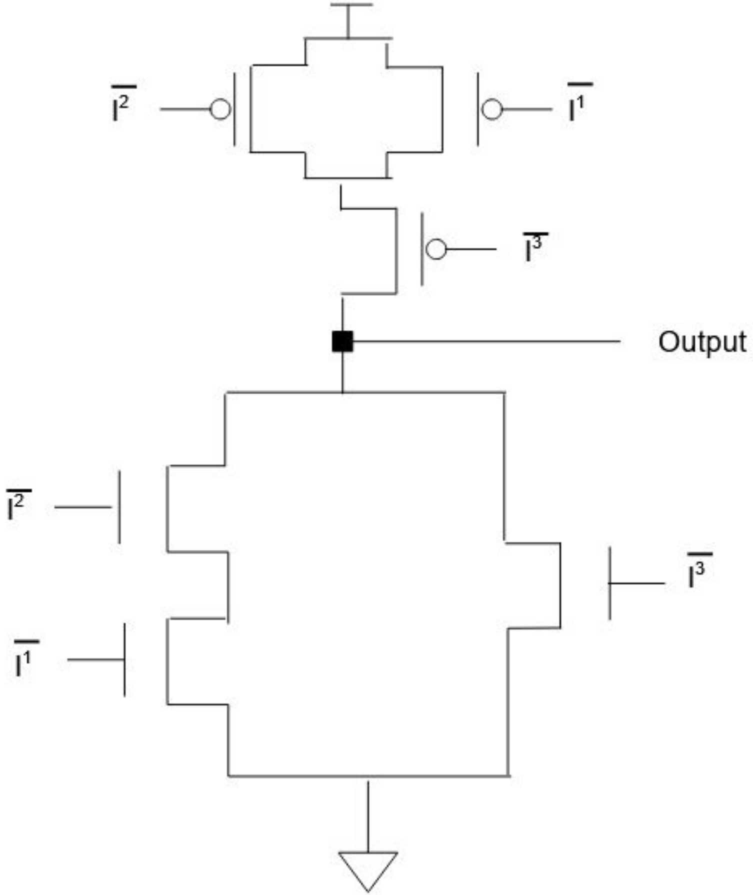
Step 7: Design PUN (skipping step 6)

- From PDN, we can get PUN by converting nMOS to pMOS, swapping parallel with series and series with parallel



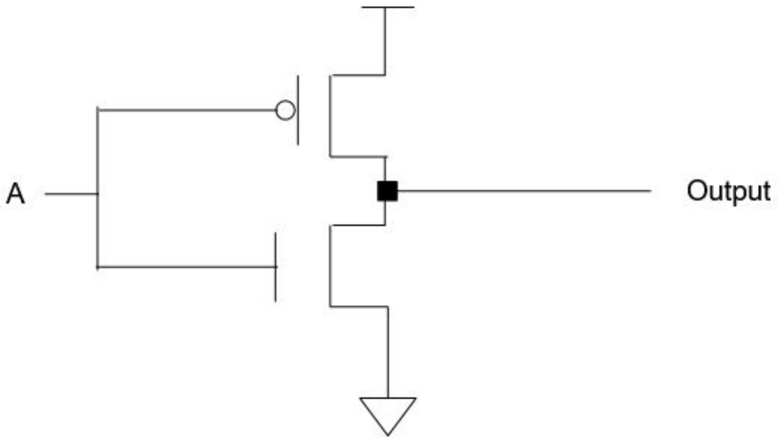


Step 8: Combine



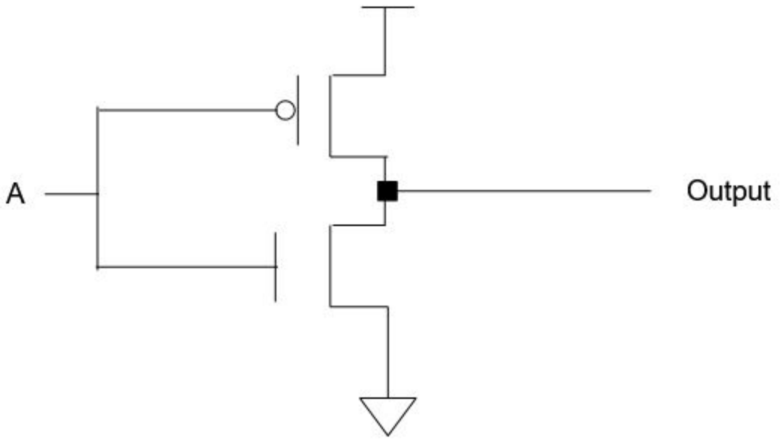
CMOS Examples #1

- Is this legal, if so, what’s the truth table? If not, why?



