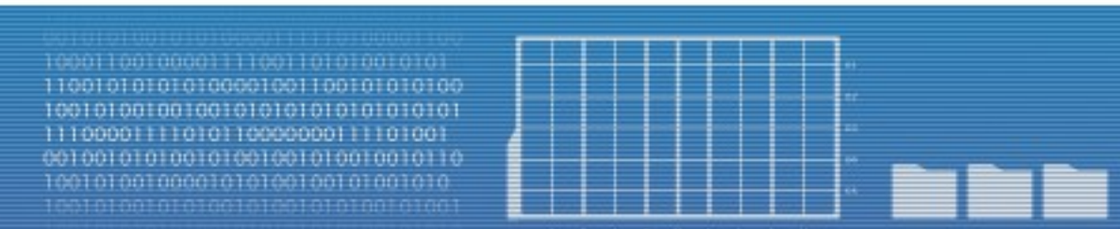
DEPARTMENT OF
INFORMATION SYSTEMS
AND COMPUTER SCIENCE

CMOS Logic Design

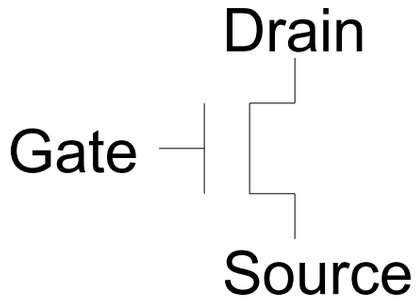
Analog Transistors for Digital Gates

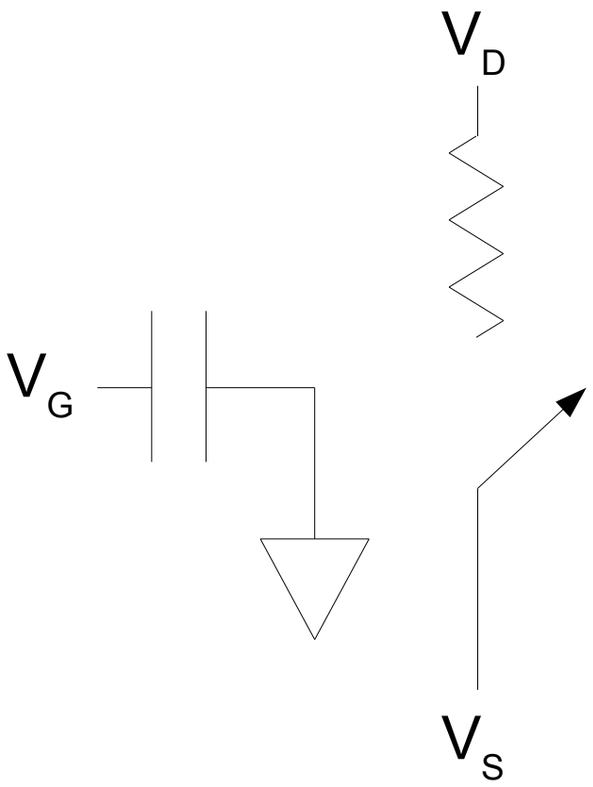
Lecture Time!

- Transistors: ***N***o ***P***roblem
- CMOS: Teamwork in Basic Electronics
- CMOS Gates: From Paper to Circuits

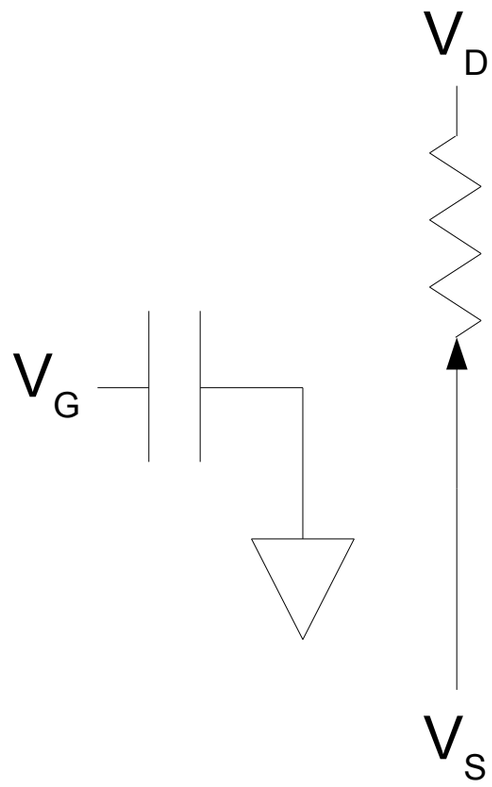


The NFET (N-channel Field Effect Transistor)





