TÉCNICO LISBOA

Circuit Theory and Electronics Fundamentals

Lecture 21: Digital Circuits

- Analogue computation problems
- Digital binary number representations and algebra
- The MOS transistor as an ON/OFF controlled switch
- The simplest digital circuit: the logic inverter
- The NMOS inverter with resistive load
- The CMOS inverter

Analogue circuit use cases

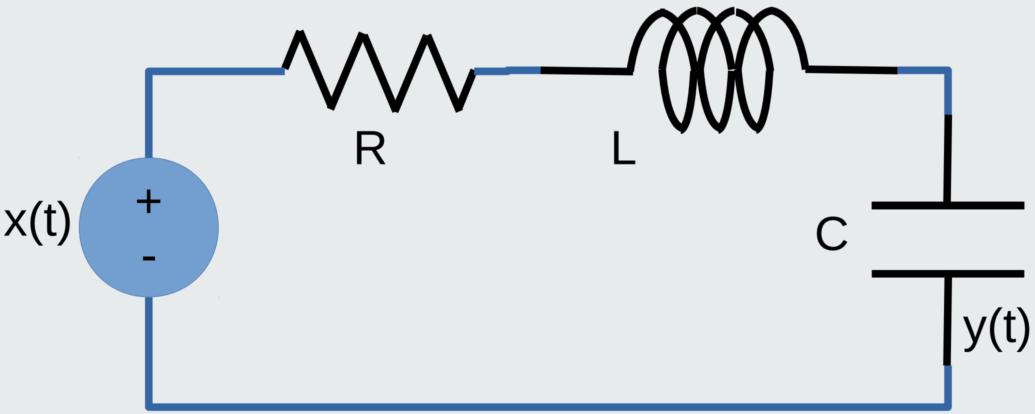
- Circuits are useful in two ways:
  - Generating and distributing energy
  - Information processing
- Analogue circuits can handle energy well
  - AC/DC converter
  - The audio amplifier was also an energy problem: amplify a signal so that it has enough a power to operate a mechanical sound speaker
- Analogue circuits can also be used to process information

Analogue computation

- Processing information using analogue circuits
  - Represent a variables as a voltages or a currents
  - Design circuits that process these variables and compute problem solutions represented by voltages or currents
- Example: use an analogue circuit to solve a second order LODE:

$$\frac{d^2 y(t)}{dt^2} + b\frac{dy(t)}{dt} + c\,y(t) = x(t)$$

Analogue computation example



$$\frac{d^2 y(t)}{dt^2} + b\frac{dy(t)}{dt} + c\,y(t) = x(t)$$

$$\frac{d^2 y(t)}{dt^2} + \frac{R}{L}\frac{dy(t)}{dt} + \frac{1}{LC}\,y(t) = x(t)$$

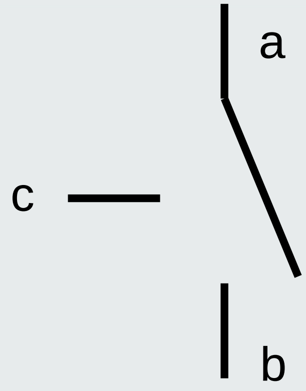
$$b = \frac{R}{L},\ c = \frac{1}{LC}$$

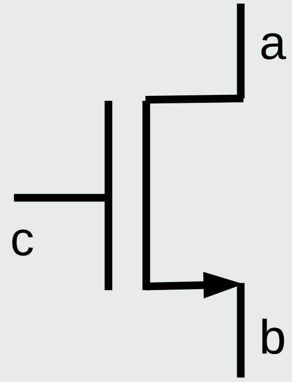
- Coefficients given by R, L and C…
- Not programmable
- Temperature dependent
- Imprecise: fabrication guarantees limited accuracy (tolerance values given)
- Expensive: R, L and C are expensive
- Voluminous: to improve precision R,L and C must be large
- Memory: how can we store a voltage or current which is a time function?

