

# **Розділ 1**

Девид М. Харріс і Сара Л. Харріс

## **Цифрова схемотехніка і архітектура комп'ютера**

**Дані слайди перекладені з англійської мови на російську командою співробітників університетів і компаній з Росії, України, США у складі:**

Олександр Барабанов - доцент кафедри комп'ютерної інженерії факультету радіофізики, електроніки і комп'ютерних систем Київського національного університету імені Тараса Шевченка, к.ф.-м. н., Київ, Україна;

Антон Брюзгін - начальник відділу АТ «Вибро-прибор», Санкт-Петербург, Росія.

Євгеній Короткий - доцент кафедри конструювання електронно-обчислювальної апаратури факультету електроніки Національного технічного університету України «Київський Політехнічний Інститут», керівник відкритої лабораторії електроніки Lampra, к.т.н., Київ, Україна;

Євгенія Литвинова – заступник декана факультету комп'ютерної інженерії і керування, д.т.н., професор кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна;

Юрій Панчул - старший інженер по розробленню і верифікації блоків мікропроцесорного ядра в команді MIPS I6400, Imagination Technologies, відділок в Санта-Кларі, Каліфорнія, США;

Дмитро Рожко - інженер-програміст АТ «Вибро-прибор», магістр Санкт-Петербурзького державного автономного університету аерокосмічного приладобудування (ГУАП), Санкт-Петербург, Росія;

Володимир Хаханов – декан факультету комп'ютерної інженерії і керування, проректор з наукової роботи, д.т.н., професор кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки, Харків, Україна;

Світлана Чумаченко – завідувач кафедри автоматизації проектування обчислювальної техніки Харківського національного університету радіоелектроніки, д.т.н., професор, Харків, Україна.

**Дані слайди перекладені з російської мови на українську студентами спеціальності “Комп'ютерна інженерія”**

# Зміст

- Мета курсу
- Мистецтво керувати складністю
- Цифрова абстракція
- Системи числення
- Логічні елементи
- Логічні рівні
- КМОП транзистори
- Енергоспоживання

## Мета курсу

- Зрозуміти, що відбувається всередині комп'ютера
- Вивчити основні принципи цифрової схемотехніки
- Навчитися розробляти проекти зростаючої складності
- Навчитися проектувати мікропроцесори

# Мистецтво керування

Абстракція

Дисципліна

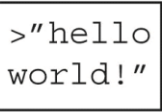


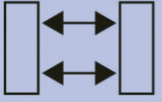
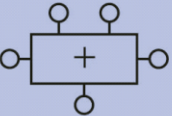
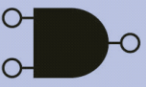
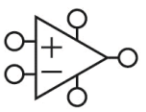


Три базових принципи:

- Ієрархічність
- Модульність
- Регулярність

# Абстрація

- Виключення з розгляду деталей, які у даному контексті неважливі

focus of this course

Application Software		programs
Operating Systems		device drivers
Architecture		instructions registers
Micro-architecture		datapaths controllers
Logic		adders memories
Digital Circuits		AND gates NOT gates
Analog Circuits		amplifiers filters
Devices		transistors diodes
Physics		electrons

# Дисципліна

- Умисне обмеження вибору можливих проектних рішень
- Приклад: Цифрова дисципліна
  - Використання дискретних значень напруг замість неперервних
  - Цифрові системи простіше проектувати, ніж аналогові – можна створити більш складний пристрій
  - Аналогові попередники були витіснені цифровими системами:
    - наприклад, цифрові камери, цифрове телебачення, сотові телефони, компакт-диски

# Три базових принципи

- **Ієрархічність**
  - Система розділяється на модулі і підмодулі
- **Модульність**
  - Кожний модуль має чітко визначені функції і інтерфейси
- **Регулярність**
  - Заохочення подібності, що дозволяє багатократно використовувати модулі



# Цифрова абстракція

Більшість фізичних величин **неперервні**

- Потенціал провідника
- Частота колевань
- Положення тіла

Цифрова абстракція розглядує **дискретну множину**  
можливих значень

# Цифрова дисципліна: двійкові змінні

Два дискретні значення:

1 і 0

1, Істина, Більша величина

0, Хиба, Мала величина

**1 і 0:** Величина напруги, кут повороту шестерні, рівень рідин і т.д.

Цифрові схеми використовують значення **напруги** для подання 0 і 1

***Бім (Bit):*** Двійкова цифра (Binary digit)

# Джордж Буль, 1815 - 1864

- Народився у сім'ї небагатого ремісника
- Самостійно вивчав математику і став викладачем Королівського коледжу в Ірландії.
- Написав роботу *Дослідження законів мислення* (1854)
- Ввів двійкові змінні
- Ввів три основних логічних операторів: І, АБО, НЕ (AND, OR, NOT)

# Системи числення

Десяткові числа:

$$\begin{array}{cccc} 1000 & 100 & 10 & 1 \\ 5 & 3 & 7 & 4_{10} \end{array}$$

Двійкові числа:

$$\begin{array}{cccc} 8 & 4 & 2 & 1 \\ 1 & 1 & 0 & 1_2 \end{array}$$

# Системи числення

Десяткове число

$$5\ 3\ 7\ 4_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

Двійкове число

$$1\ 1\ 0\ 1_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$

## Степені числа 2

- $2^0 = 1$
- $2^1 = 2$
- $2^2 = 4$
- $2^3 = 8$
- $2^4 = 16$
- $2^5 = 32$
- $2^6 = 64$
- $2^7 = 128$
- $2^8 = 256$
- $2^9 = 512$
- $2^{10} = 1024$
- $2^{11} = 2048$
- $2^{12} = 4096$
- $2^{13} = 8192$
- $2^{14} = 16384$
- $2^{15} = 32768$

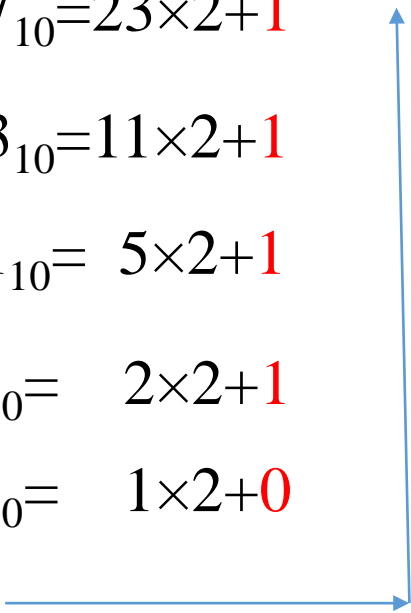
Бажано запам'ятати до  $2^{10}$

# Перетворення системи числення

Перетворення двійкового числа у десяткове

$$10011_2 = 16 \times 1 + 8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 19_{10}$$

Перетворення десяткового числа у двійкове

$$\begin{array}{l} 47_{10} = 23 \times 2 + \mathbf{1} \\ 23_{10} = 11 \times 2 + \mathbf{1} \\ 11_{10} = 5 \times 2 + \mathbf{1} \\ 5_{10} = 2 \times 2 + \mathbf{1} \\ 2_{10} = 1 \times 2 + \mathbf{0} \\ \mathbf{1} \end{array} \quad \begin{array}{l} \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \\ \uparrow \end{array} \quad = 101111$$


# Діапазони значень десятикових і двійкових чисел

$N$ -розрядне десятикове число

Скільки значень?  $10^N$

Діапазон?  $[0, 10^N - 1]$

Приклад: Трерозрядне десятикове число

$10^3 = 1000$  можливих значень

Діапазон:  $[0, 999]$

$N$ -бітове двійкове число

Скільки значень?  $2^N$

Діапазон:  $[0, 2^N - 1]$

Приклад: Трирозрядне двійкове число

$2^3 = 8$  можливих значень

Діапазон:  $[0, 7] = [\text{от } 000_2 \text{ до } 111_2]$



# Шістнадцяткові числа

Шістнадцяткова цифра	Десятковий еквівалент	Двійковий еквівалент
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

# Перетворення шістнадцяткових чисел

- Перетворення шістнадцяткових чисел у двійкові:
  - Перетворити  $4AF_{16}$  (також записується як  $0x4AF$ ) в двійкове число
  - $0100\ 1010\ 1111_2$
- Перетворення шістнадцяткових чисел у десяткові:
  - Перетворити  $4AF_{16}$  в десяткове число
  - $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

# Біти, півбайти і байти

- Біти

10010110

most significant bit      least significant bit

- Байти і

- півбайти (nibble)

byte

10010110

nibble

- Байти

CEBF9AD7

most significant byte      least significant byte

# Великі степені 2

- $2^{10} = 1 \text{ кіло} \approx 1000 \text{ (1024)}$
- $2^{20} = 1 \text{ мега} \approx 1 \text{ мільйон (1,048,576)}$
- $2^{30} = 1 \text{ гіга} \approx 1 \text{ мільярд (1,073,741,824)}$

Чому дорівнює  $2^{24}$ ?

-  $2^4 \times 2^{20} \approx 16 \text{ мільйонів}$

Скільки значень може задати 32-бітова змінна?

