520.216

Lecture 6
The Digital Abstraction

Andreas G. Andreou

Lecture Notes based on:

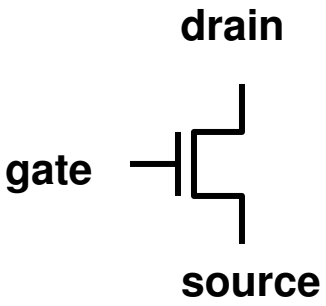
Introduction to VLSI
Stanford University

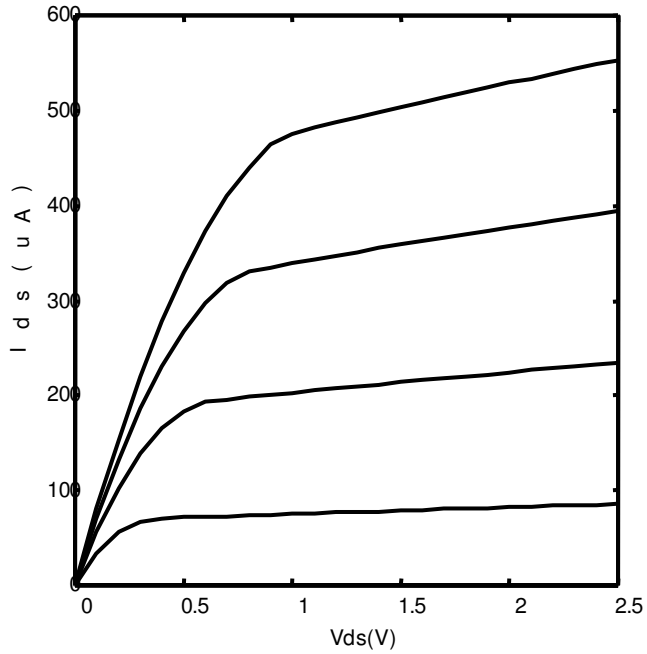
by

Subhasish Mitra

Real Transistors

- The voltage on the gate controls the current that flows between the source and drain. The transistor model is often displayed by drawing its current-voltage curve.
- But this is pretty complicated …





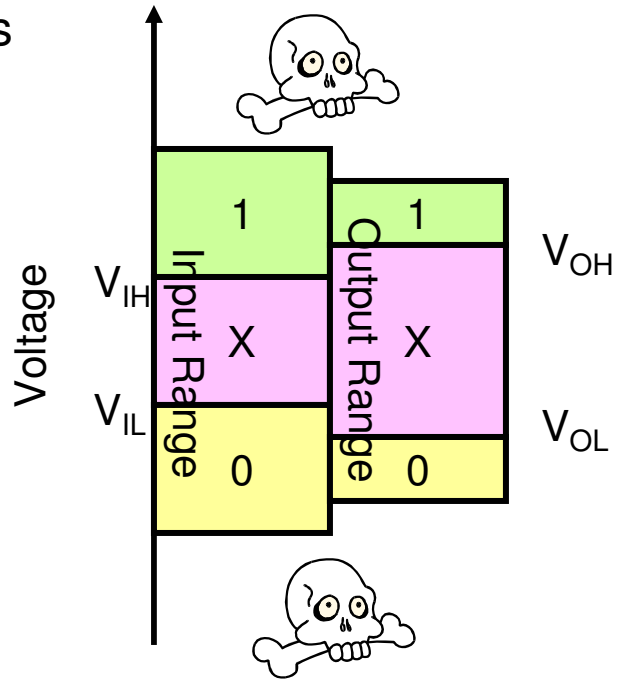
Simplify using Digital Abstraction

Rather than worrying about the precise voltages on the terminals of the transistor, guarantee that voltages will fall within two regions, one represents a logic ‘0’ and the other a ‘1’.

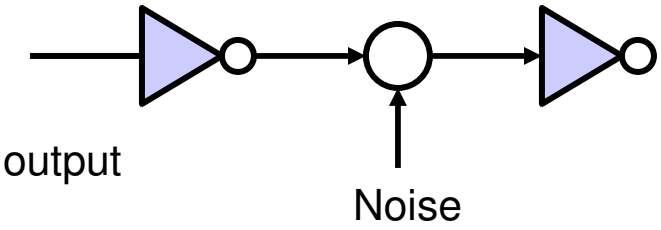
- Need to compute the output only for inputs in the allowable range
  - Much simpler than before
  - Model transistor as being either conducting, or off
- Need to ensure that the output is always in the allowable voltage range
  - Need to make sure you produce valid digital outputs to the next stage
  - Also want to have level restore

The Digital Abstraction

- Divide voltage into discrete regions
  - Logic 0
  - Logic 1
  - X - between 0 and 1
  - Out of range
    - may damage devices



- Each logic gate *restores* the signal
  - Output noise < input noise
  - Noise is not cumulative
    - In fact it is attenuated
  - Noise margin
    - How much noise won't change output