Digital computation to the rescue

- Make circuits that store numbers, not voltage or current time functions
- Use discrete mathematics instead of continuous mathematics
- Time is discrete: functions are just number ordered sets
- Use margins to become independent of temperature or fabrication imprecisions
- Can be implemented with transistors
- Transistors are relatively inexpensive
- Transistors have nanometric size therefore occupy a low volume
- Transistor can efficiently implement memory cells to store numbers
- How is what we are going to learn in this week

The transistor as a controlled switch





- Controlled switch:
- If v(c) > $V_{threshold}$ *a* is connected to *b*: switch is ON
- If v(c) < $V_{threshold}$ *a* is unconnected to *b*: switch is OFF
- Two states, ON and OFF: suggests the use of base 2 numbers
- Can be implemented with a MOS transistor if $V_{threshold} = v(b)+V_T$
- With enough of these controlled switches any processing or memory function can be implemented!

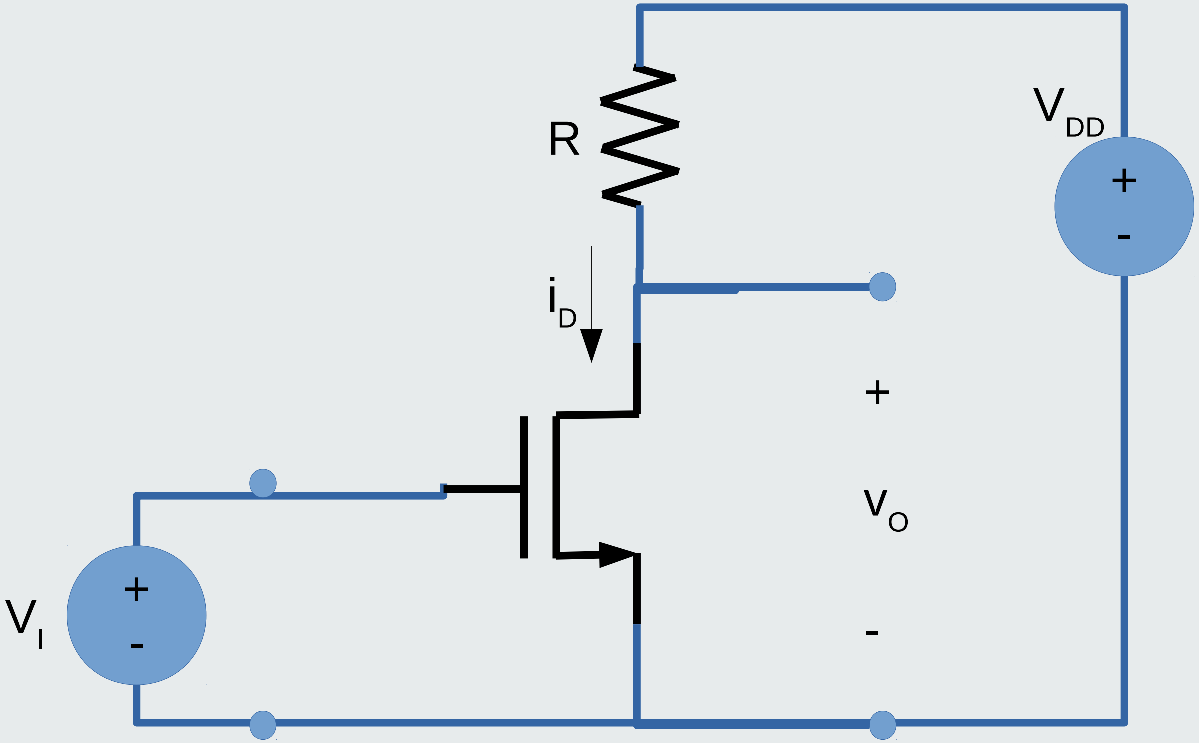
Binary numbers

- **Base 2** is as good as any other number base and is easy to implement with transistors
- Two digits: **0** and **1**
- Counting in base 2: 0, 1, 10, 11, 100, 101, 110, etc
- Besides representing numbers binary digits may also be used to represent the logic value of a proposition: **true** or **false**
- Modern computer programs can perform both logic and arithmetic operations
- Basic circuits implementing basic operations can be combined to produce astoundingly complex operations such as the relativistic movement of celestial bodies or solutions to the Schrödinger’s wave equation
- So what are the basic operations?