-  $2^2 \times 2^{30} \approx 4 \text{ мільярди}$

## Додавання

- Десяткове

$$\begin{array}{r} 11 \leftarrow \text{carries} \\ 3734 \\ + 5168 \\ \hline 8902 \end{array}$$

- Двійкове

$$\begin{array}{r} 11 \leftarrow \text{carries} \\ 1011 \\ + 0011 \\ \hline 1110 \end{array}$$

# Переповнення

- Цифрові системи працюють з **фіксованою кількістю розрядів**
- Переповнення: коли результат не поміщається у доступній кількості разрядів
- Приклад додавання  $11 + 6$

$$\begin{array}{r} 111 \\ 1011 \\ + 0110 \\ \hline 10001 \end{array}$$

Переповнення

# Двійкові числа із знаком

- Числа в прямому коді
- Числа в доповнительному коді

# Числа в прямому коді

- Один знаковий біт,  $N-1$  бітів величини
- Знаковий біт є старшим (самим лівим) бітом

- Позитивні числа:

- знаковий біт = 0

$$A : \{a_{N-1}, a_{N-2}, \dots, a_2, a_1, a_0\}$$

- Негативні числа:

- знаковий біт = 1

$$A = (-1)^{a_{n-1}} \sum_{i=0}^{n-2} a_i 2^i$$

- Пример, 4-бітове представлення числа  $\pm 6$  в прямом коде

- $+6 = \mathbf{0110}$

- $-6 = \mathbf{1110}$

- Діапазон  $N$ -бітового числа у прямому коді:

- $\mathbf{[-(2^{N-1}-1), 2^{N-1}-1]}$



# Числа у прямому коді

- Недоліки:
  - Стандартний спосіб додавання не працює, наприклад,  $-6 + 6$ :

1110

+0110

10100 (не правильно!)

- Два подання числа 0 ( $\pm 0$ ):

1000

0000

# Числа в доповняльному коді

- Не мають проблем прямого кода
  - Алгоритм додавання працює
  - Єдине подання 0
- Старший біт має вагу  $-2^{N-1}$

$$A = a_{n-1}(-2^{n-1}) + \sum_{i=0}^{n-2} a_i 2^i$$

- Найбільше позитивне 4-бітове число: **0111**
- Найбільше (за модулем) негативне 4-бітове число: **1000**
- Старший біт, як і раніше, вказує знак (1 = негативне, 0 = позитивне)
- Діапазон  $N$ -бітового числа в доповняльному коді:  **$[-(2^{N-1}-1), 2^{N-1}-1]$**

# Операція “доповнення до двох”

- Зміна знаку числа, заданого в доповняльному коді
- Метод:
  1. Інвертувати біти
  2. Додати 1
- Приклад: Змінити знак  $3_{10} = 0011_2$ 
  1. **0011**
  2. **1100**
  3. **+ 1**  
**1101 = -3<sub>10</sub>**

## Приклади чисел в доповняльному коді

- Знайти подання в доповняльному коді  $6_{10} = 0110_2$ 
  - 1001
  - + 1
$$1010 = -6_{10}$$
- Чому дорівнює десяткове подання числа в доповняльному коді  $1001_2$ ?
  - 0110
  - + 1
$$0111_2 = 7_{10}, \text{ отже } 1001_2 = -7_{10}$$

# Додавання чисел в доповняльному коді

- Додати числа  $6 + (-6)$  у доповняльному коді

$$\begin{array}{r} 111 \\ 0110 \\ + 1010 \\ \hline 10000 \end{array}$$

- Додати числа  $-2 + 3$  у доповняльному коді

$$\begin{array}{r} 111 \\ 1110 \\ + 0011 \\ \hline 10001 \end{array}$$

# Збільшення кількості бітів знаковим розширенням

- Знаковий біт копіюється у всі нові старші біти
- Значенн числа не змінюється
- **Приклад 1:**
  - 4-бітове подання:  $3 = 0011$
  - 8-бітове подання:  $00000011$
- **Приклад 2:**
  - 4-бітове подання:  $-5 = 1011$
  - 8-бітове подання:  $11111011$

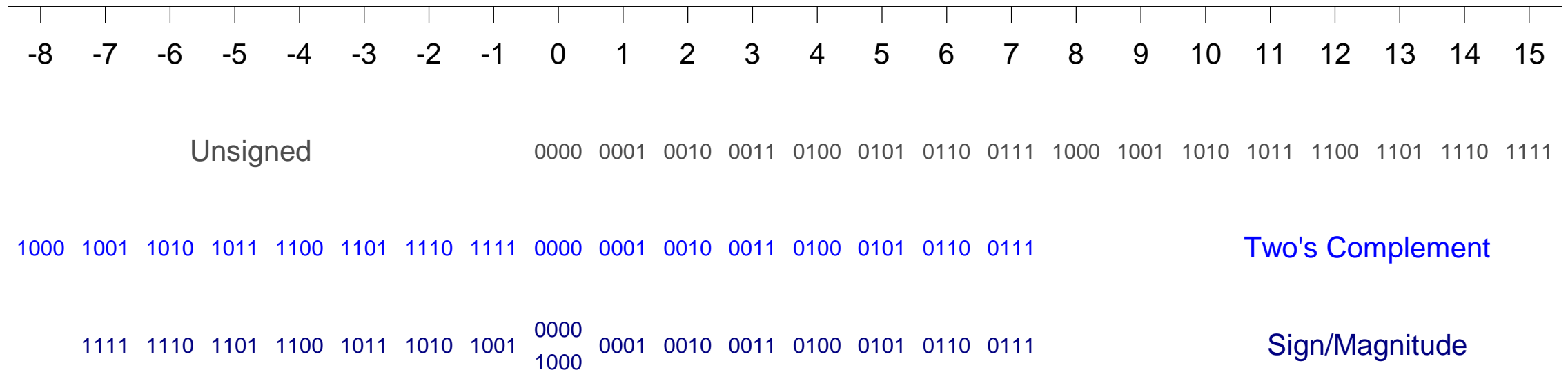
# Збільшення кількості бітів доповнення нулями

- Всі нові старші біти приймають нульове значення
- Значення негативних чисел змінюється
- **Приклад 1:**
  - 4-бітова величина:  $0011_2 = 3_{10}$
  - 8-бітова величина:  $00000011 = 3_{10}$
- **Приклад 2:**
  - 4-бітова величина:  $1011_2 = -5_{10}$
  - 8-бітова величина:  $00001011 = 11_{10}$

# Порівняння способів подання двійкових чисел

Подання	Діапазон
Числа без знаку	$[0, 2^N-1]$
Числа в прямому коді	$[-(2^{N-1}-1), 2^{N-1}-1]$
Числа в доповняльному коді	$[-2^{N-1}, 2^{N-1}-1]$

Приклад: 4-бітове подання:



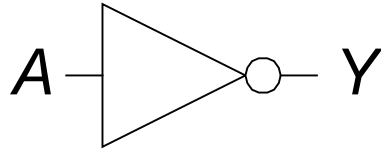


# Логічні елементи

- **Виконують логічні функції**
  - Інверсія (НЕ), І (AND), АБО (OR), І-НЕ(NAND), АБО-НЕ(NOR), і т.д.
- **З одним входом**
  - елемент НЕ, буфер
- **С двома входами**
  - І, АБО, І-НЕ, АБО-НЕ, Виключне АБО, Виключне АБО-НЕ
- **З декількома входами**