CMOS Examples #1

- Is this legal, if so, what's the truth table? If not, why?



Legal!

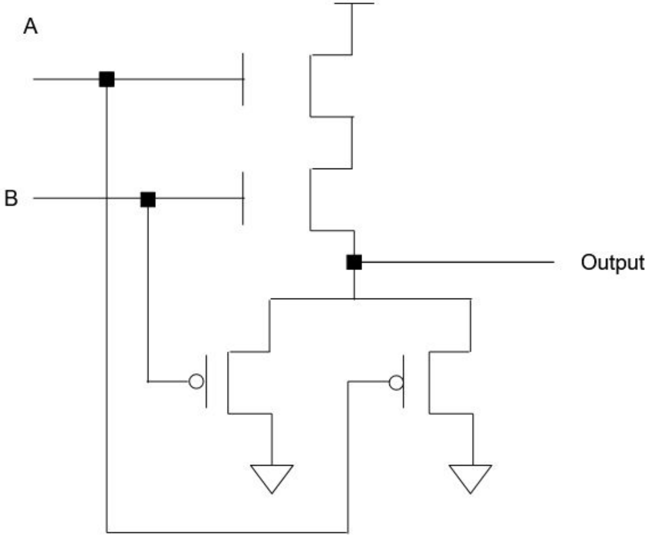
A	Output
0	1
1	0

output = ~A

This is the "not gate"

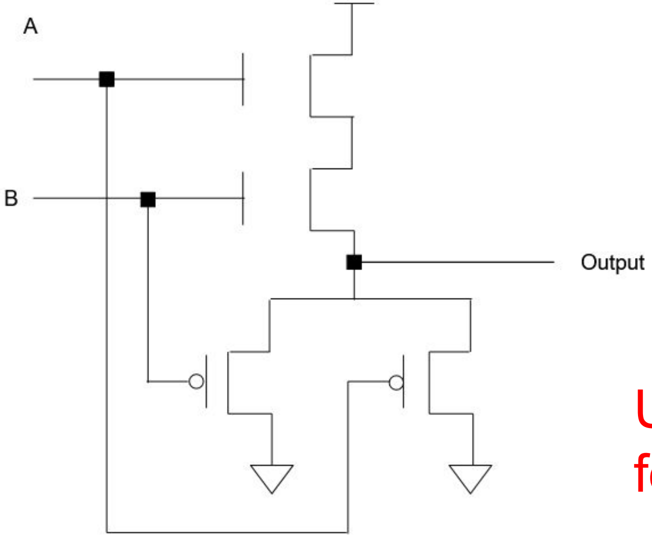
CMOS Examples #2

- Is this legal, if so, what’s the truth table? If not, why?



CMOS Examples #2

- Is this legal, if so, what's the truth table? If not, why?



Uses nMOS for PUN

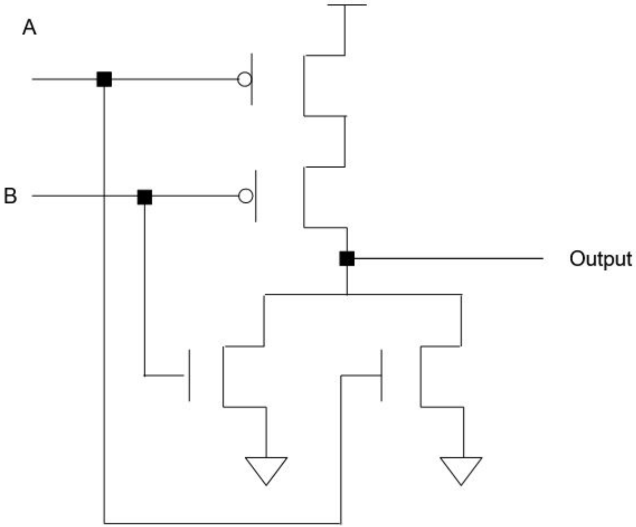
Illegal

Why?