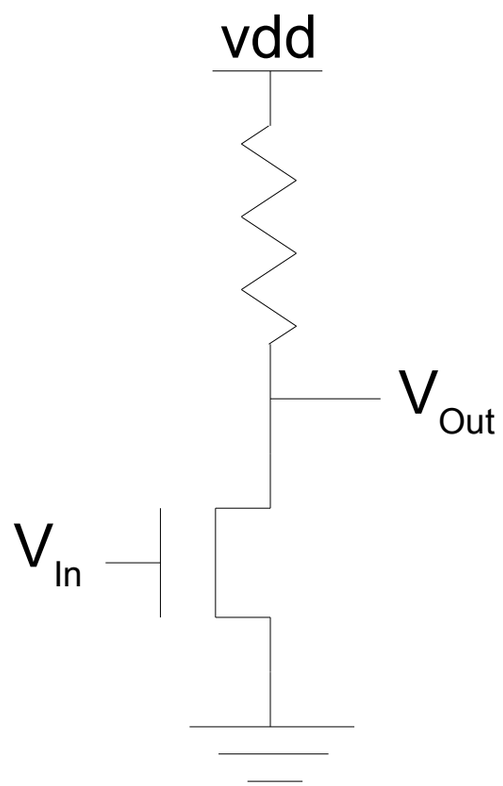
Switch OFF if V_G is LOW / 0



Switch ON if V_G is HIGH / 1

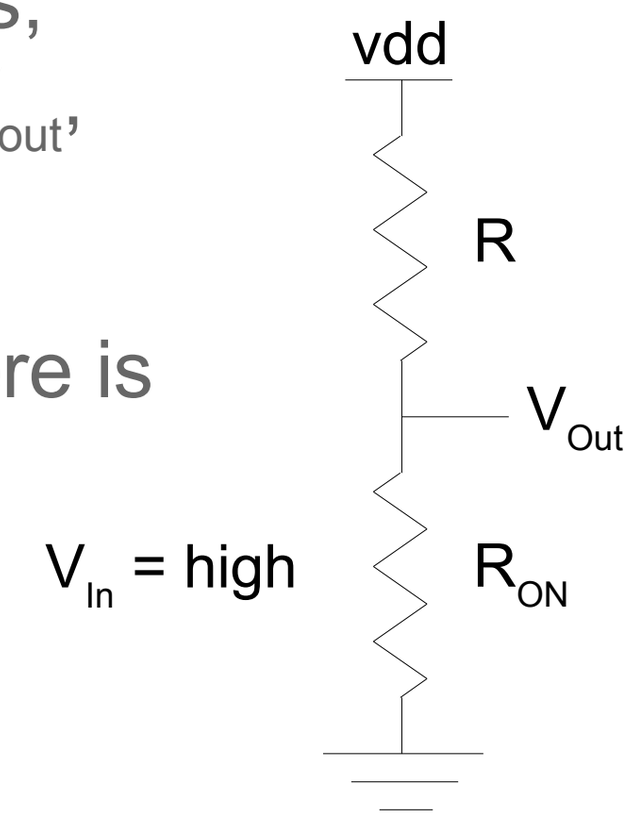
Proposed NMOS Inverter

- N-channel Metal Oxide Semiconductor (see Lab 1)
  - When V_{In} is low, resistor R pulls V_{Out} up to VDD.
  - What if V_{In} is high?