Binary number operations

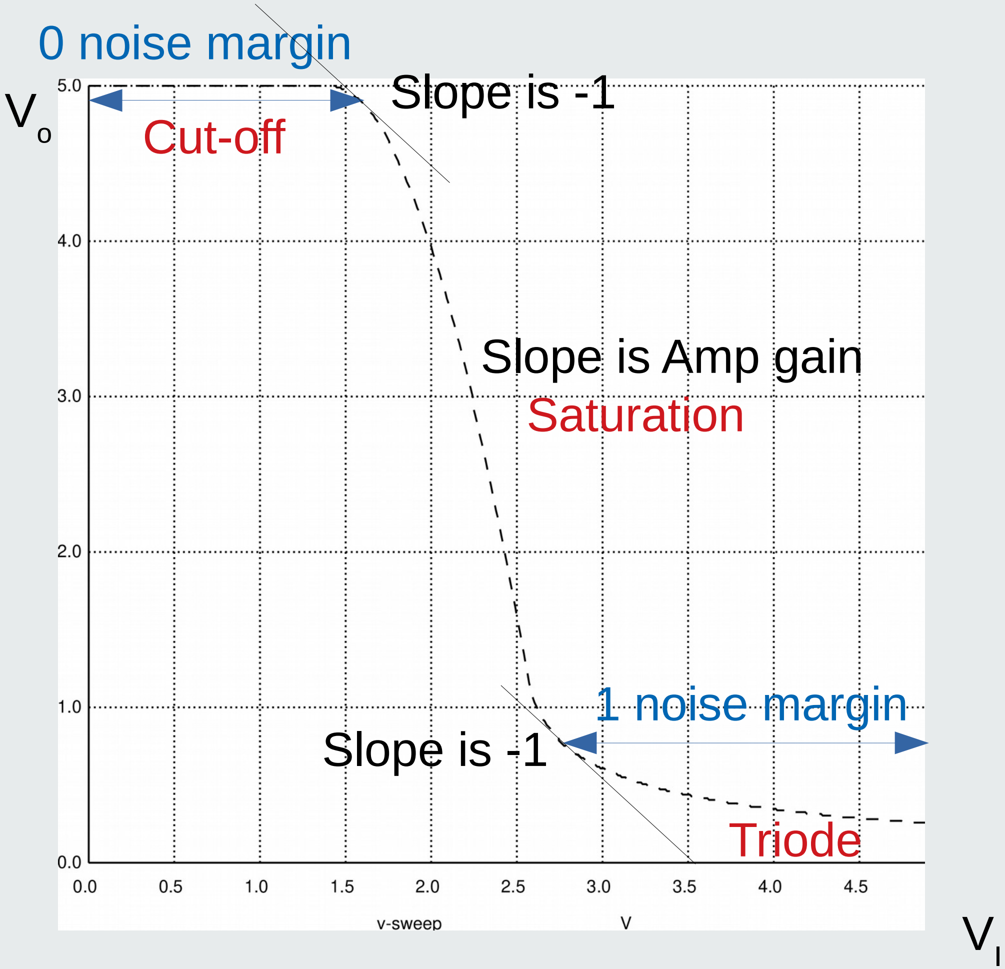
- The basic binary operations are the logic operations
  - NOT: logic negation of a proposition
  - AND: logic conjunction of propositions
  - OR: logic disjunction of propositions
  - XOR: logic exclusive disjunction of propositions
  - NAND: AND followed by NOT
  - NOR: OR followed by NOT
  - XNOR: XOR followed by NOT
- It is useful to be able to implement all the above functions (***gates***) and even more complex macros to minimize the number of transistors used
- But a Turing Complete machine can be built with just NAND gates, just NOR gates, just XOR or XNOR gates
- Example: a NAND gate can be directly implemented with 4 transistors only; if implemented as an AND followed by a NOT gate it will take 8 transistors: double the size for the same function!