Input Output Voltage Ranges and Noise Margin

V_{IL}: Maximum input voltage recognized as logic 0

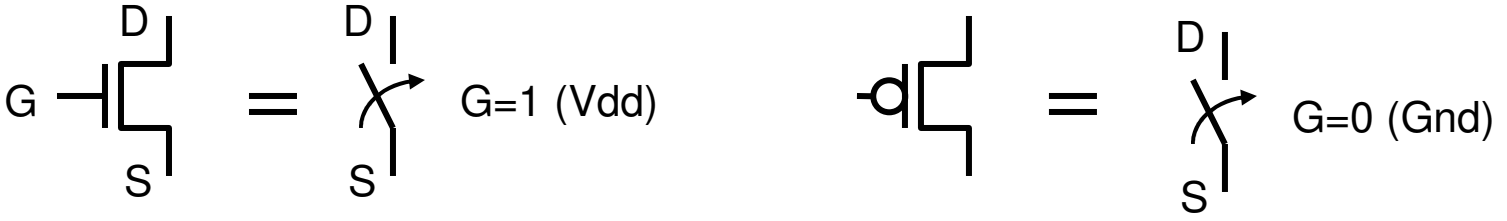
V_{IH}: Minimum input voltage recognized as logic 1

V_{OL}: Maximum output voltage corresponding to logic 0

V_{OH}: Minimum output voltage corresponding to logic 1

- Consider Inverter
  - Suppose Output = V_{OH}
- Suppose this inverter output connected to another inverter input
- If $V_{OH} < V_{IH}$: PROBLEM
  - V_{OH} better be > V_{IH}
- $V_{OH} - V_{IH}$ = NMH (noise margin high)
- NML (Noise margin Low): similar

Simplest Model for MOS transistors



NMOS transistors PMOS transistors

- Let Logic value 1 be Vdd, value 0 be Gnd
- NMOS devices are switches
  - G is 1 -> the drain D and source S are connected
  - G is 0 -> the drain D and source S are not connected
- PMOS devices are switches
  - G is 0 -> the drain D and source S are connected
  - G is 1 -> the drain D and source S are not connected

Terminology

Note that the source and drain terminals are really the same, but by convention the source terminal is the one with the lower voltage on it. Thus, the maximum voltage between the gate and the {source, drain} is the voltage between the source and the gate. (This fact will be important later.)

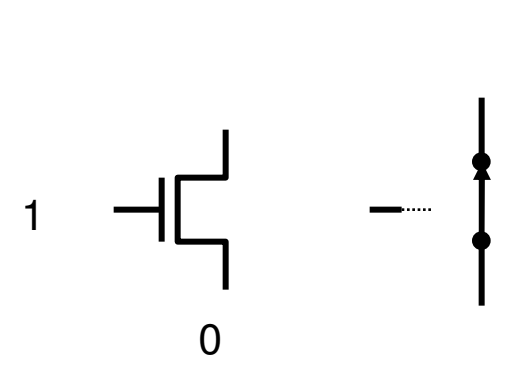
The voltage on the gate controls the connection between the source and the drain. When the gate is high, the source and drain are connected together. When it is low, the terminals are disconnected.

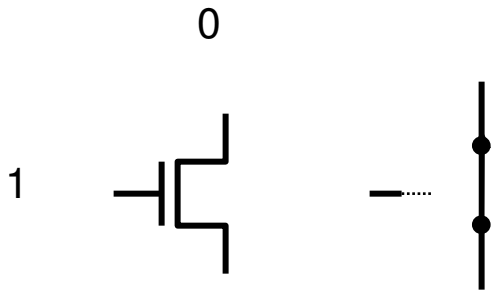
CAUTION: do NOT use the words “open” and “closed” to describe switches. Is open an open electrical circuit (no flow), or an open fluid valve (flow)? You get opposite results, depending on which analogy you use.

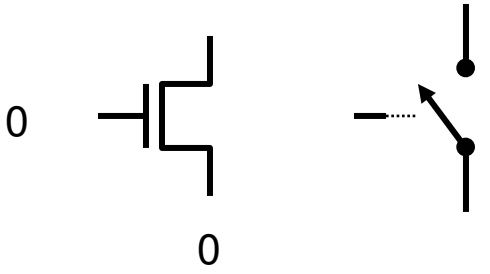
This description is for nMOS transistors. For pMOS everything is reversed. The source is the higher voltage terminal, and the transistor is on when the gate is much lower than the source. More on pMOS later

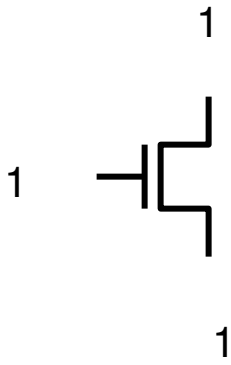
- Transistor Examples

The state of these transistors:





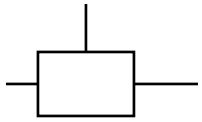




Complicated, it is really off. More on this later. Assume it is on for now

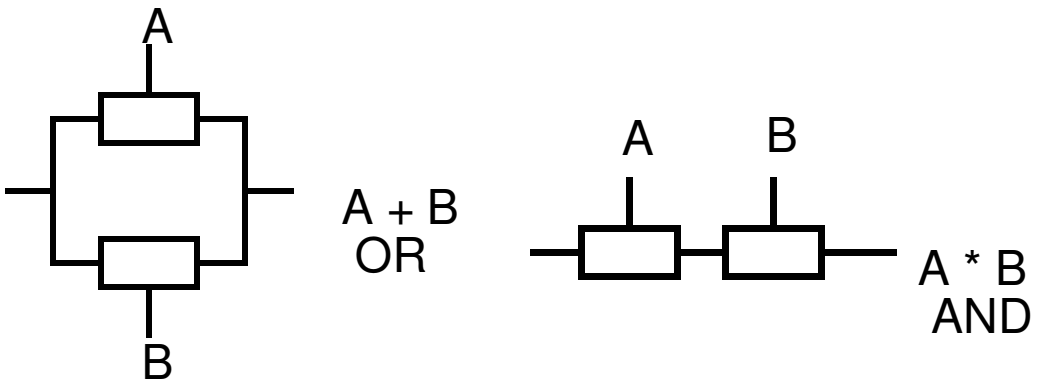
Switch Networks

- Since transistors can be modeled as switches
  - Look at what we can make out of switches
  - Draw an abstract switch as



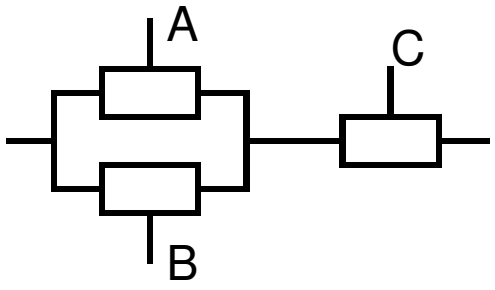
    - Control (gate) terminal is on top
- Can build switch networks
  - Are not logic gates themselves!!!
  - Are a collection of switches that still have two non-control terminals
    - Define function of a switch network as the inputs conditions that connect the two non-control terminals of the network
- Structure of switch network sets its logic functions:
    - ‘OR’ functions are constructed by parallel switches
    - ‘AND’ function are constructed by series switches

- Switch Networks

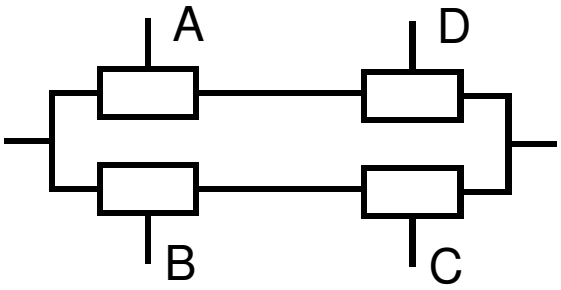


- The function of a switch network is true when the two terminals of the network are connected together. Since for parallel switches the terminals are connected if either switch is on, the function is OR. For series switches the network is conducting only if both switches are on, hence an AND.

-General Switch Networks



(A + B) C



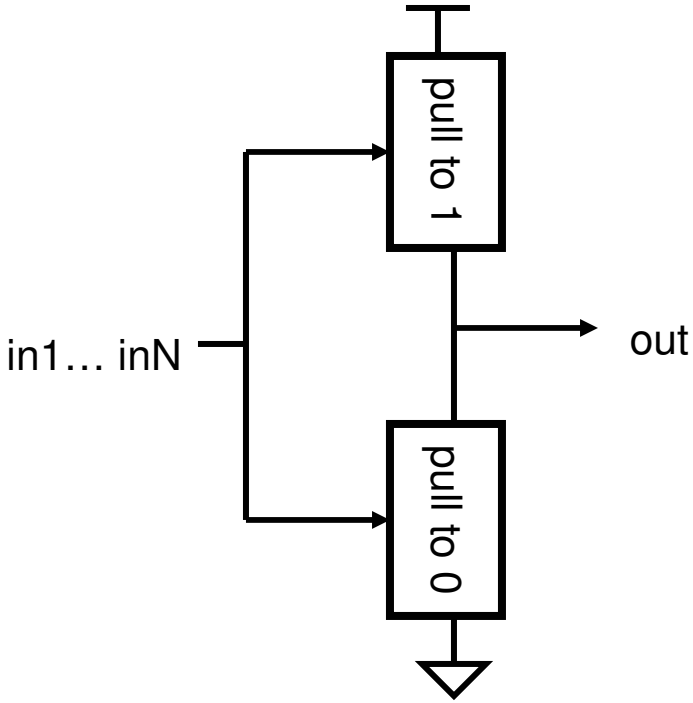
(AD) + (BC)

- More complex connections are possible
- Composition rules are simple. Use a recursive definition:
  - Parallel combination of switch networks yields an OR of the component switch networks' functions
  - Series combination of switch networks yields on AND of the component switch networks' functions.