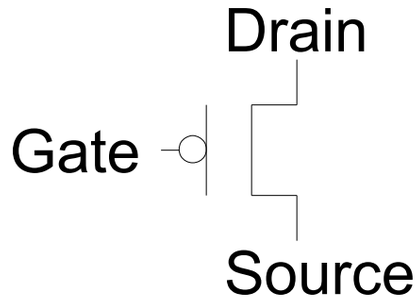


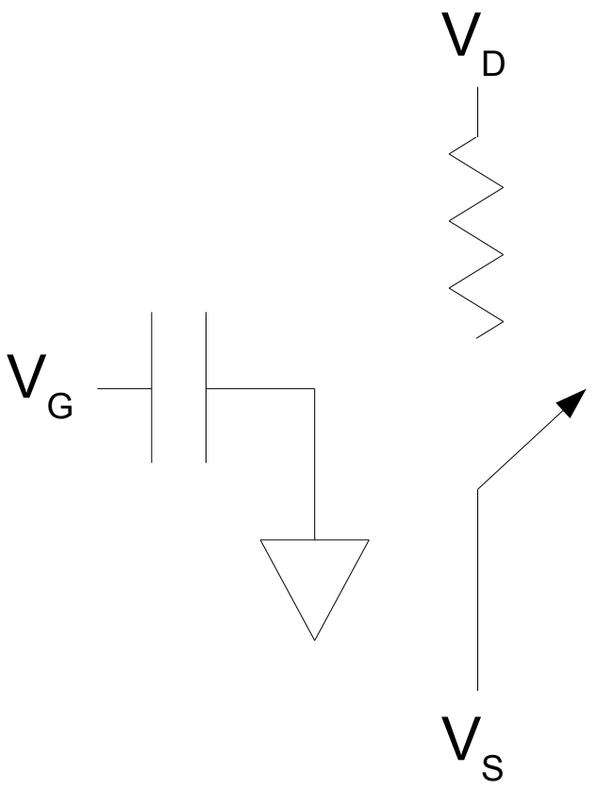
Problems with NMOS Inverter

- When V_{In} is high, and if R_{ON} is significantly less than R, V_{Out} close to but not quite zero! (again, see Lab 1)
- Large R causes other complications, such as slow rising transition at V_{out}, so increasing R is not advised.
- We need another switch so that there is no direct connection between VDD and ground!
  - (except maybe during unavoidable transitions)

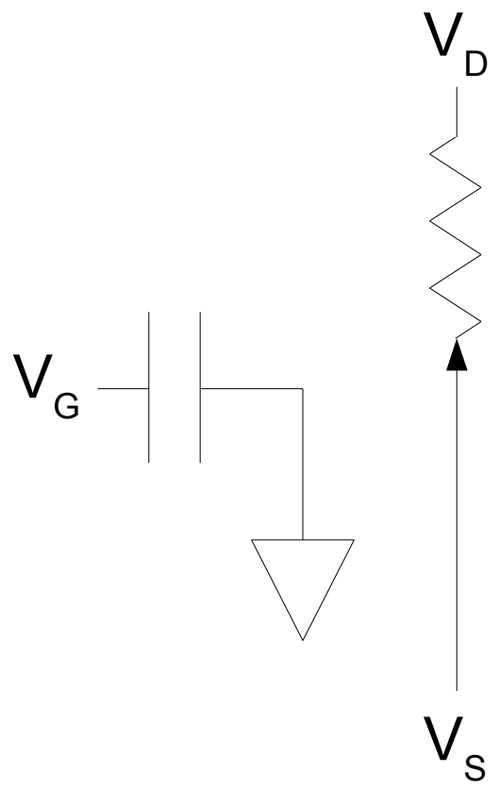


The PFET (P-channel Field Effect Transistor)