# Логічні елементи з одним входом

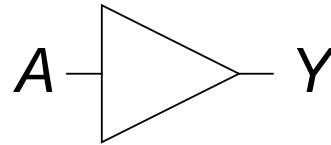
**NOT**



$$Y = \overline{A}$$

A	Y
0	1
1	0

**BUF**

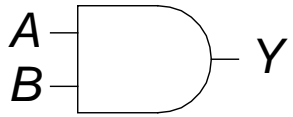


$$Y = A$$

A	Y
0	0
1	1

# Логічні елементи з двома входами

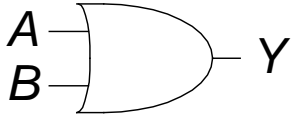
**AND**



$$Y = AB$$

A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

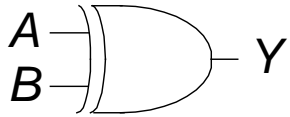
**OR**



$$Y = A + B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

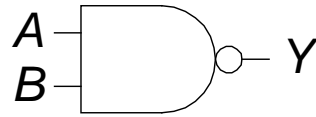
**XOR**



$$Y = A \oplus B$$

A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

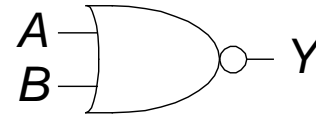
**NAND**



$$Y = \overline{AB}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

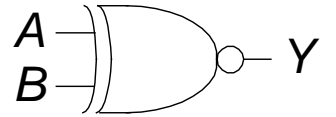
**NOR**



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

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

**XNOR**

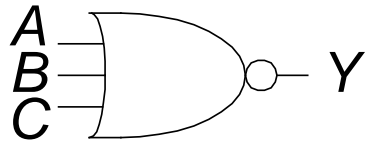


$$Y = \overline{A \oplus B}$$

A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

# Логічні елементи з декількома входами

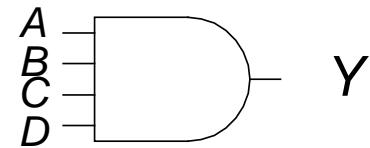
## NOR3



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

A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

## AND4



$$Y = ABCD$$

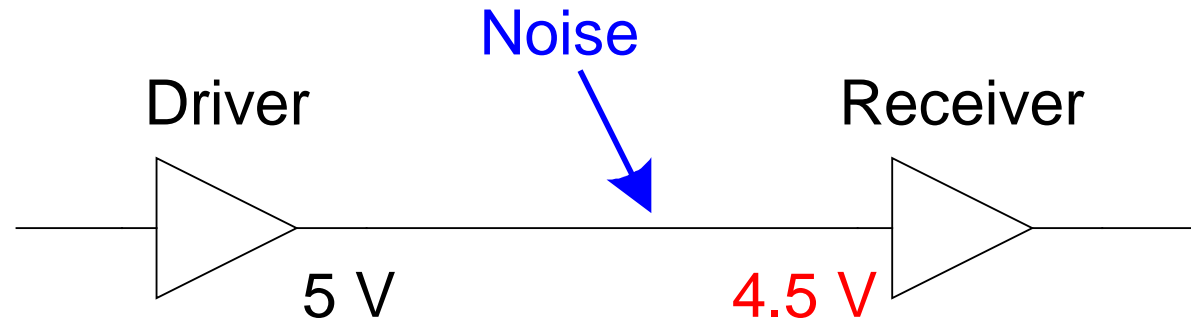
A	B	C	Y
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

# Логічні рівні

- Дискретні рівні напруги задають 1 і 0
- Наприклад:
  - 0 = *земля* (GND) або 0 В
  - 1 =  $V_{DD}$  або 5 В
- Як трактувати напругу 4.99 В? Це 0 чи 1?
- Як трактувати напругу 3.2 В?
- Різні діапазони для входів і виходів забезпечують роботу схем при наявності завад і шумів

# Що таке завада?

- **Люба завада, яка спотворює сигнал**
  - Наприклад, опір провідників, завади джерела живлення, наведення від сусідніх провідників і т.д.
- **Приклад:** елемент (йог вихідний каскад) видає 5 В, але через опір довгого провідника на приймач поступає 4.5 В

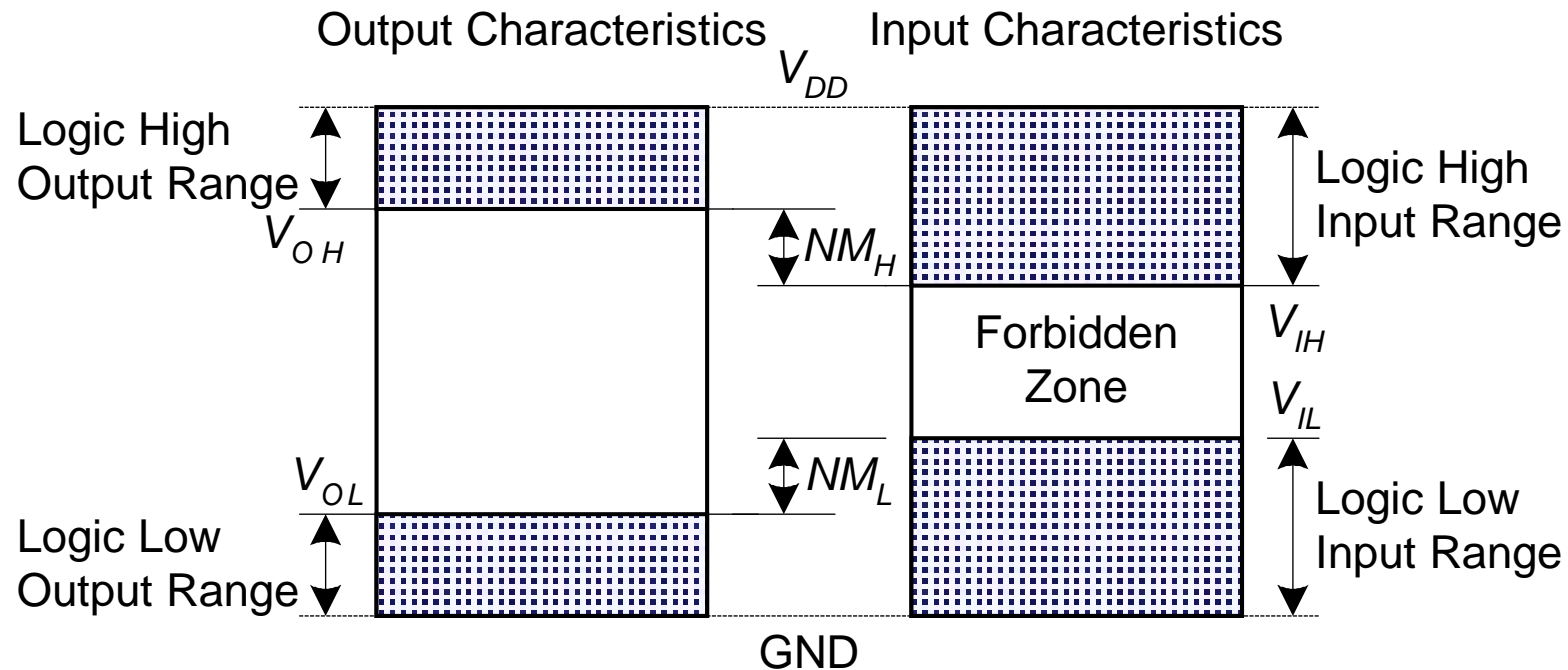


# Статическая дисциплина

- Якщо на вхід елемента поступають коректні логічні значення, то на його виході формуються коректні вихідні сигнали
- Для задання дискретних величин використовується обмежений діапазон напруг



Допустимі рівні шумів

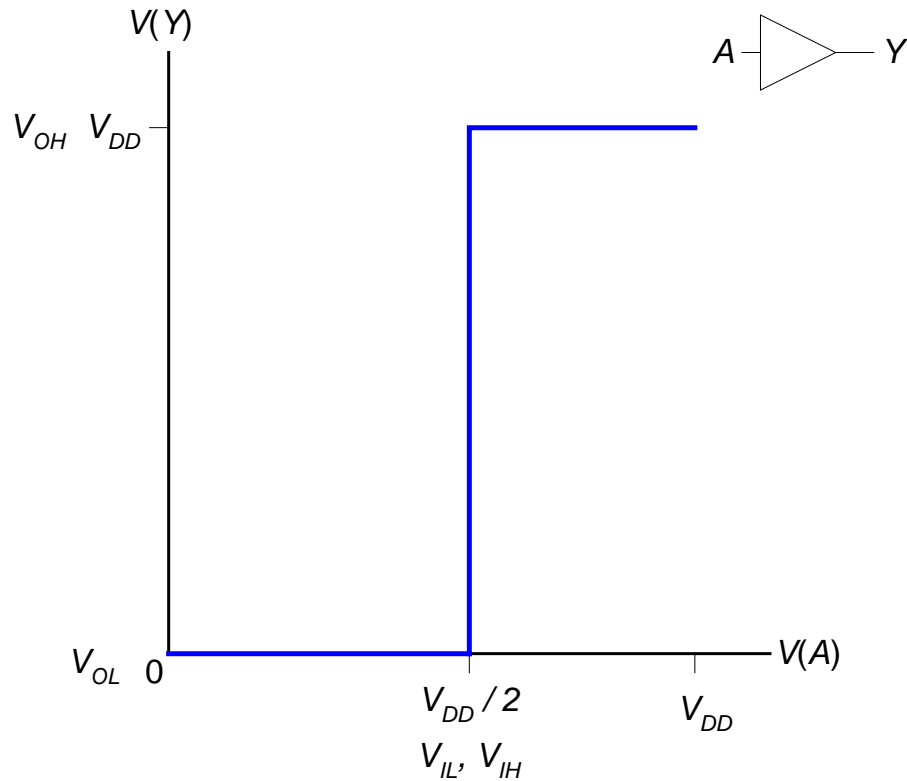


$$NM_H = V_{OH} - V_{IH}$$

$$NM_L = V_{IL} - V_{OL}$$

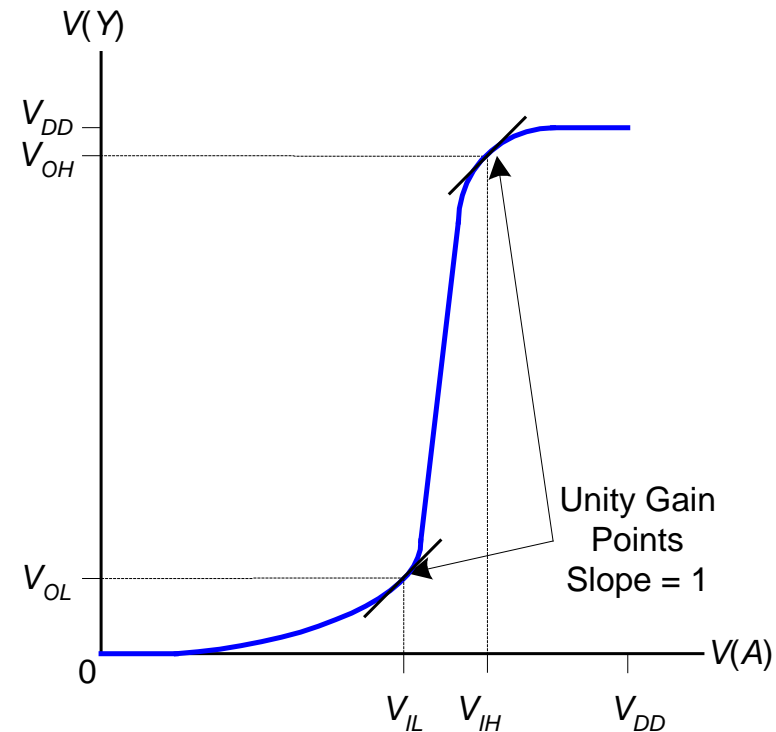
# Передавальна характеристика на постійному струмі