Building Logic Functions Out of Switches

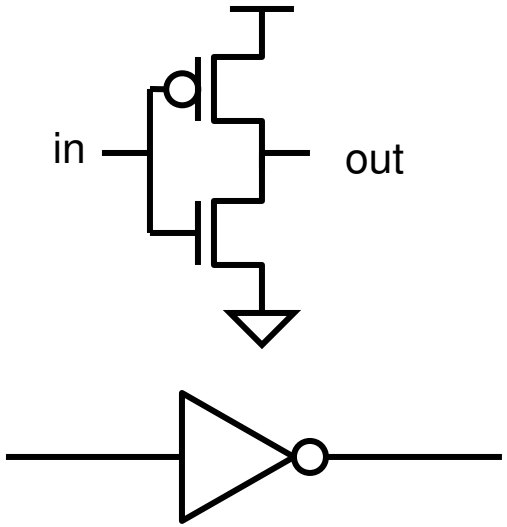
- Have switches, but we want to build logic gate
- Logic gate abstraction:
  - Unidirectional
    - Inputs drive output, but output does not drive inputs
  - Digital gate
    - Reduces input noise
  - Operation completely described by a boolean operator
- Not a difficult task if you follow a few simple rules
  - Connect switches so output is always driven to either Vdd, Gnd
    - Output noise should be small, if power supply noise is small
  - GND and Vdd are never connected to one another
    - Don’t want to short the power supply out.

Basic Idea: Build Two Switch Networks: Pull-to-1 and Pull-to-0 network



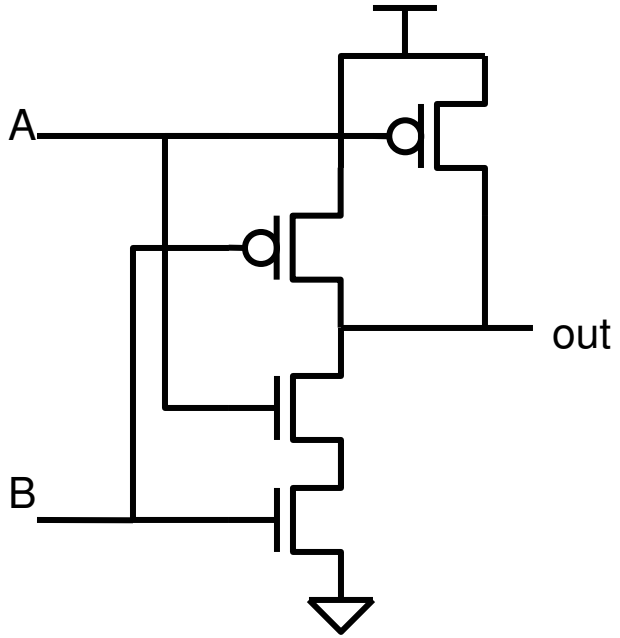
- Requirement 1: output is always driven by one of the two branches
- Requirement 2: output is never driven by both at the same time
- Make the switch functions complementary

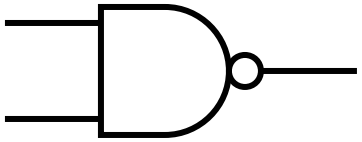
Degenerate Case: Inverter



- Meets the rules
  - Out is always driven (Either pMOS or nMOS is always active)
  - Vdd and Gnd are never shorted
    - At least with valid inputs

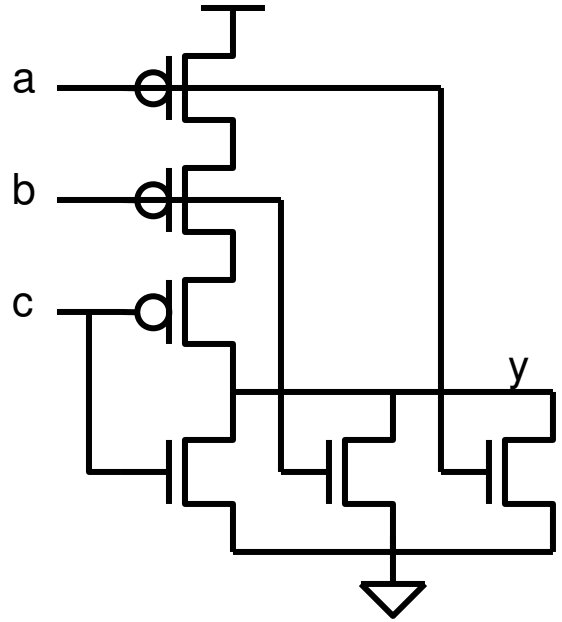
NAND Gates

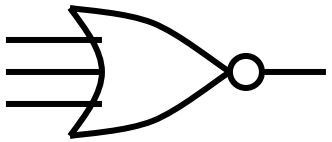




- nMOS in series pulling to Gnd
- pMOS in parallel pulling to Vdd
- Out always driven by pMOS's or nMOS's but never both
- Number of inputs ("fan-in") can be increased