The NOT gate or inverter



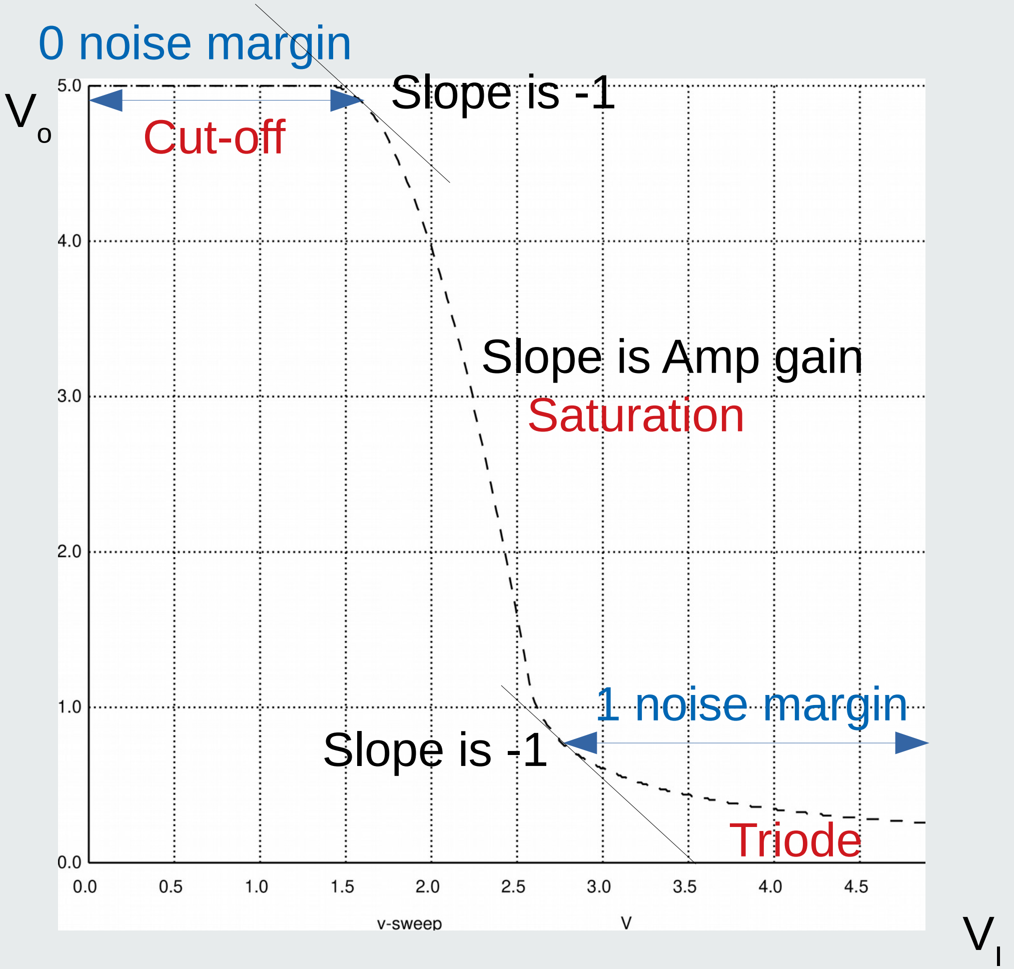
- The simplest logic gate is an inverter
- It can be implemented with just 2 transistors as we will see
- It can also be implemented with a common source amplifier!
- To see how let's simulate the $v_o(v_i)$ characteristic
- $V_I < V_T$ the transistor is in the cut-off region, $i_D=0$, and $V_O=V_{DD}$
- $V_T < V_I < V_{IX}$ the transitor is in the saturation region, i_D increases with V_I, and V_O decreases
- $V_I > V_{IX}$, the transistor is in the triode region and $V_O<V_I-V_T$