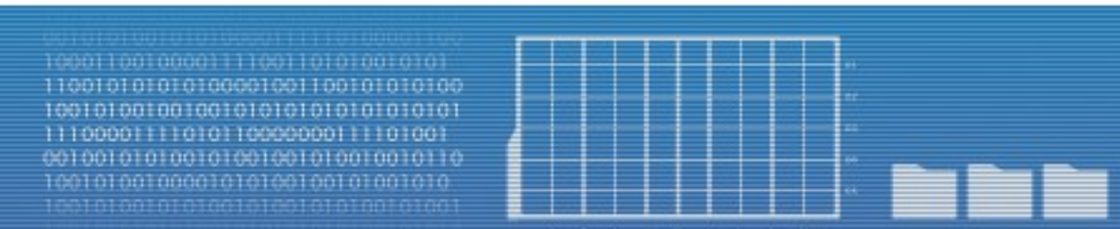
Switch OFF if V_G is HIGH / 1



Switch ON if V_G is LOW / 0

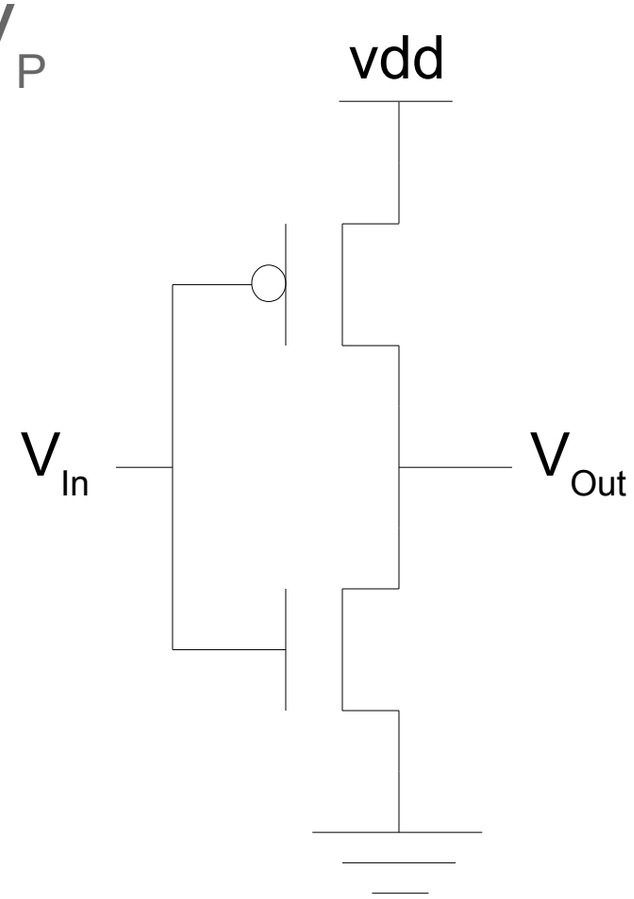
Confused? Can't Memorize?

- Note: Some electronics books say switch is 'closed' if on or connected, 'open' if off or disconnected.
- Can't remember the difference between NFETs and PFETs?
- 0P1N = OPIN = a mispronounced 'open'
  - 0P = PFET is ON if input voltage is signal 0
    - low, ground, etc.
  - 1N = NFET is ON if input voltage is signal 1
    - high, vdd, etc.