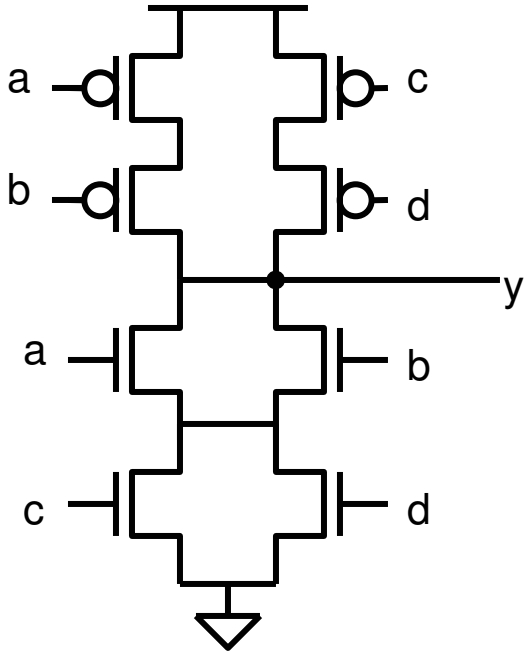
NOR Gates

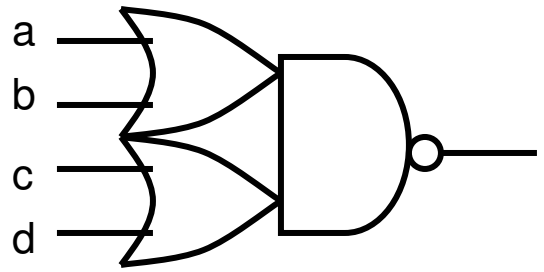




- pMOS in series
- nMOS in parallel
- Either pMOS or nMOS drives but not both
- Dual of NAND gate

More Complex Functions Possible





- Large number of possibilities
- Typical standard cell library will have many permutations

Demorgan Law is Your Friend

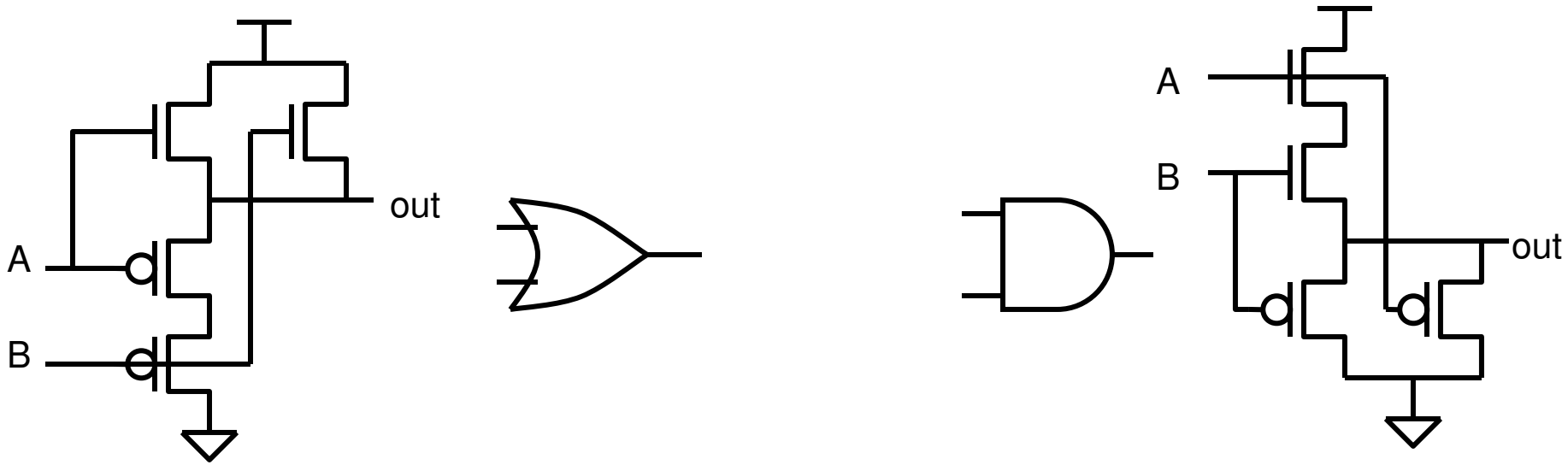
- Remember Demorgan's Law?
  - (a + b)' = a' b'
  - (a b)' = a' + b'
  - More generally the complement of a function is the dual of the function with inverted inputs.
  - Dual = exchanging the AND and OR operations

a' = NOT (a)

- Can apply to arbitrarily complex expressions.

- Now think about CMOS gates
  - Pullup switch function is complete dual of pulldown function
    - And pMOS gates invert all the inputs
    - Implies the pullup network is the dual topology of pulldown

How About AND & OR?



- Seems to obey the rules
  - Output always driven by pMOS or nMOS network
  - Vdd and GND never shorted together

- Doesn't work in reality
  - Transistors aren't really switches (Related to HW problem)

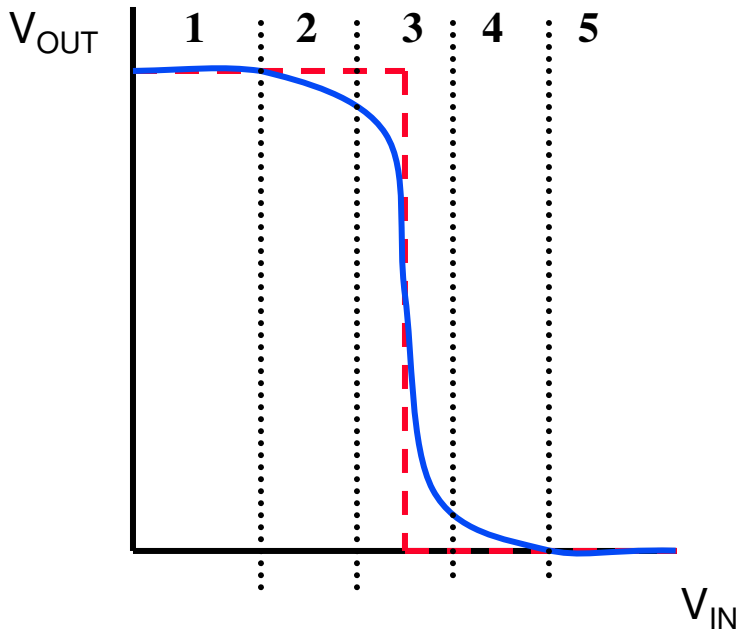
Basic Problem With Simple Model

- Voltage is a relative measurement
  - 5V does not mean anything
  - It must be 5V between A and B
  - PLEASE PLEASE remember this!!!