The NOT gate $v_o(v_i)$ curve



- $V_I < V_T$ the transistor is in the cut-off region, i_D=0, and V_O=V_{DD}
- $V_T < V_I < V_{IX}$ the transitor is in the saturation region, i_D increases with V_I , and V_O decreases
- $V_I > V_{IX}$, the transistor is in the triode region and $V_O < V_I - V_T$
- R=50kΩ ! Needs very high resistor for high gain (large and expensive)
- Noise margins could be better:
  - 0 noise margin: input voltage range interpreted as 0
  - 1 noise margin: input voltage range interpreted as 1
- When the output is high the output resistance is R, which is bad (high value)
- When the output is low the circuit consumes power (current flows)

Exercise: derive $V_O(V_I)$ analytically



Point where NMOS transistor enters Saturation region

$$V_I = V_T, V_O = V_{DD}$$
$$V_I = 1.4\,V, V_O = V_{DD}$$

Point where NMOS transistor Leaves Saturation and enters Triode region

$$V_O = V_I - V_T \Leftrightarrow V_{DS} = V_{GS} - V_T$$
$$I_D = k(V_I - V_T)^2$$
$$I_D = \frac{V_{DD} - V_O}{R}$$
$$\frac{V_{DD} - V_O}{R} = kV_O^2$$
$$\cancel{V_O = -1.0435\,V} \vee V_O = 0.8633\,V$$
$$V_I = 2.2633\,V$$

Exercise: derive $V_O(V_I)$ analytically (continued)

$V_O(V_I)$ in the various intervals

$$V_O[V]=\begin{cases} VDD, & 0\leq V_I\leq V_T \\ V_{DD}-Rk(V_I-V_T)^2 & V_T\leq V_I\leq 2.2633\,V \\ \dfrac{2Rk(V_I-V_T)+1}{2Rk}-\dfrac{\sqrt{4R^2k^2(V_I-V_T)^2-4RkV_{DD}+4Rk(V_I-V_T)+1}}{2Rk} & V_I>2.2633\,V \end{cases}$$

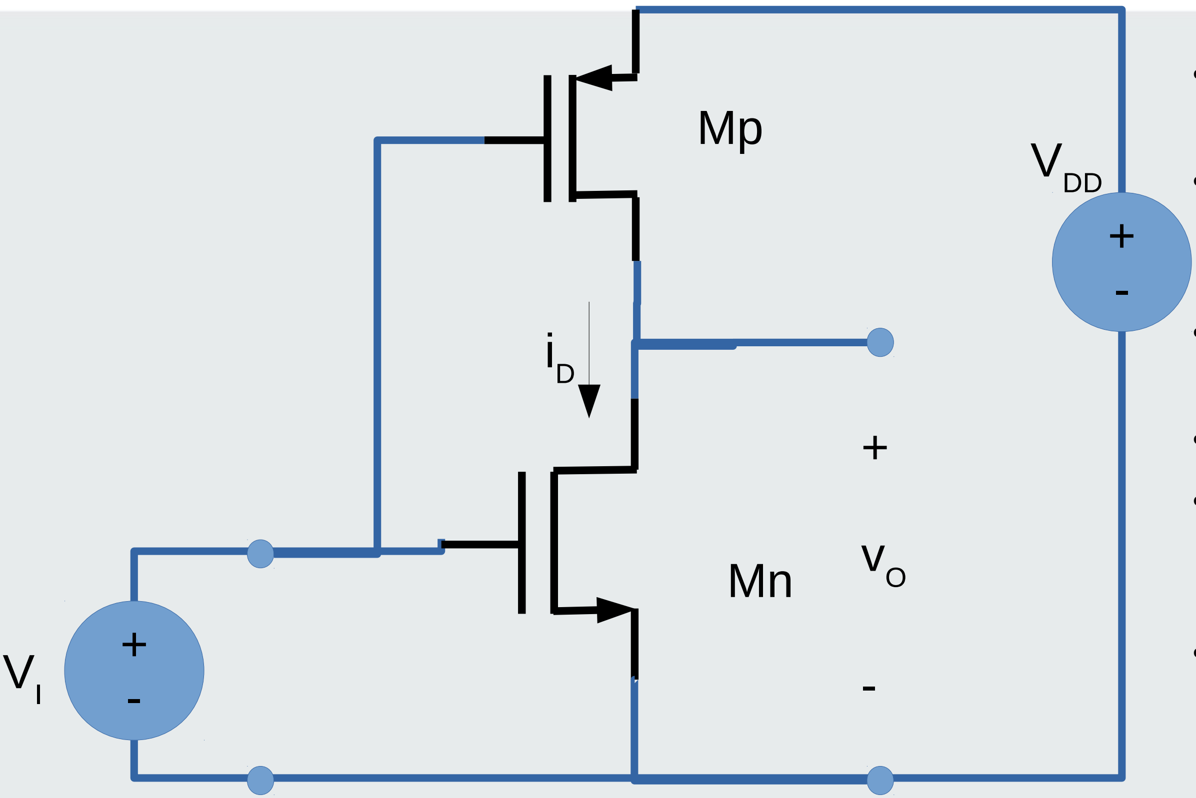
$I_D=k[2(V_{GS}-V_I)V_{DS}-V_{DS}^2]$ (triode region)