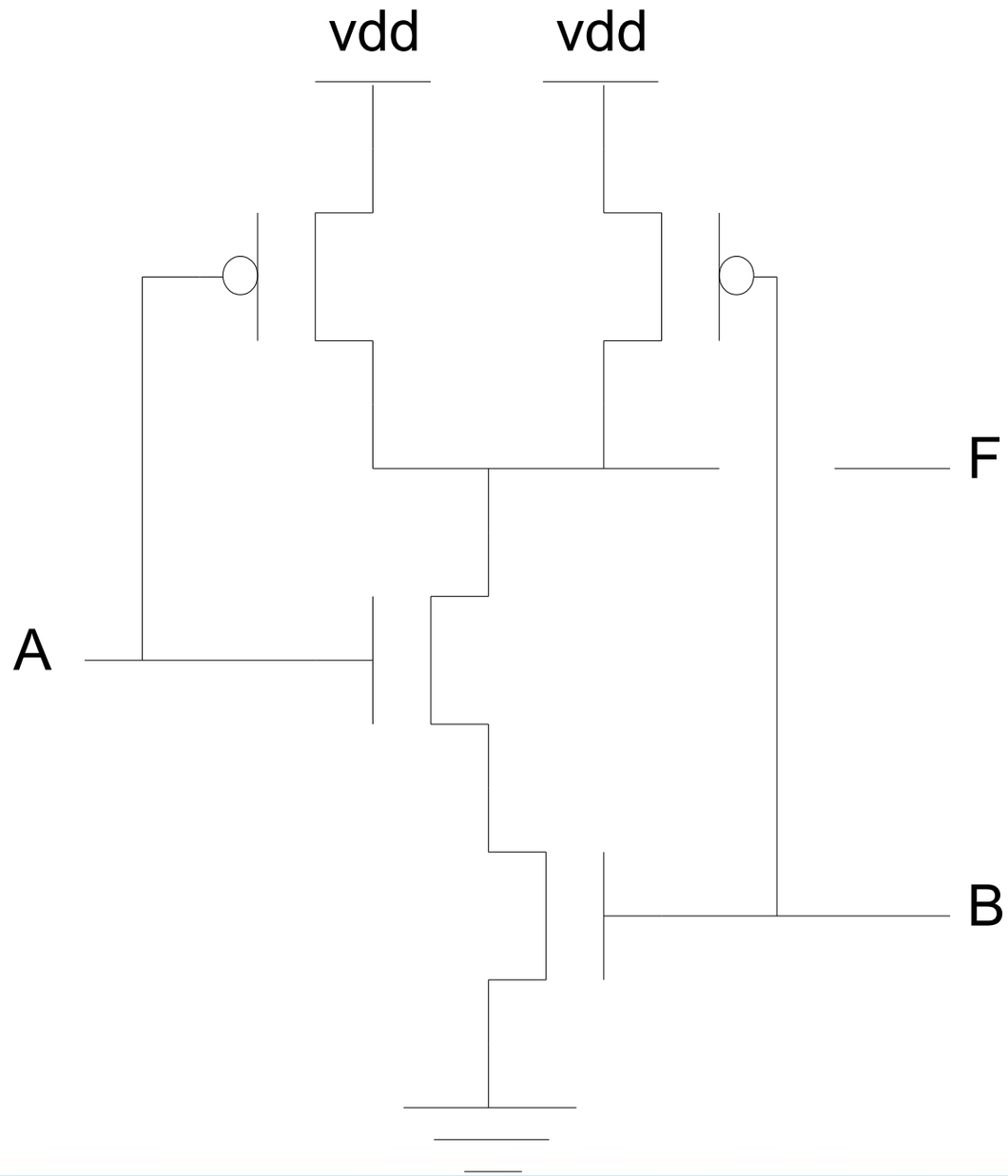
Proposed CMOS Inverter

- "Complementary" MOS (see Lab 2)
- V_{OL} = 0V, V_{OH} = VDD, $V_{IL} = V_N$, $V_{IH} = V_P$
  - V_N and V_P are the voltage levels which turn on the transistors.
  - Let's try drawing the VTC of this device!



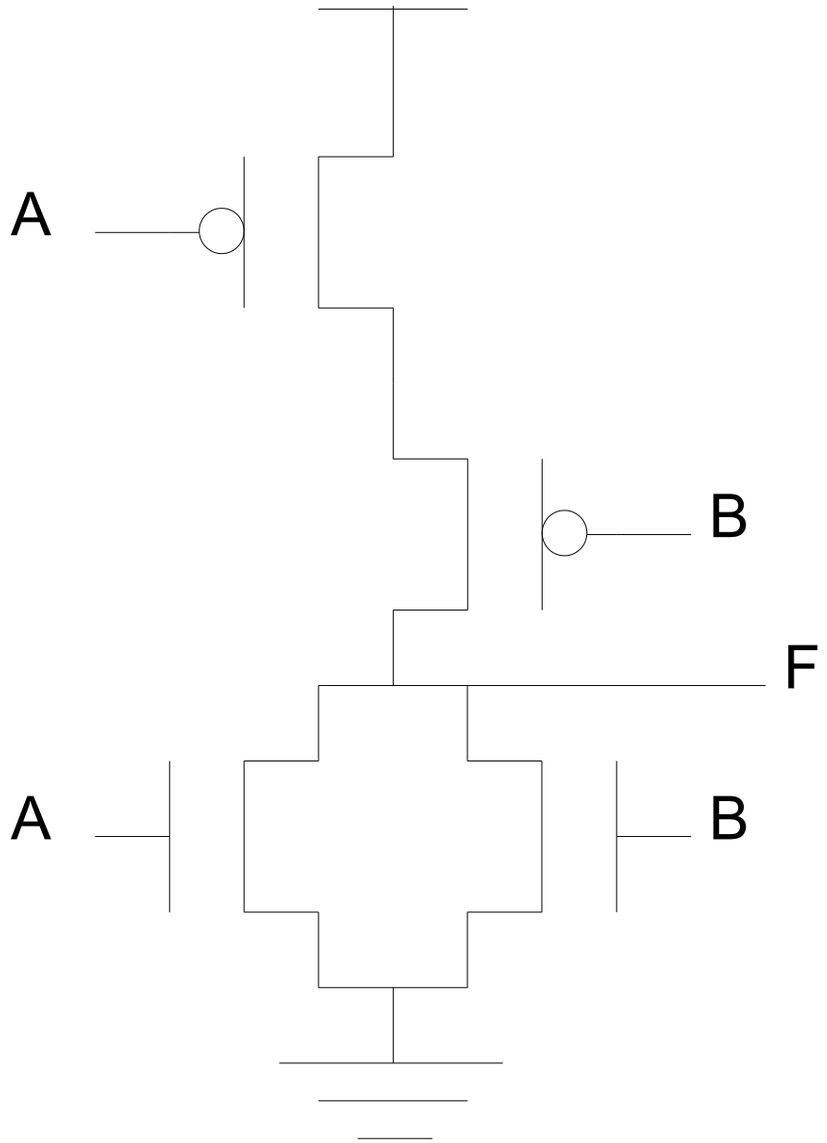
2-Input Logic Functions

- What function does this circuit compute?