Ідеальний буфер:



$$NM_H = NM_L = V_{DD}/2$$

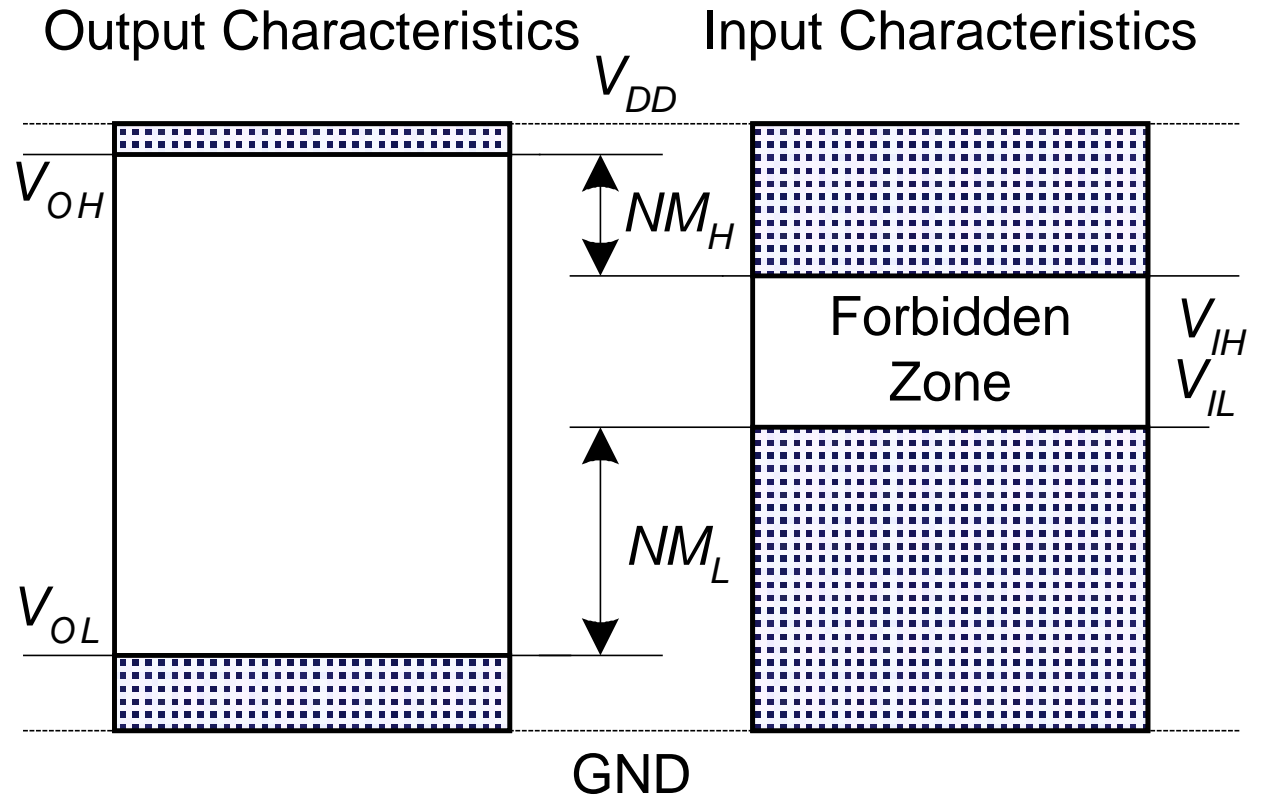
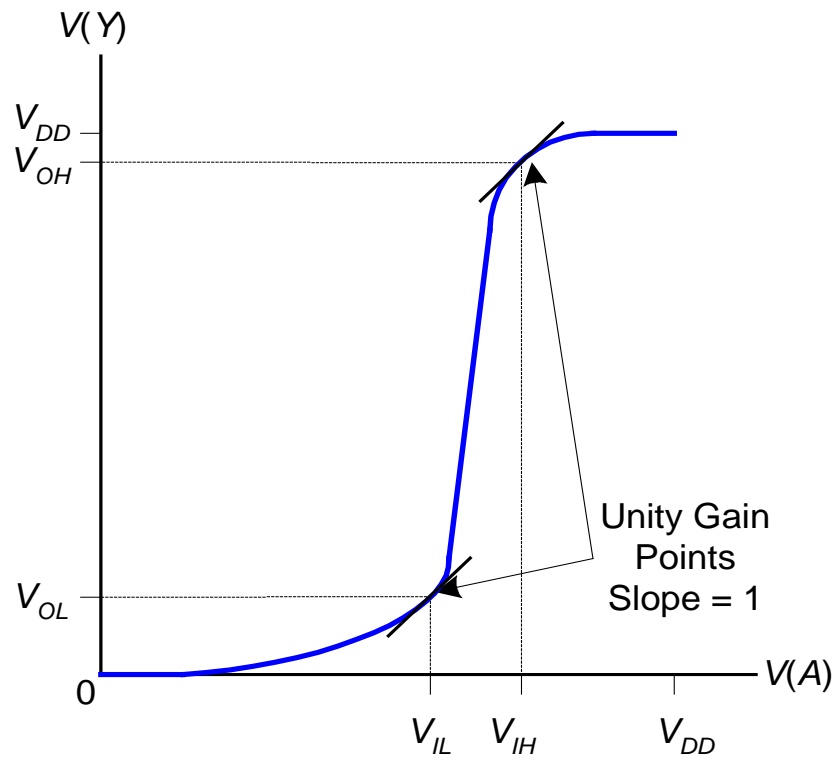
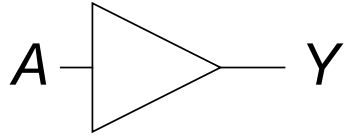
Реальний буфер:



$$NM_H, NM_L < V_{DD}/2$$



# Передавальна характеристика на постійному струмі



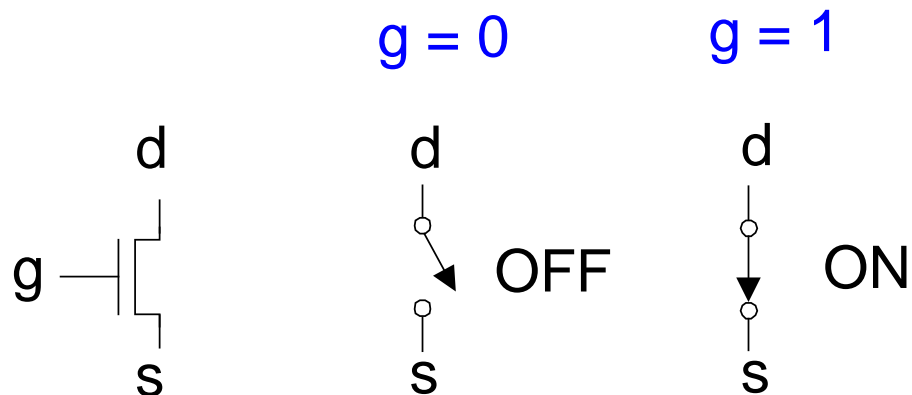
# Зміна напруги живлення $V_{DD}$

- В 1970 і 1980 роки,  $V_{DD} = 5\text{ В}$
- В наступні роки  $V_{DD}$  зменшується
  - Зменшується нагрів транзисторів і зменшується енергоспоживання
- 3.3 В, 2.5 В, 1.8 В, 1.5 В, 1.2 В, 1.0 В, ...
- З'єднання мікросхем з різними напругами живлення вимагає уваги

Логічні родини	$V_{DD}$	$V_{IL}$	$V_{IH}$	$V_{OL}$	$V_{OH}$
TTL	5 (4.75 - 5.25)	0.8	2.0	0.4	2.4
КМОН	5 (4.5 - 6)	1.35	3.15	0.33	3.84
LVTTL	3.3 (3 - 3.6)	0.8	2.0	0.4	2.4
LVCМOS	3.3 (3 - 3.6)	0.9	1.8	0.36	2.7