Uses pMOS for PDN

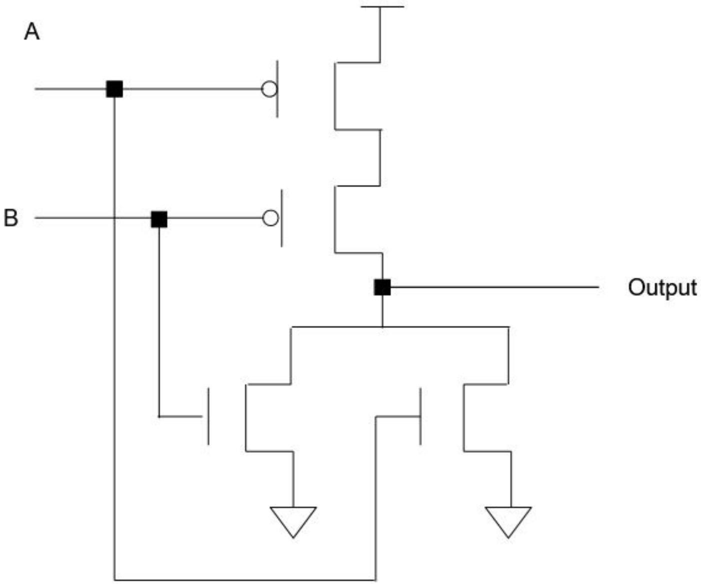
CMOS Examples #3

- Is this legal, if so, what's the truth table? If not, why?



CMOS Examples #3

- Is this legal, if so, what's the truth table? If not, why?



Legal!

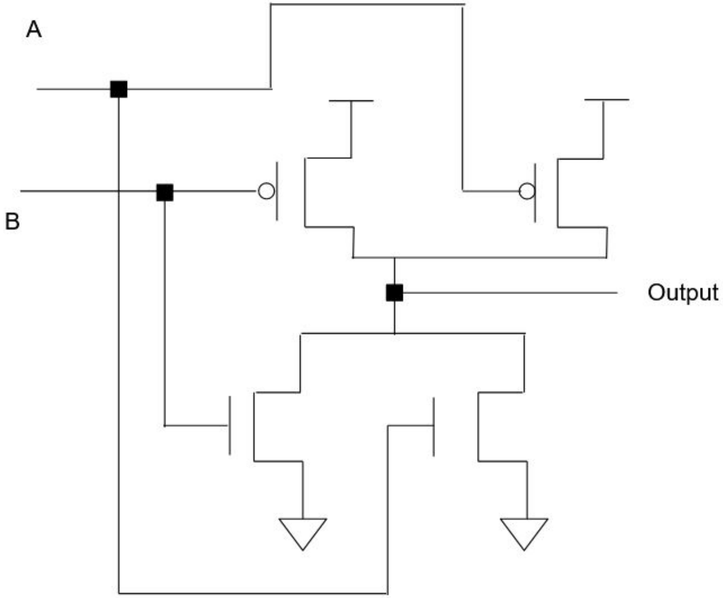
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

output = ~A & ~B
= ~(A | B)

This is the "nor gate"

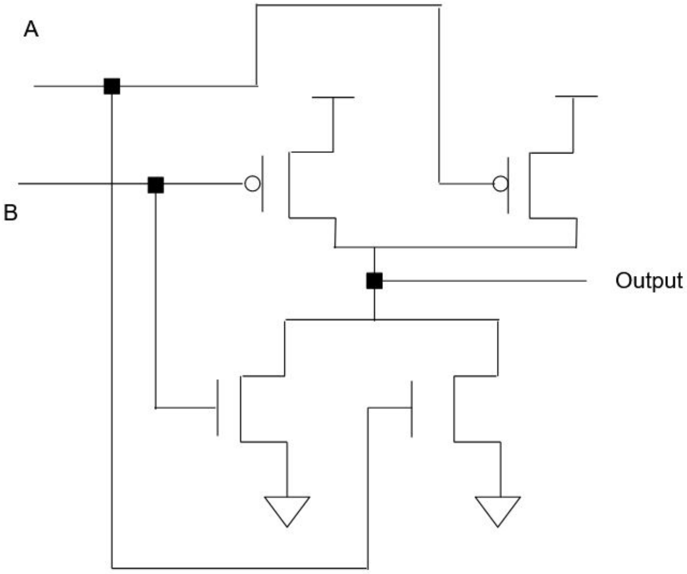
CMOS Truth Tables #4

- Is this legal, if so, what’s the truth table? If not, why?