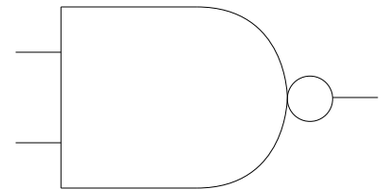


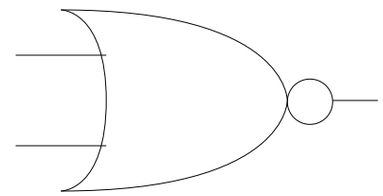
2-Input Logic Functions

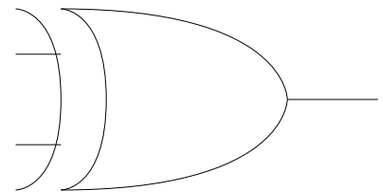
- What about this one?



- Are you familiar with these?







Advantages of CMOS over NMOS

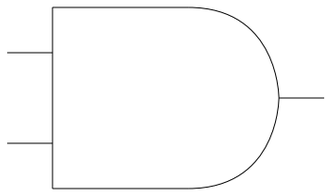
- Much Better Output Voltages
  - No more voltage divider.
  - Output connected to either vdd or ground but not both.
- Faster
  - No more need for big pullup R.
  - Can have smaller RC time constants.
- Much less power consumption
  - No current between vdd and ground except during transition times.
  - If values are not changing, very little power consumption.
    - CMOS memory is used for configuration settings in desktops since it can be powered by small batteries.

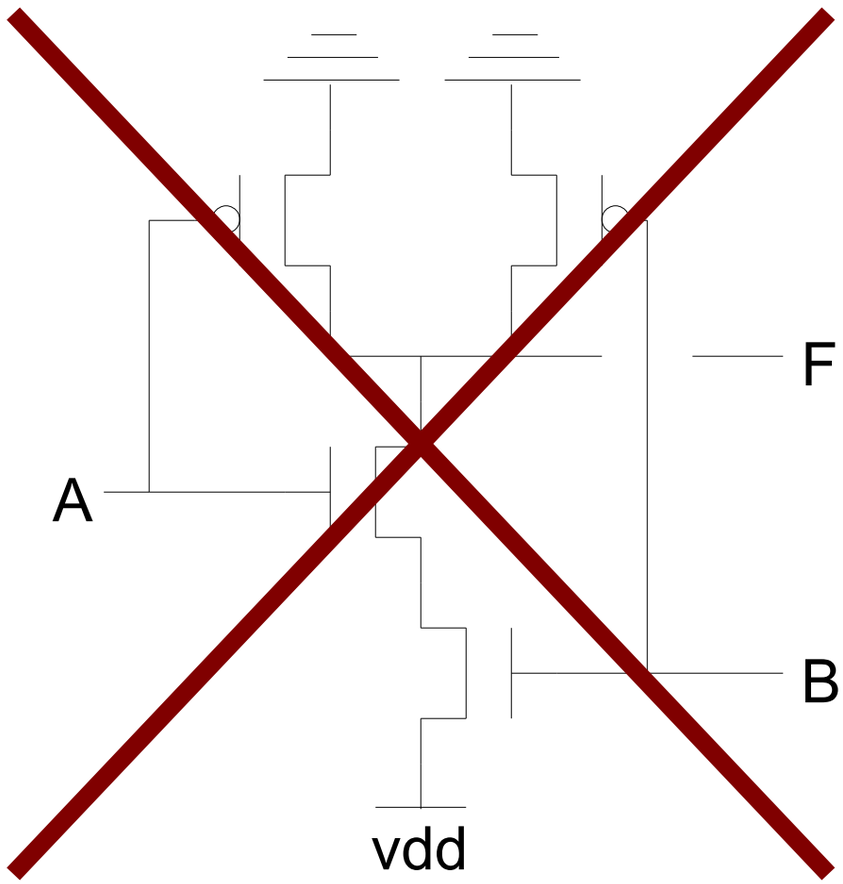
Transistor Limitations

- These transistors are not perfect and have physical limitations.
- Theoretically, NFETs and PFETs are just switches, but in reality:
  - NFETs - good for 0's, bad for 1's
  - PFETs - good for 1's, bad for 0's
    - But we're CS, not CE or PS, so we're not going to discuss the reason. We just need to know that they work like that, so that we can use them to make our RELIABLE circuits.
- That's why CMOS always has NFETs connected to ground, and PFETs to power/VDD.