- In previous model nMOS conducts when Gate is at Vdd
  - At Vdd relative to what, you should ask
  - Truth is that Gate to Source must be Vdd
- In previous model pMOS conducts when Gate is at Gnd
  - At Gnd relative to what, you should ask
  - Truth is that Gate to Source must be –Vdd

- Check previous logic gates, this is always true!

Voltage Transfer Function: INV/NAND/NOR



1 – PMOS resistor, NMOS off

2 – PMOS resistor, NMOS current src

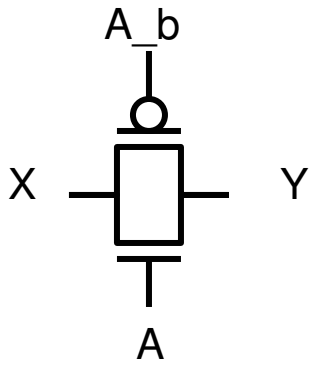
3 – PMOS current src, NMOS current src

4 – PMOS current src, NMOS resistor

5 – PMOS off, NMOS resistor

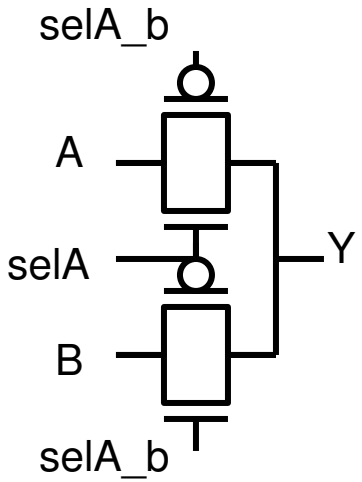
- A great digital gate
  - Gain = 0 near voltage near Vdd or Gnd
    - Noise is attenuated
  - Relatively sharp transition between 1 and 0 output
    - Allowable input range is pretty large without getting wrong answer!

Beyond Gates: Pass Transistor Structures



- Concept of switch is really versatile
  - But if we don't know the value being switched, neither NMOS nor PMOS alone implements it
- Solution: use nMOS and pMOS in parallel
  - Drive gates with complementary signals
  - Completely bidirectional
- How can we take advantage of this?

Transmission Gate Muxes

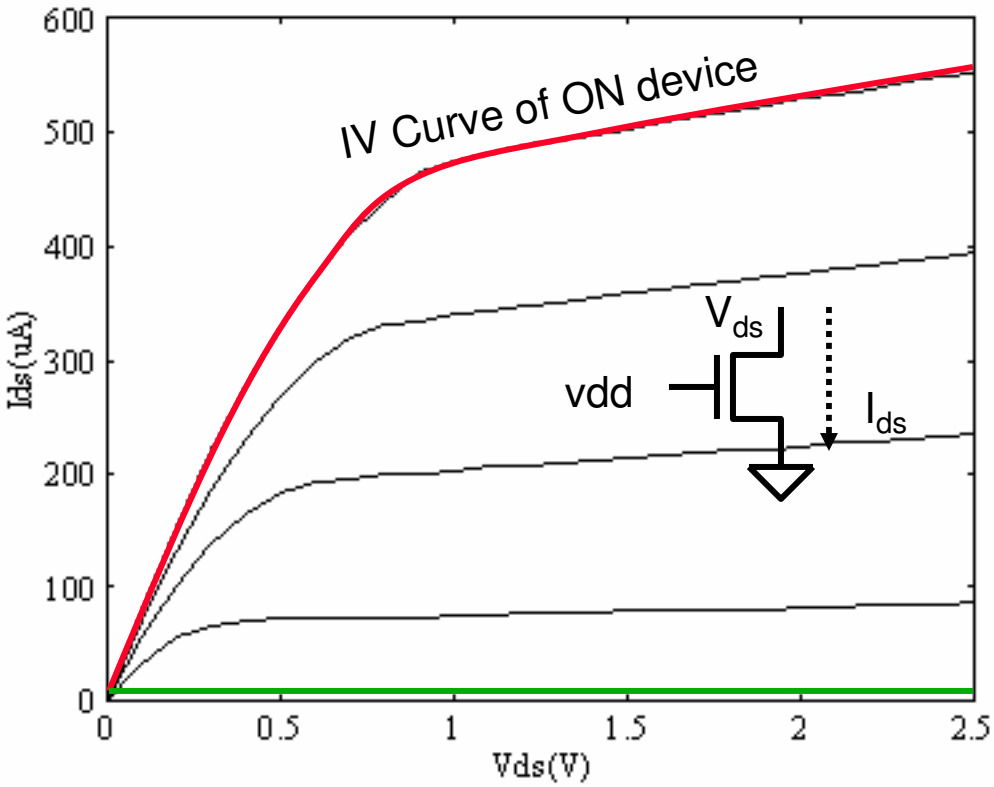


- Arbitrary number of inputs can be muxed together
  - As long as selects are mutually exclusive
- Can be cascaded in series to make more complicated networks
- CAREFUL: Not restoring as drawn (no gain)
  - Inverter often used to buffer output
- More on using transmission gates later