CMOS Truth Tables #4

- Is this legal, if so, what's the truth table? If not, why?



Illegal

Why?

Consider inputs
a = 0
b = 1

Ground and power both connected to output!!

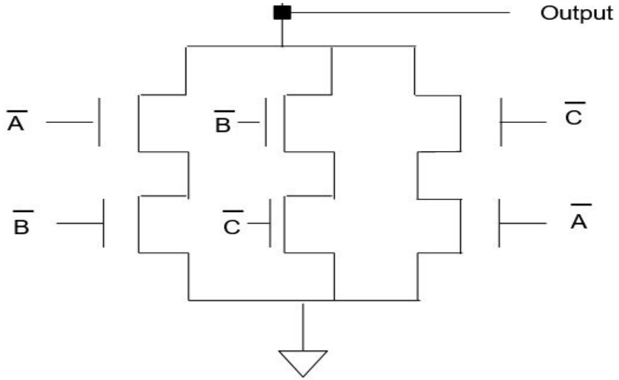
More Complex CMOS Design Practice!!!

- Design a CMOS circuit with 3 inputs, A, B and C, and produces a high output if and only if 2 or more of the inputs are high
  - You can assume that you also have access to the inverses of A B and C (i.e. ~A, ~B, ~C) and that you can use those as inputs to your circuit

More Complex CMOS Design Practice!!!

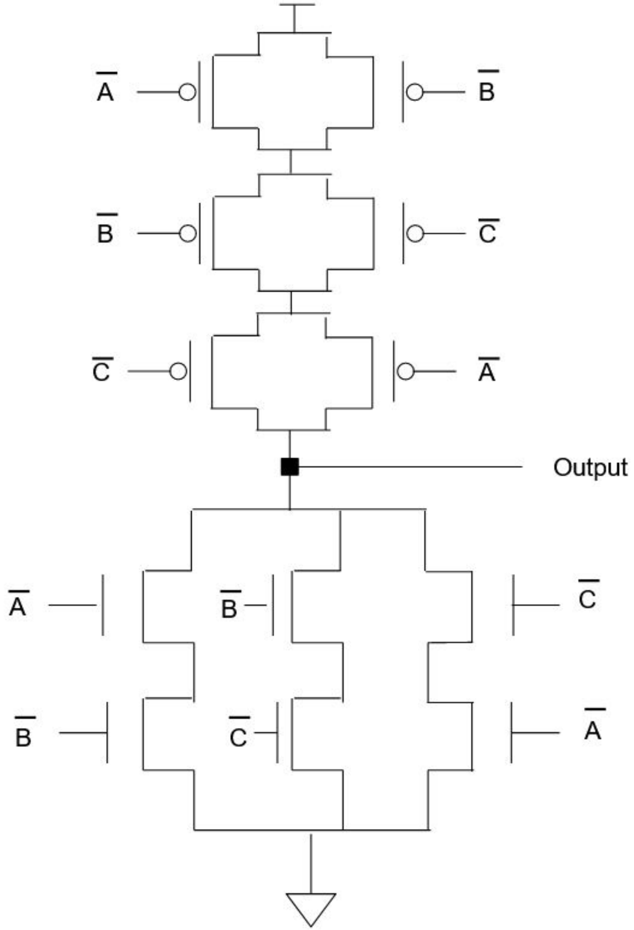
- Design a CMOS circuit with 3 inputs, A, B and C, and produces a high output if and only if 2 or more of the inputs are high
  - You can assume that you also have access to the inverses of A B and C (i.e. ~A, ~B, and ~C) and that you can use those as inputs to your circuit

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	1	1
0	1	0	0
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1



(AB) + (BC) + (CA) = output

Can get the PUN from PDN by inverting the relationships. Series becomes parallel. Parallel becomes series



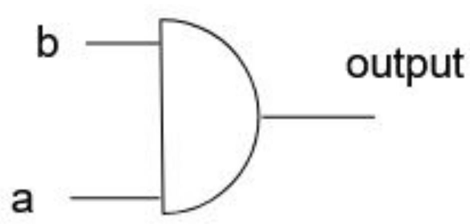
Logic Gates and PLA

Gates

- Given some input signals (either 0’s or 1’s), provides some output

- Example:

AND gate



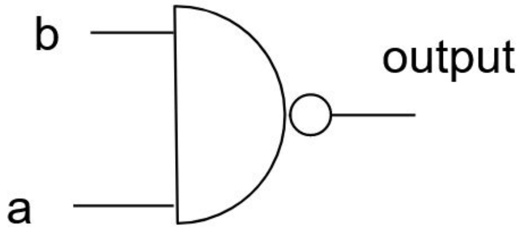
A	B	Output
0	0	0
0	1	0
1	0	0
1	1	1

Gates

- Given some input signals (either 0’s or 1’s), provides some output

- Example:

NAND gate



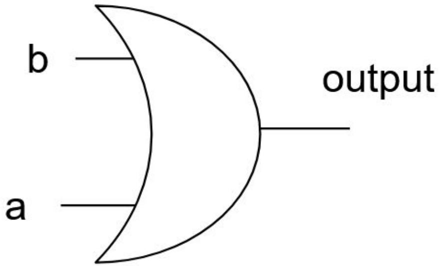
A	B	Output
0	0	1
0	1	1
1	0	1
1	1	0

Gates

- Given some input signals (either 0's or 1's), provides some output

- Example:

OR gate



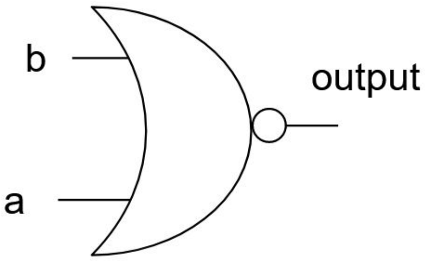
A	B	Output
0	0	0
0	1	1
1	0	1
1	1	1

Gates

- Given some input signals (either 0’s or 1’s), provides some output

- Example:

NOR gate



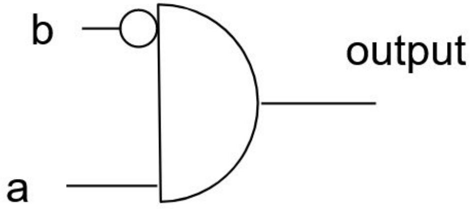
A	B	Output
0	0	1
0	1	0
1	0	0
1	1	0

Gates

- Given some input signals (either 0's or 1's), provides some output

- Example:

NOT's can be applied to inputs too

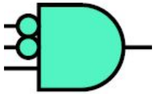


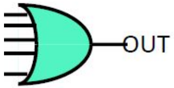
A	B	Output
0	0	0
0	1	0
1	0	1
1	1	0

Any Questions? Feedback? Thoughts? Feelings?

PLA’s

- Can implement ANY truth table
  - Sometimes this may be very big depending on number of inputs
- Steps:
  - Write truth table
    - Start with the ‘0’ bit pattern and increment so that you don’t forget any combinations! (it is also easier to read)
  - AND gate any combinations that yield a “1” in the truth table
    - Can invert an input before AND-ing a combination
  - OR the results of the AND gates





PRAXIS: Given 4 bits that represent an unsigned integer, output “1” if the integer is prime. Construct a truth table first, then create a PLA

(Prime numbers are > 1 and don’t have any factors other than itself and 1).

PLA Praxis

- Given 4 bits that represent an **unsigned** integer, output “1” if the integer is prime. Construct a truth table first, then create a PLA
    - ○ (Prime numbers are > 1 and don’t have any factors other than itself and 1).

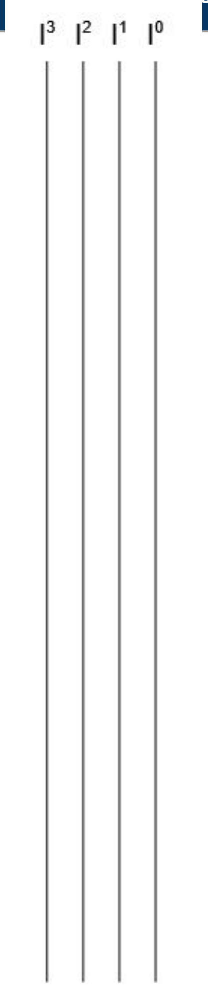
Dec	I^3	I^2	I^1	I^0	Out
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

PLA Praxis

Dec
2
3
5
7
11
13

I^3	I^2	I^1	I^0	Out
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1

Make the PLA!