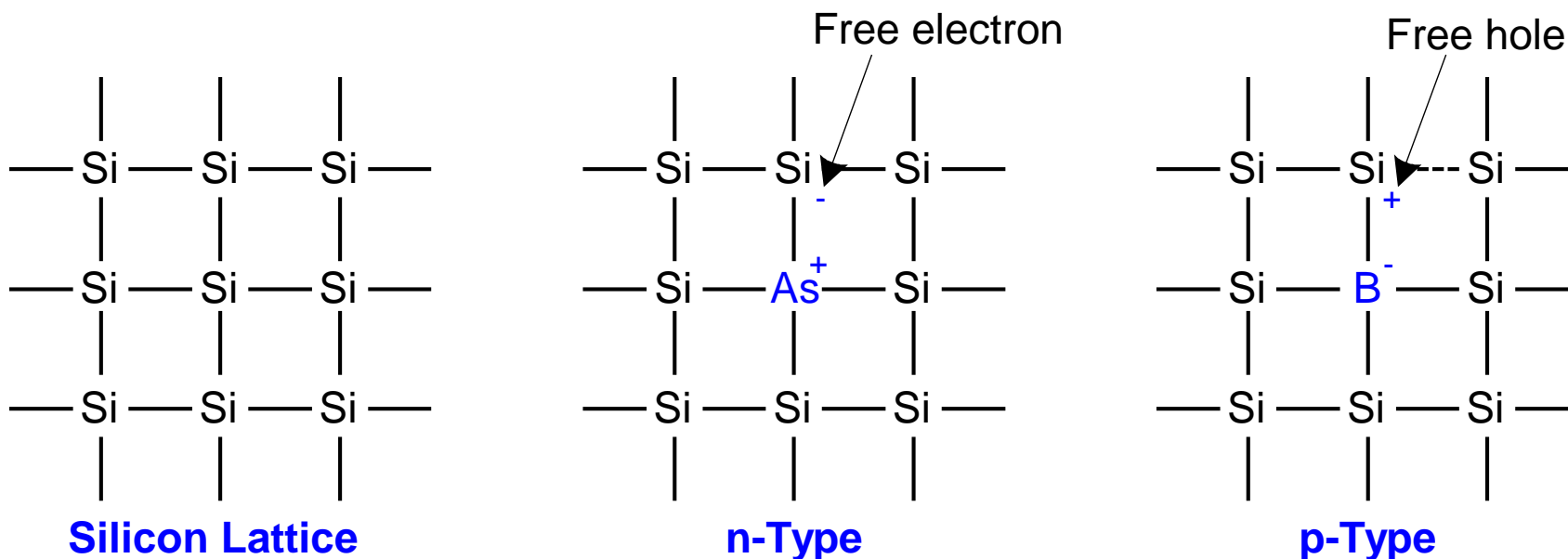
# Транзистори

- Логічні елементи складаються з транзисторів
- Трехходовий керований напругою вимикач
  - З'єднання двох входів залежить від напруги на третьому вході
  - d і s з'єднані (ON) коли g равно 1



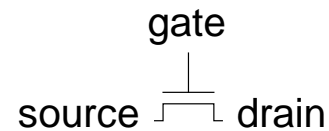
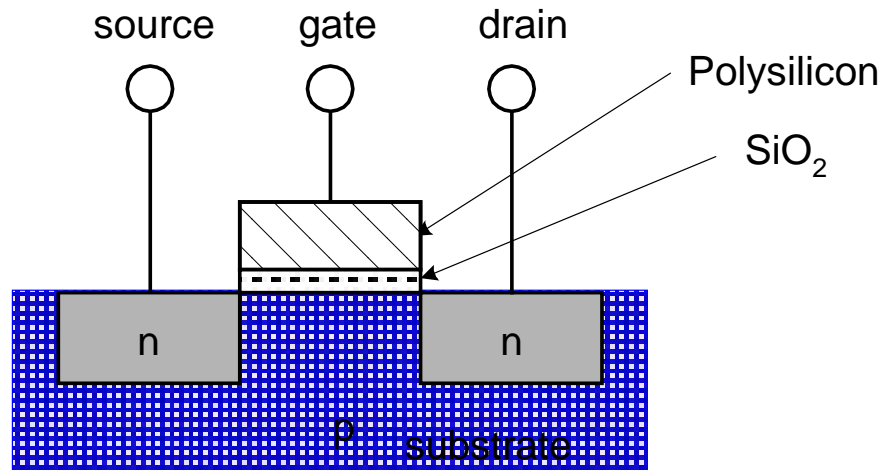
# Кремній

- Транзистори створюються з напівпровідникового матеріалу, кремнію
- Чистий кремній поганий провідник (вільні носії заряду відсутні)
- Легований кремній добрий провідник (є вільні носії заряду)
  - n-типу (вільні носії заряду негативні (*negative*), електрони)
  - p-типу (вільні носії заряду позитивні (*positive*), дірки)



# МОН транзистори

- **Метал-оксид-напівпровідникові (МОН) транзистори:**
  - Полікремнієвий (використовується як метал) затвор
  - Оксидний (діоксид кремнію) ізолятор
  - Легований **кремній**

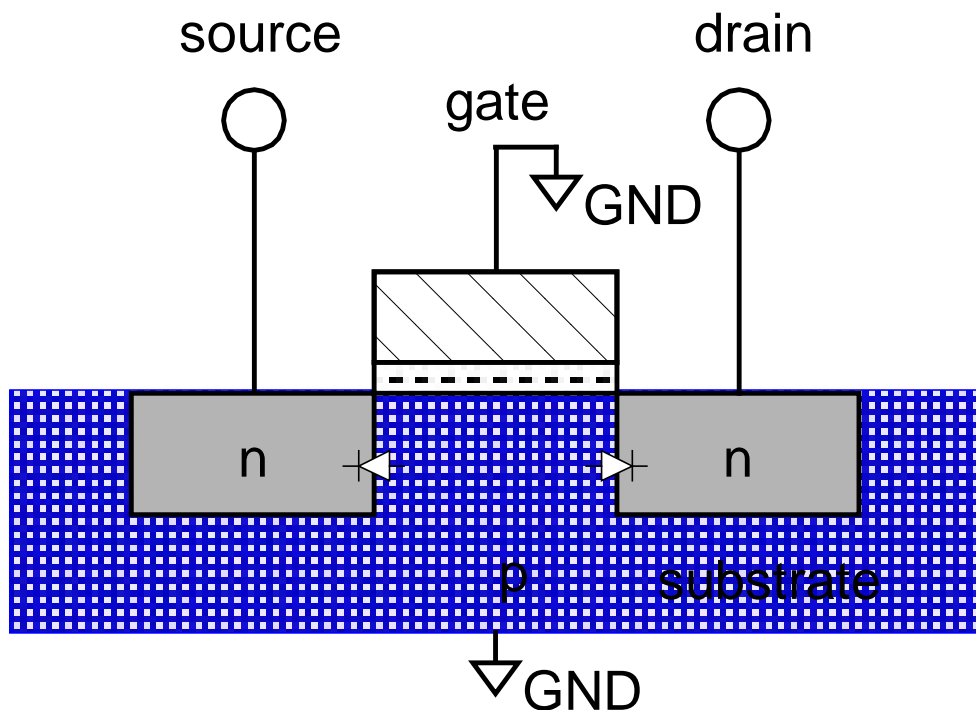


nMOS

# Транзистори: n-MOS

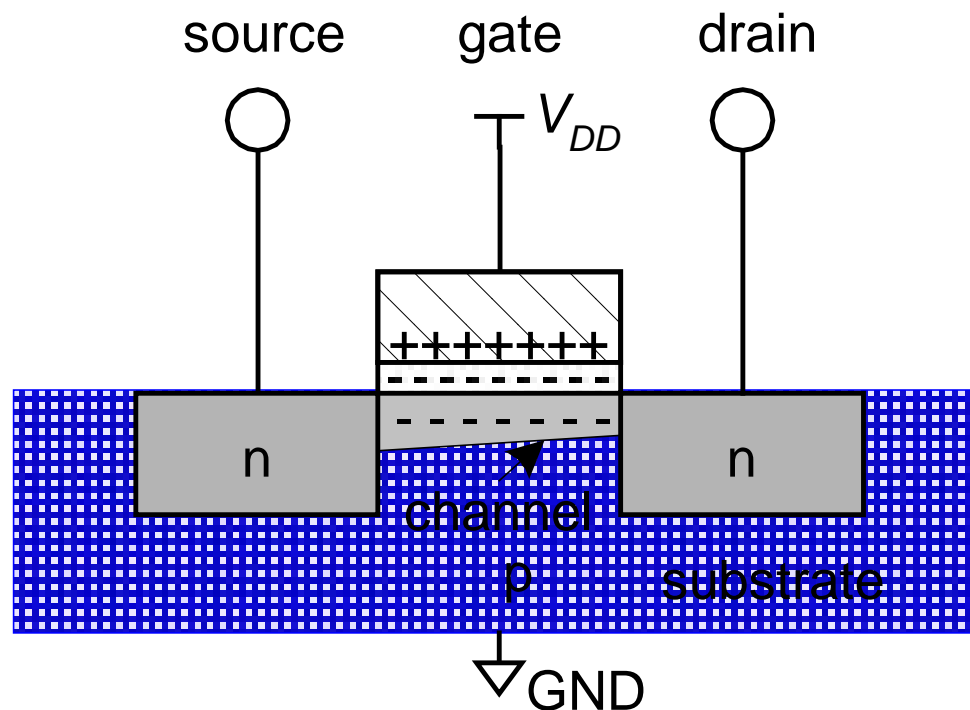
Gate = 0

OFF (виток і стік не з'єднані)