Transistor Limitations

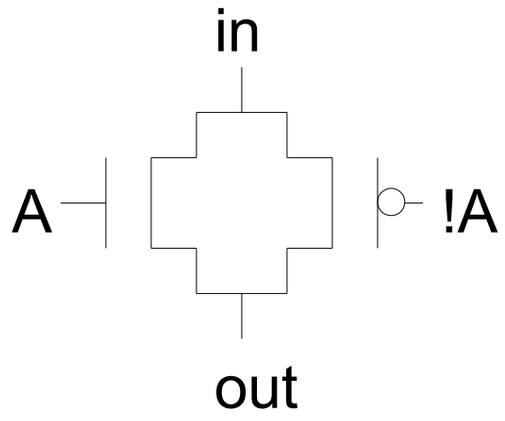
- In other words, you can't get AND by swapping the vdd and ground nodes in a NAND circuit.
  - So what should we do?
  - What if !A and !B (not-A and not-B) were available?
- And at no extra transistor cost!





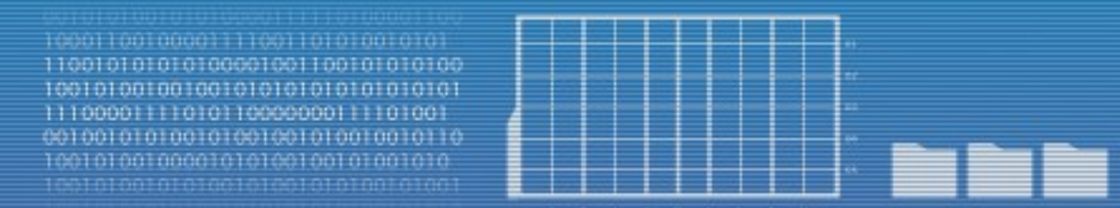
Transistor Limitations

- Transmission Gate
  - Basically a switch
  - Needs both an NFET and a PFET
  - Needs A and !A