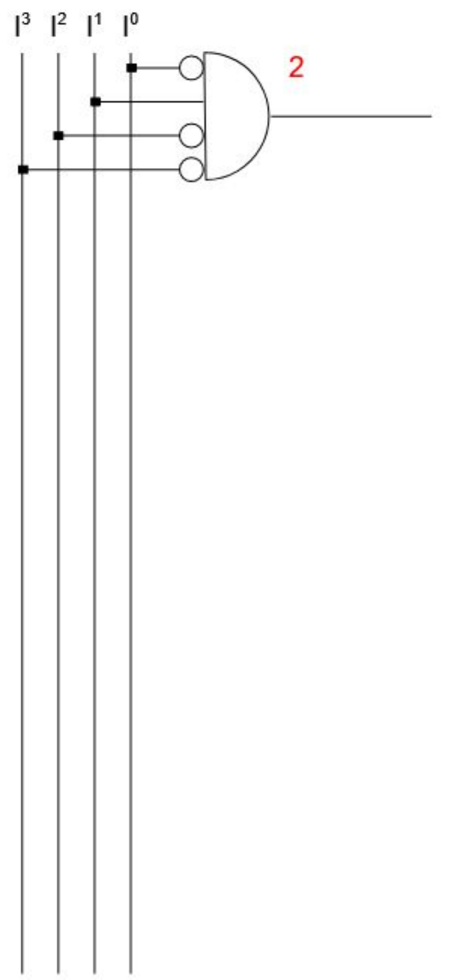


Start with inputs

PLA Praxis

Dec
2
3
5
7
11
13

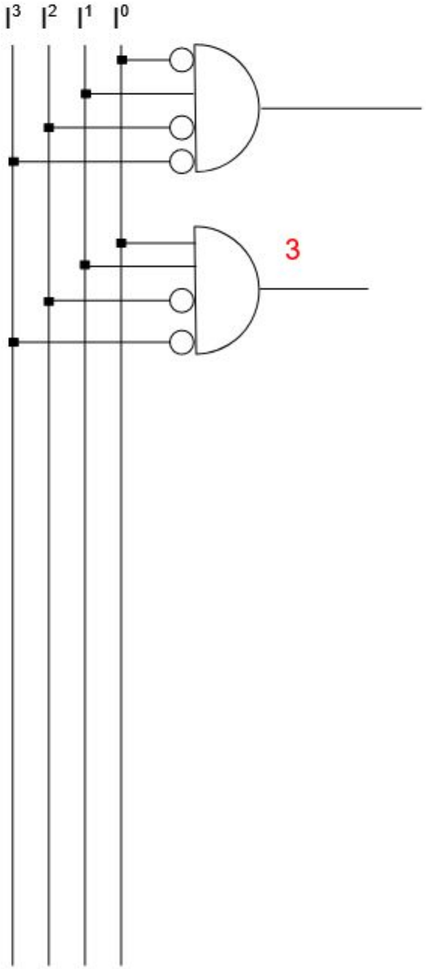
I^3	I^2	I^1	I^0	Out
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1



PLA Praxis

Dec
2
3
5
7
11
13

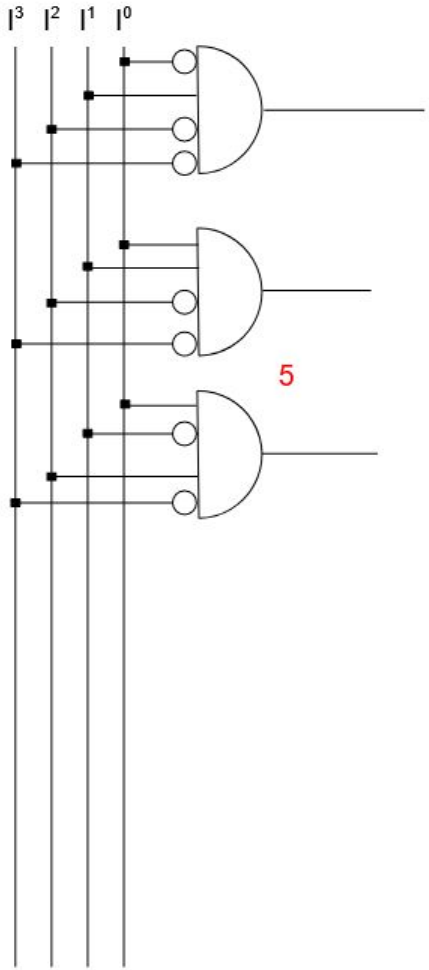
I^3	I^2	I^1	I^0	Out
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1