Arbitrary Functions with Transmission Gate MUXes

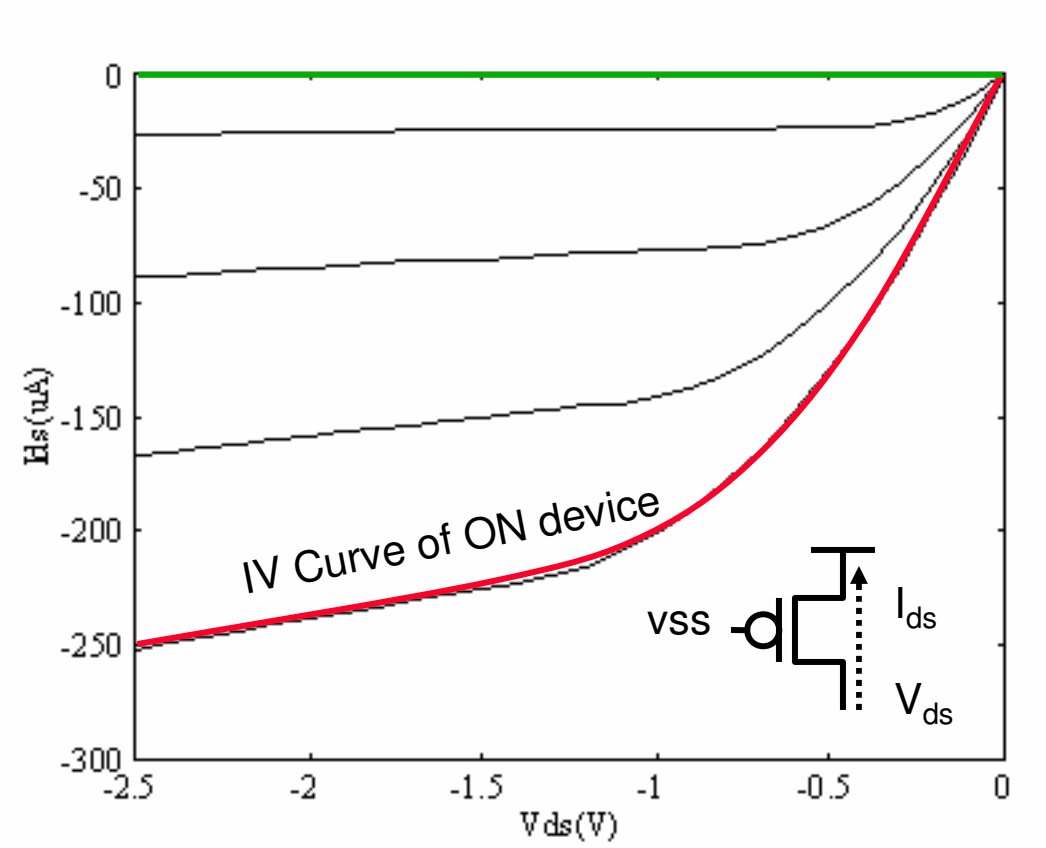
- Arbitrary logic functions using MUXes
  - e.g., n-input logic function using 2^(n-1) to 1 MUX
    - Shannon decomposition
  - Can sometimes do even better than that
- Implement MUXes with transmission gates