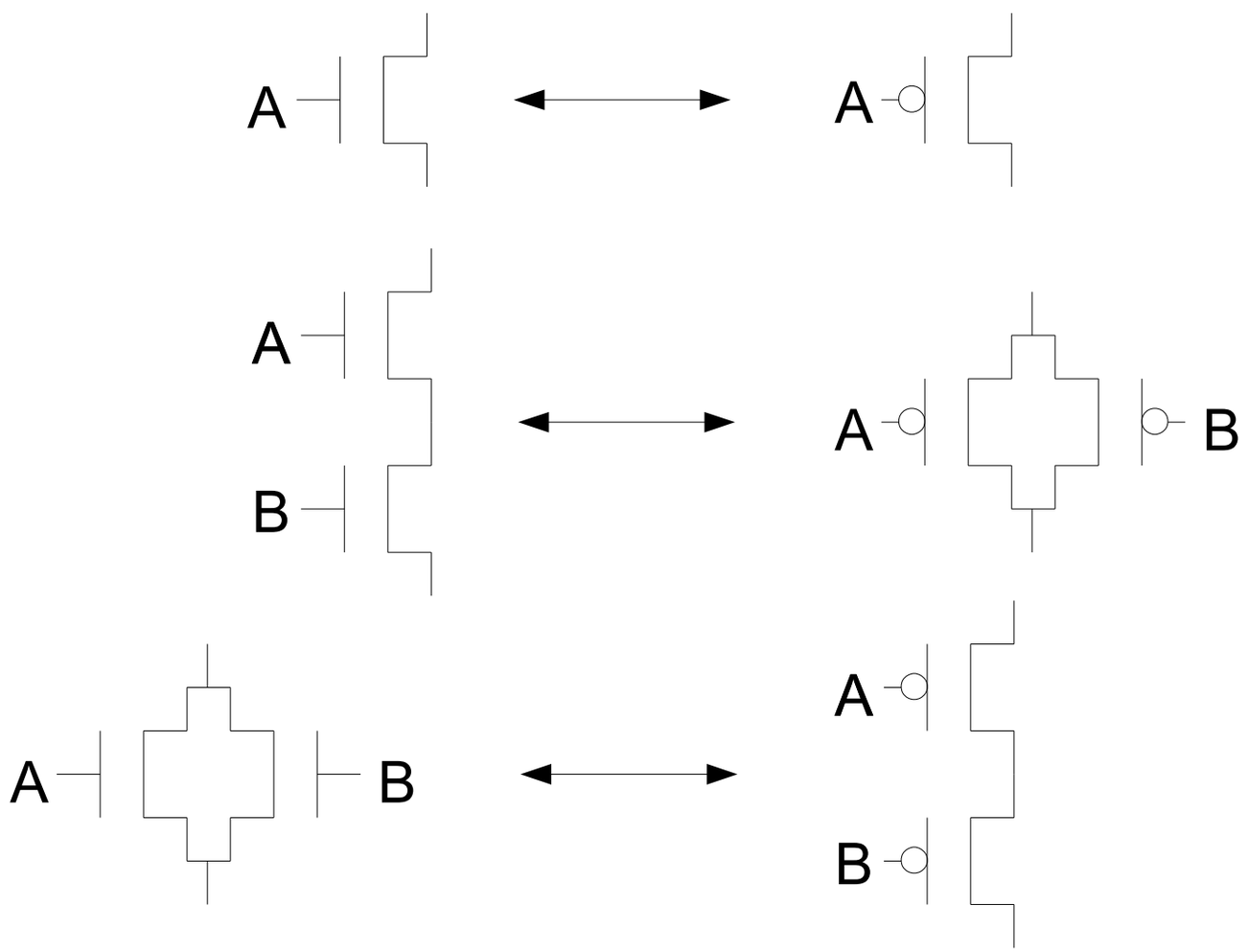
Building CMOS gates

- Have a boolean function that you need to implement?
- Build PULLDOWN or PULLUP first:
  - AND: series
  - OR: parallel
    - Note that the AND and OR here might not pertain to the actual function, but instead to certain conditions that determine the function's output.
    - Example: In a 2-input NAND, output 1 if A or B is 0. In the pullup side, note that PFETs are in parallel.
- Then, build the other part as the *dual* or complement of the first part.



CMOS complements

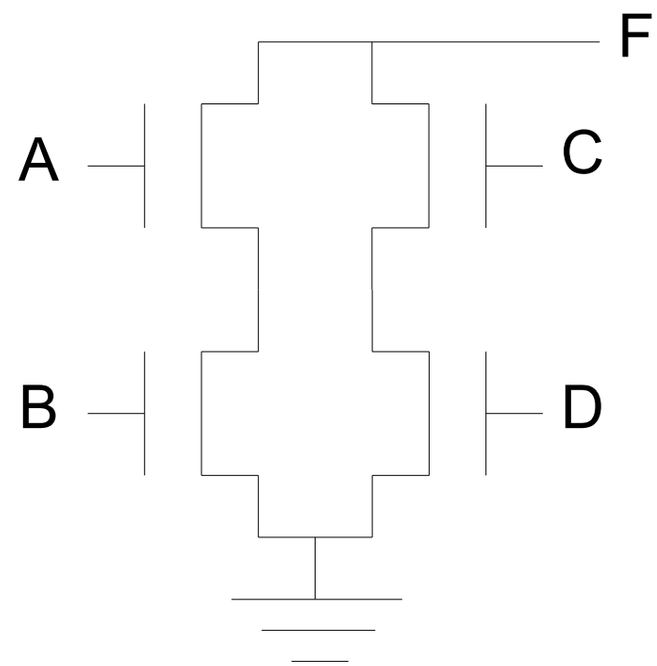
- If one is ON, the other should be OFF.



- Not a visual person? This can be done through math.

Example

- Given the pulldown part only, what function should the circuit compute?
- Build the pullup part.
  - Use visual guide for obtaining duals.



- Verify that both pulldown and pullup part compute the same function.
  - Use math (de Morgan's law).

More CMOS Functions

- 2-input AND?
- 4-input OR?
- 2-input XOR?
- F = A + !B ?
- F = !(AB) + C(!D) ?
  - Note: For all these exercises, assume that inverted inputs are available.
    - You can use !A immediately as a transistor's gate input without building an inverter for input A.
  - General rule: For every input, exactly 2 transistors must be added to the circuit.
    - Exception being the XOR gate?