PLA Praxis

Dec
2
3
5
7
11
13

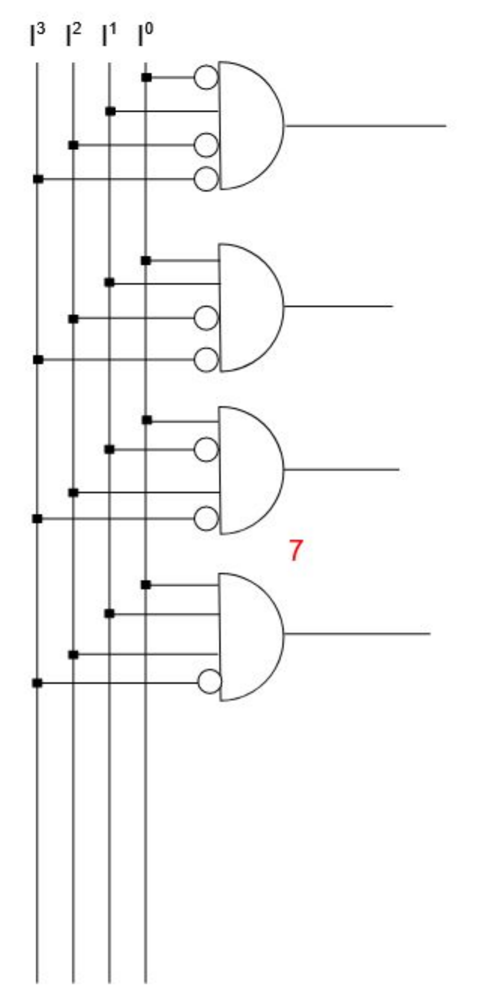
I^3	I^2	I^1	I^0	Out
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1



PLA Praxis

Dec
2
3
5
7
11
13

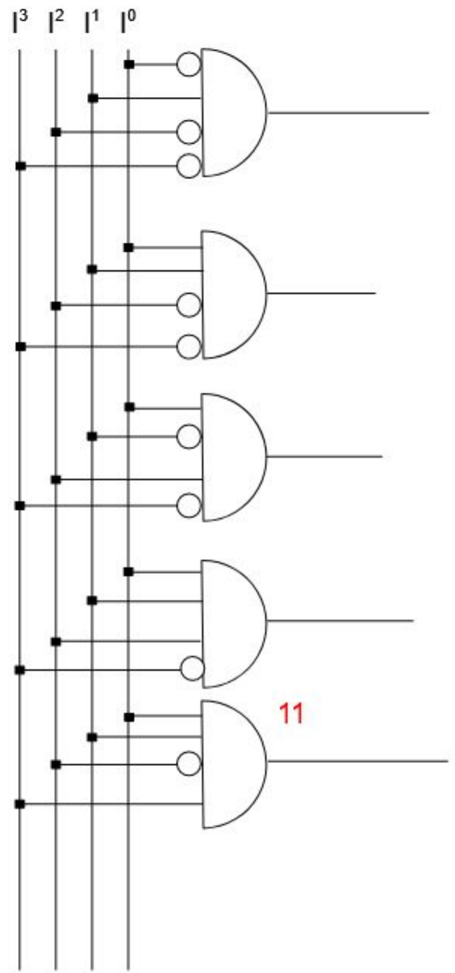
I^3	I^2	I^1	I^0	Out
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1



PLA Praxis

Dec
2
3
5
7
11
13

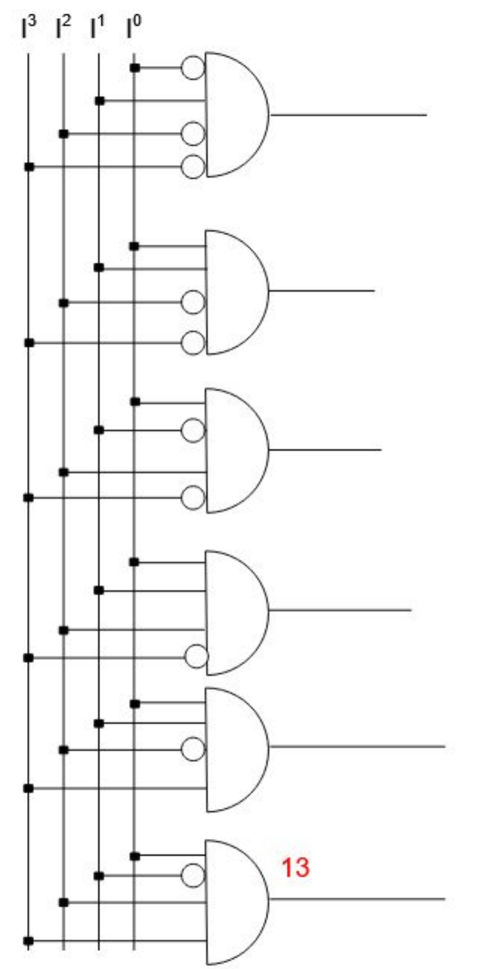
I^3	I^2	I^1	I^0	Out
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1