Operating Region for an nMOS Pulling Low



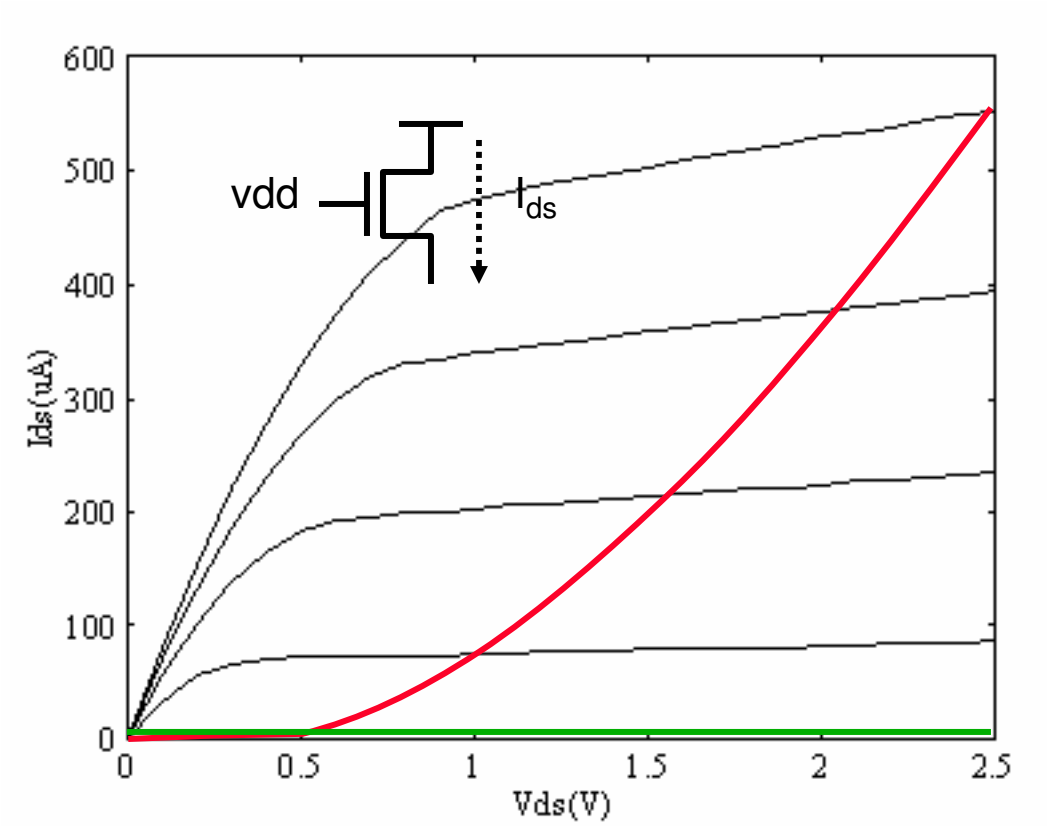
- Operates on one of two curves
- on
  - Looks like a current source initially (high V_{ds})
  - Looks like a resistor later (low V_{ds})
- off
  - Open circuit always

Output current for an pMOS Pulling High



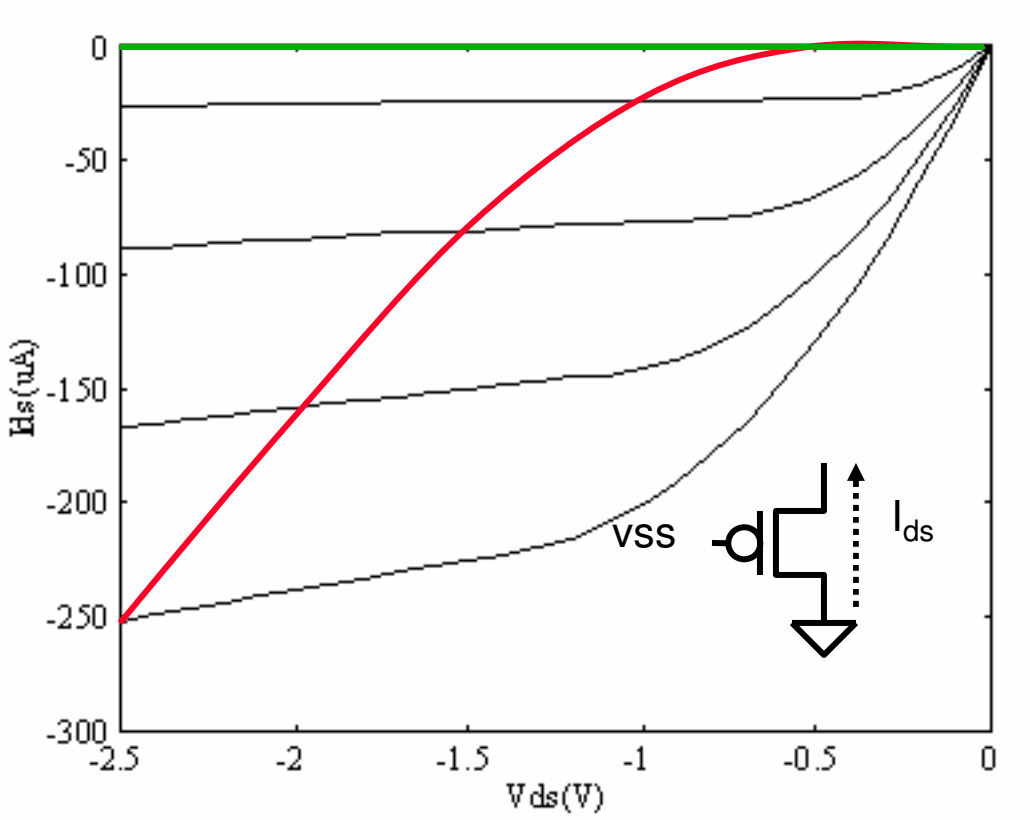
- Same behavior as NMOS
  - Open circuit when off
  - Current source or resistor when on

Output Current for a nMOS Pulling High



- V_{gs} isn't fixed:
  - Vg fixed
  - Vs is moving
    - It's the output
  - V_{gs} =Vdd – V_s
- I_{DS} =0 when V_{gs}= Vth
  - Vout = Vdd –Vth
  - Takes forever to get there

Output Current for a pMOS Pulling Low



- V_{gs} isn't fixed:
  - Vg fixed
  - Vs is moving
    - It's the output
  - $V_{gs} = - V_{ds}$
- I_{DS} =0 when V_{gs}= -Vth
  - Vout = Vth
  - Takes forever to get there