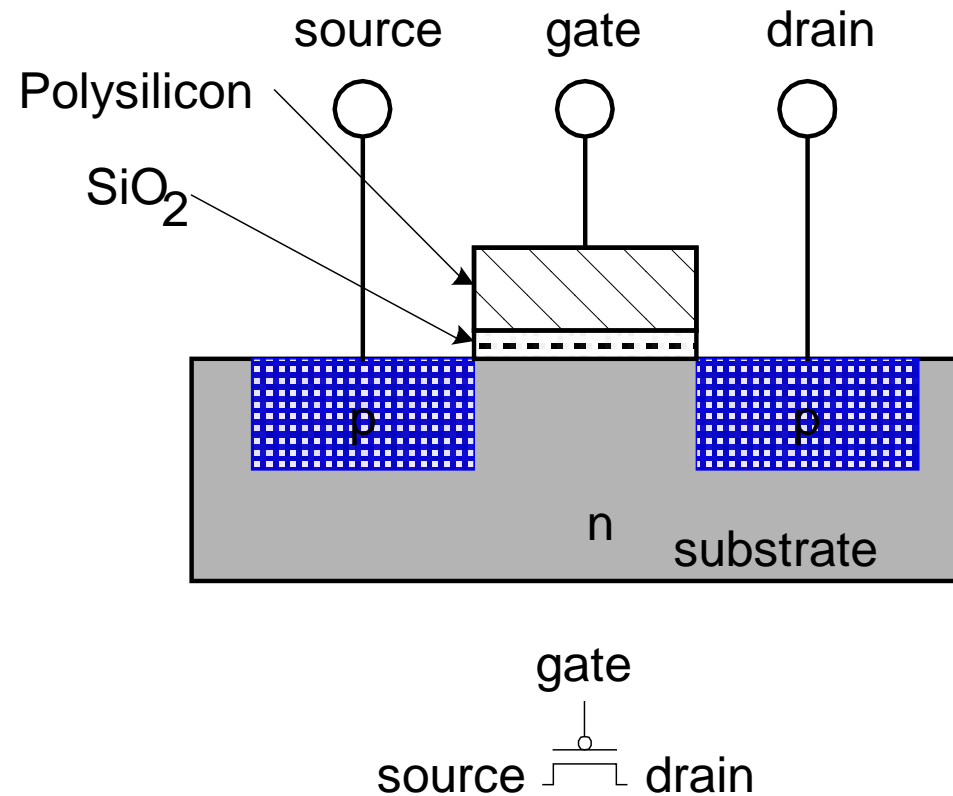
Gate = 1

ON (виток і стік з'єднані)



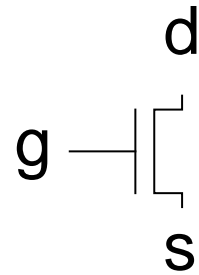
# Транзистори: р-МОН

- р-МОН транзистор працює протилежним чином
  - ON, коли Gate = 0
  - OFF, коли Gate = 1

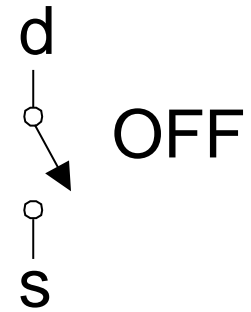


# Робота транзистора

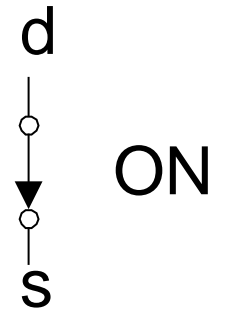
nMOS



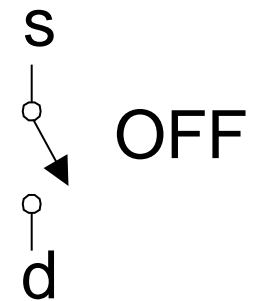
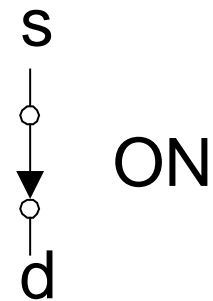
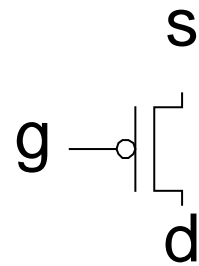
$g = 0$



$g = 1$



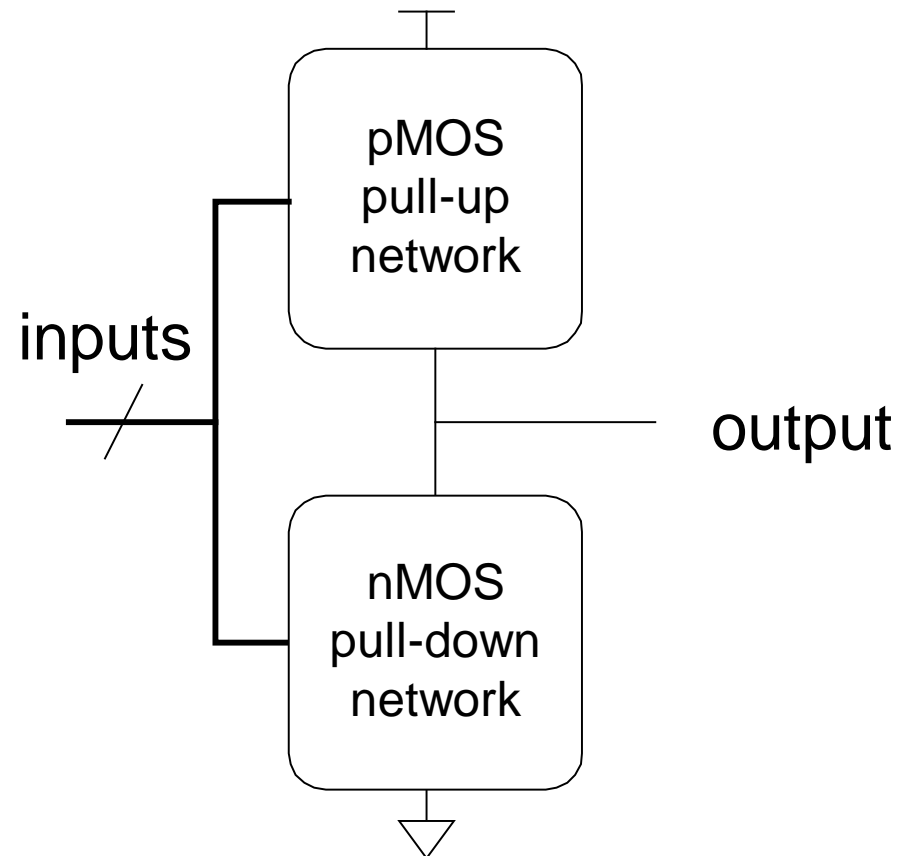
pMOS





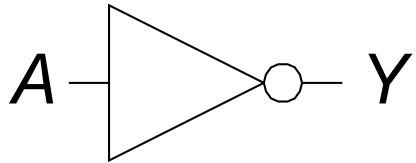
# Робота транзистора

- **n-MOH:** Добре передає 0, так як витік з'єднаний з GND
- **p-MOH:** Добре передає 1, так як витік з'єднаний з  $V_{DD}$



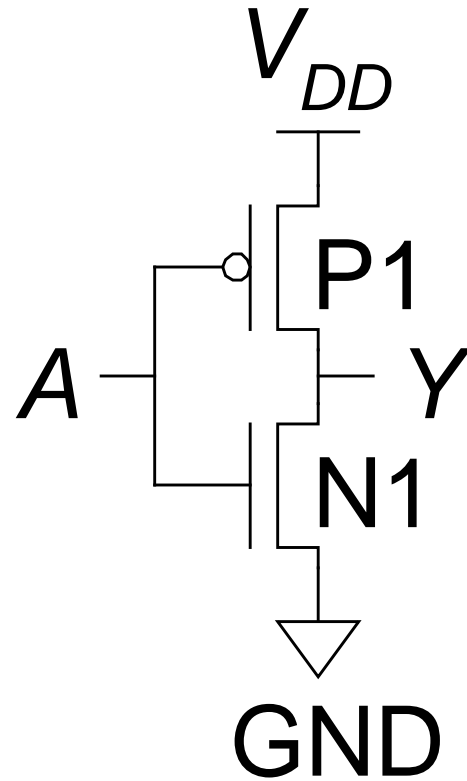
# Логічні елементи КМОН: Логічний елемент НЕ:

**NOT**



$$Y = \overline{A}$$

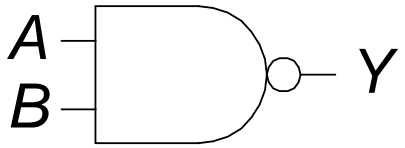
A	Y
0	1
1	0



A	P1	N1	Y
0	Вкл.	Викл.	1
1	Викл.	Вкл.	0

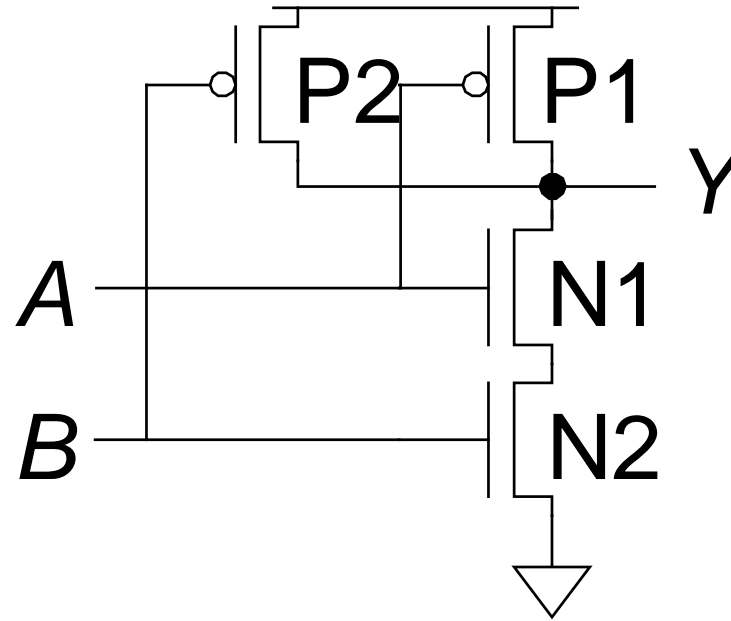
# Логічні елементи КМОН: Логічний елемент І-НЕ:

**NAND**



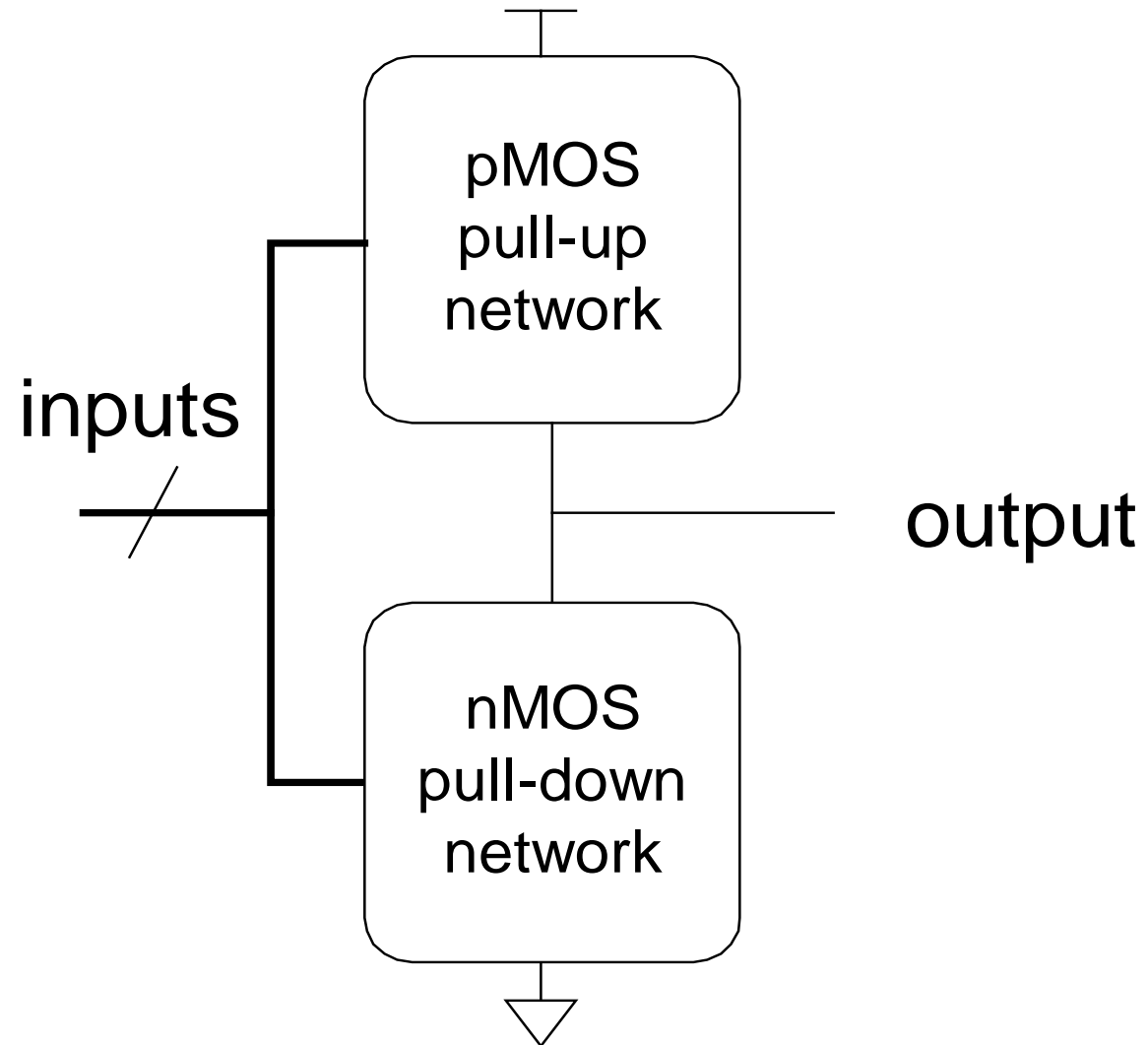
$$Y = \overline{AB}$$

A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



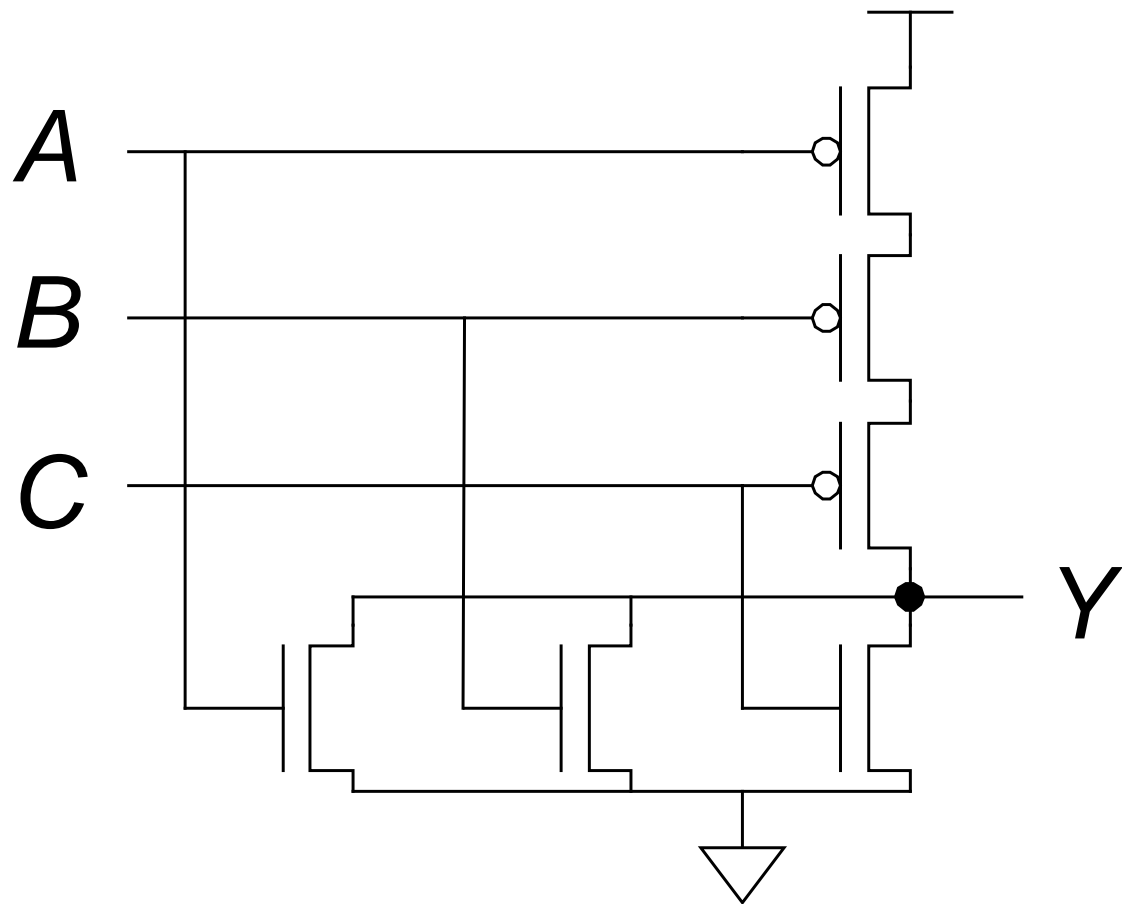
A	B	P1	P2	N1	N2	Y
0	0	Вкл.	Вкл.	Выкл.	Выкл.	1
0	1	Вкл.	Выкл.	Выкл.	Вкл.	1
1	0	Выкл.	Вкл.	Вкл.	Выкл.	1
1	1	Выкл.	Выкл.	Вкл.	Вкл.	0