To compute the noise margins solve $dV_O/dV_I = -1$

Incremental gain in the saturation region

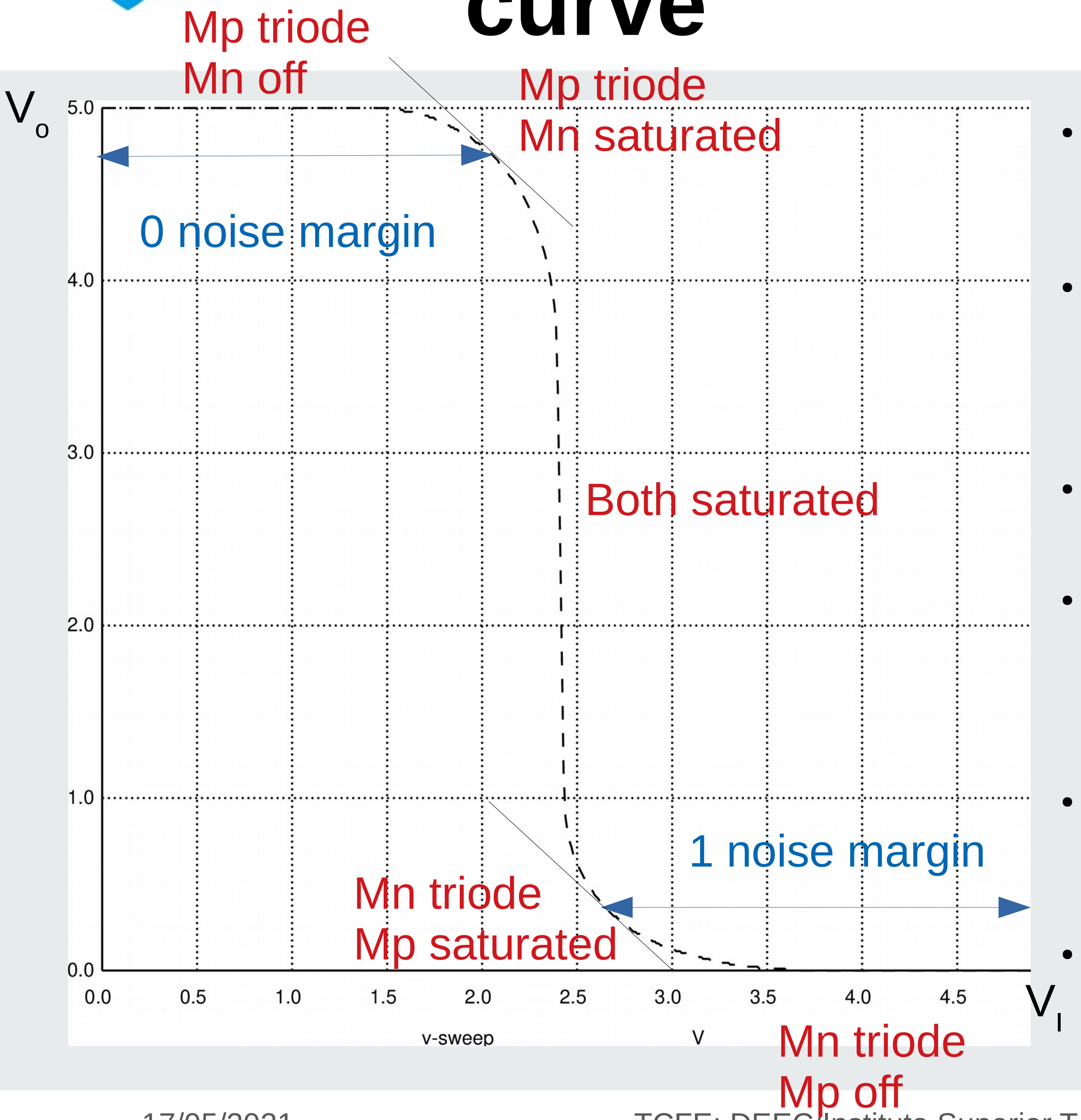
$$A_v=-g_mR$$

The CMOS inverter



- The Complementary-MOS technology for digital circuits
- Logic gates are implemented with 2 symmetric networks: the p-net and the n-net
- The CMOS inverter uses a p-transistor to replace the passive load resistor R
- The p-transistor is called an ***active*** load
- The active load switches between a very low resistance when the input is low driving a strong 1 at the output
- When the input is high the active load is off and the n-transistor delivers a very strong 0 at the output
- No static current consumption besides the small leakage current
- Output impedance is low for both 0 and 1 at the output

The CMOS NOT gate $V_O(V_I)$ curve



- $V_I < V_T$ the n-transistor is in the cut-off region and the p-transistor is in the triode region, i_D=0, and V_O=V_{DD}
- $V_T < V_I < V_{IX}$ the n-transitor is in the saturation region and the p-transistor is in the triode region, i_D increases with V_I, and V_O decreases
- $V_{IX} < V_I < V_{IY}$, both transistor are in the saturation region and V_O<V_I-V_T