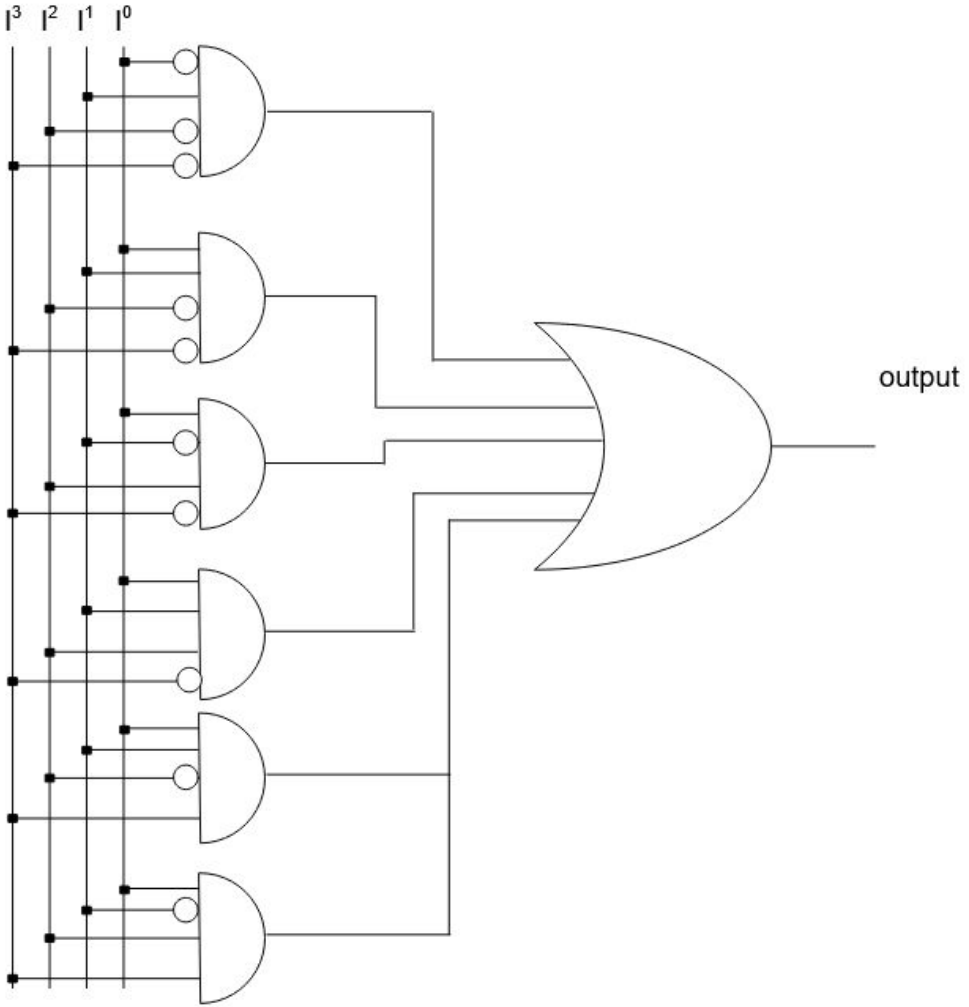


PLA Praxis

Dec
2
3
5
7
11
13

I^3	I^2	I^1	I^0	Out
0	0	1	0	1
0	0	1	1	1
0	1	0	1	1
0	1	1	1	1
1	0	1	1	1
1	1	0	1	1




I^3
I^2
I^1
I^0
output

That’s all we have for today!

Reminders:

- TA-lead recitations will take place on
  - Tuesdays 6:30-8:00pm in Moore 100A
  - Wednesday 12:00-1:30pm in Moore 100C
- Check the course website for OH times
- HW2 is due this Friday 9/30