# Структура элементу КМОН

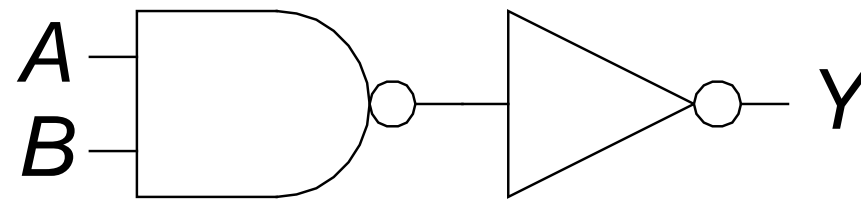


# Багатовходові логічні елементи КМОН

АБО-НЕ з трьома входами

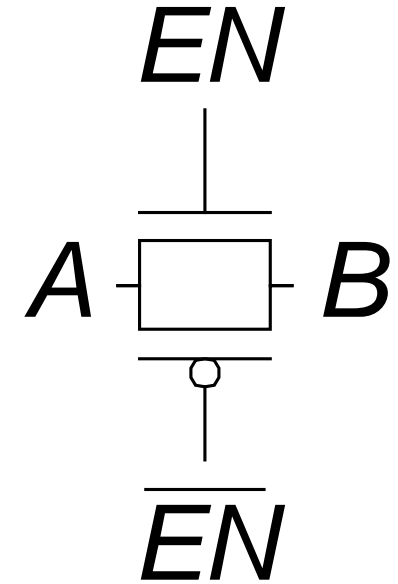


І з двома входами



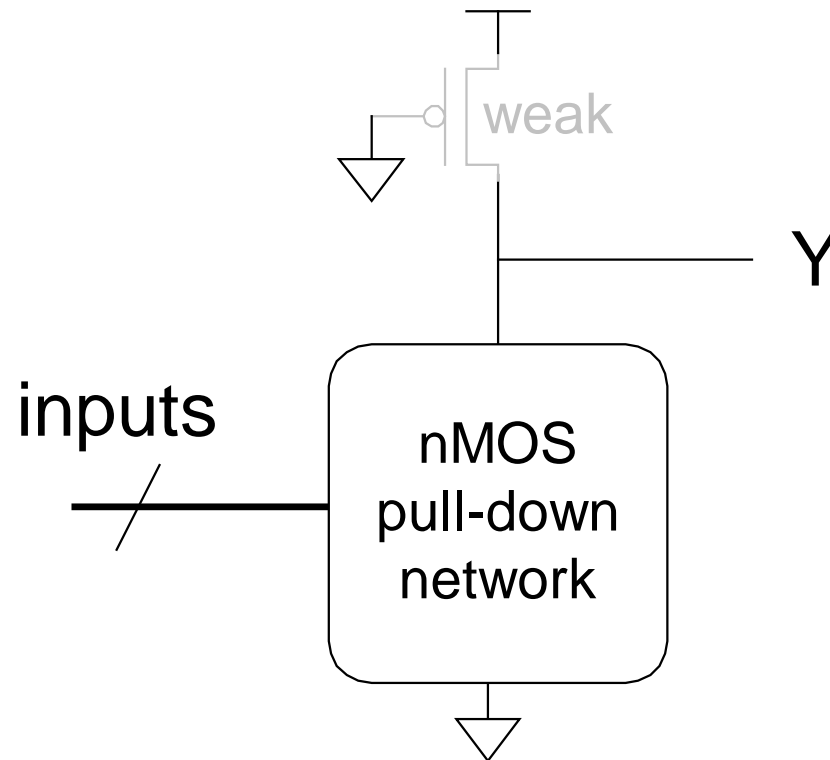
# Передавальний логічний елемент

- n-МОН погано передає 1
- р-МОН погано передає 0
- Передавальний логічний елемент кращий вимикач
- добре передає і 0 і 1
- Коли  $EN = 1$ , вимикач замкнутий (ON):
  - $EN = 1$  і  $A$  з'єднано з  $B$
- Коли  $EN = 0$ , вимикач розімкнутий (OFF):
  - $A$  і  $B$  не з'єднані



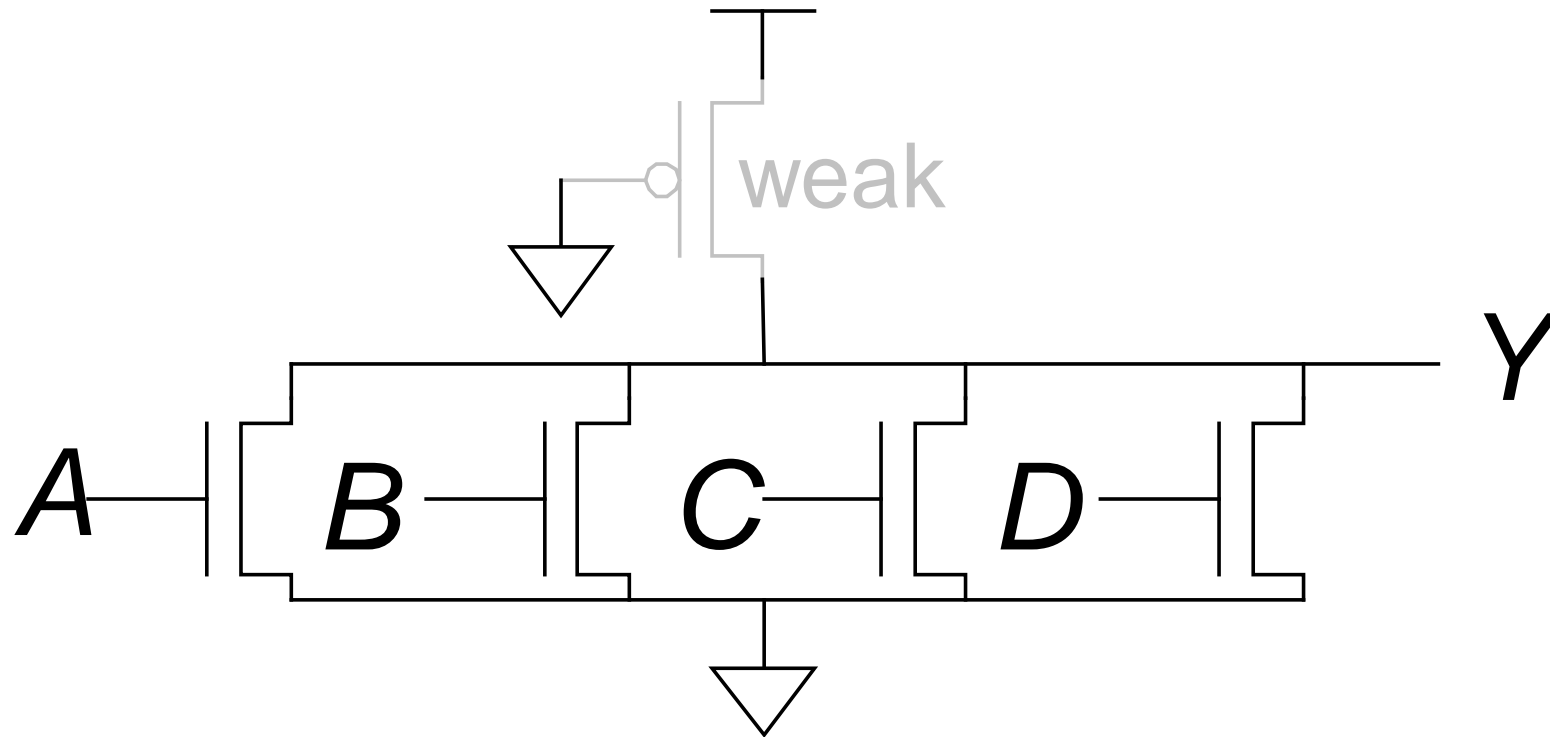
# Елементи псевдо-n-МОН

- Замінити підтягуючий ланцюг слабким завжди включеним р-МОН транзистором
- р-МОН транзистор: підтягує вихід до високої напруги, тільки якщо n-МОН ланцюг не тягне його до низької напруги



# Приклад елементу псевдо-n-MOH

Псевдо-n-MOH елемент **NOR4**





# Енергоспоживання

- Потужність = Споживання енергії в одиницю часу
- Динамічна споживага потужність
- Статична споживана потужність

# Динамічна споживана потужність

- Потужність іде на зарядку ємностей затворів транзисторів
  - Для зарядки конденсатора ємністю  $C$  до напруги  $V_{DD}$  необхідна енергія  $CV_{DD}^2$
  - Струм перемикається з частотою  $f$ : транзистор перемикається (від 0 до 1 або навпаки) з такою ж частотою
  - Конденсатор заряджається  $f/2$  разів за секунду (розрядження з 1 в 0 не потребує енергії)
- Динамічна споживана потужність:

$$P_{dynamic} = \frac{1}{2}CV_{DD}^2f$$

# Статична споживана потужність

- Споживана потужність при відсутності перемикання елементів
- Зумовлена *струмами спокою (струми витікання)*,  $I_{DD}$
- Статична споживана потужність:

$$P_{static} = I_{DD} V_{DD}$$

## Приклад оцінки

- Оцінити потужність, яку споживає бездротовий переносний комп'ютер
  - $V_{DD} = 1.2 \text{ В}$
  - $C = 20 \text{ нФ}$
  - $f = 1 \text{ ГГц}$
  - $I_{DD} = 20 \text{ мА}$

$$P = \frac{1}{2}CV_{DD}^2f + I_{DD}V_{DD}$$

$$= \frac{1}{2}(20 \text{ нФ})(1.2 \text{ В})^2(1 \text{ ГГц}) + (20 \text{ мА})(1.2 \text{ В}) = \mathbf{14.4 \text{ W}}$$