- $V_{IY} < V_I < V_{IZ}$ the p-transitor is in the saturation region and the n-transistor is in the triode region, i_D and V_O decrease with V_I
- $V_I > V_{IZ}$ the p-transistor is in the cut-off region and the n-transistor is in the triode region, i_D=0, and V_O=0
- Noise margins are excellent

Exercise: derive $V_O(V_I)$ analytically (continued)

$V_O(V_I)$ in the various intervals

$$V_O[V]=\begin{cases} VDD, & 0 \le V_I \le V_{Tn} \\ V_I - V_{Tp} + \sqrt{\dfrac{k_p(V_I-V_{Tp})^2 - k_n(V_I-V_{Tn})^2 + k_p V_{DD}^2 - 2k_p V_{DD}(V_I - V_{Tp})}{k_p}}, & V_{Tn} \le V_I \le 2.3915\,V \\ [0.9915,\ 3.5915], & V_I = 2.3915\,V \\ V_I - V_{Tn} - \sqrt{\dfrac{k_n(V_I-V_{Tn})^2 - k_p(V_I-V_{Tp})^2 + k_n V_{DD}^2 - 2k_n V_{DD}(V_I - V_{Tn})}{k_n}}, & 2.3915\,V \le V_I \le VDD - V_{Tp} \\ 0, & VDD - V_{Tp} < V_I \le V_{DD} \end{cases}$$

To compute the noise margins solve $dV_O/dV_I = -1$

$$A_v = -(g_{mn}+g_{mp})(r_{on} \| r_{op}) \approx -g_m r_o$$

$$r_{on} = r_{op} = r_o = \frac{\lambda^{-1}}{I_D}$$

p and n-transistors adapted

$$g_{mn} = g_{mp} = g_m$$

Conclusion

- Analogue computation problems
- Digital binary representations and Boolean algebra
- The MOS transistor as an ON/OFF switch
- The simplest digital circuit: the logic inverter
- The NMOS inverter with resistive load
- The CMOS inverter