

# Логические элементы

# Транзистор — основа устройства компьютера

## Число транзисторов в процессорах

- Intel Pentium II: 7 млн.
- ARM Cortex A9: 15 млн.
- Core i7 (4 ядра) 731 млн.

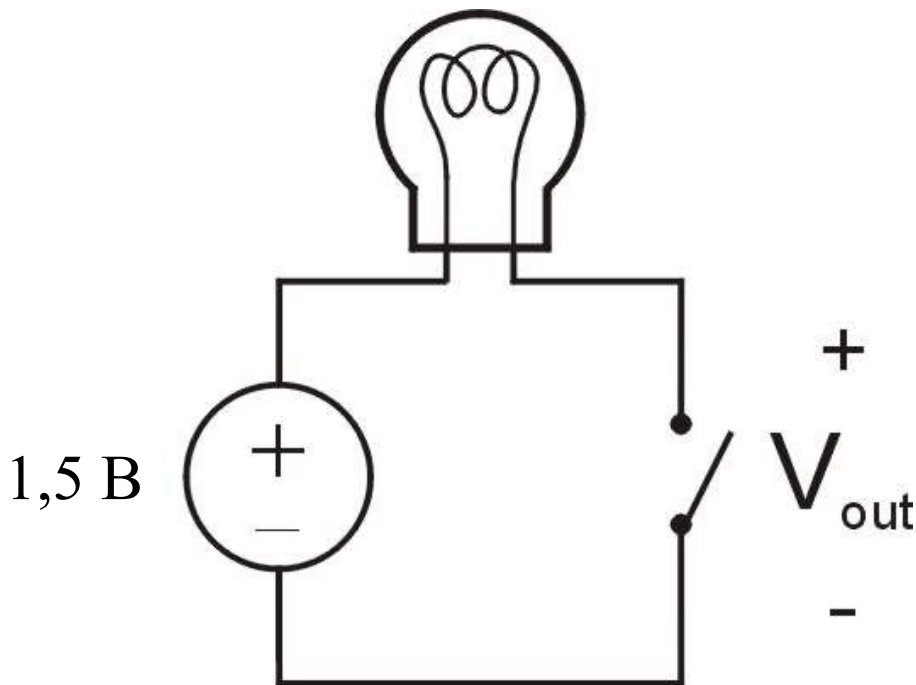
**Логически транзистор представляет собой выключатель**

**Их комбинации позволяют строить логические элементы – И, ИЛИ, НЕ, ...**

**На логических элементах строятся устройства – сумматор, мультиплексор, декодер, регистр**

**На логических устройствах строится процессор**

## Цепь с выключателем



### Выключатель **разомкнут**:

- тока нет
- лампа **не горит**
- $V_{out}$  **+1,5 В**

### Выключатель **замкнут**:

- ток идёт
- лампа **горит**
  - $V_{out}$  **0 В**

Такая цепь практически отражает два состояния

# МОП-транзистор

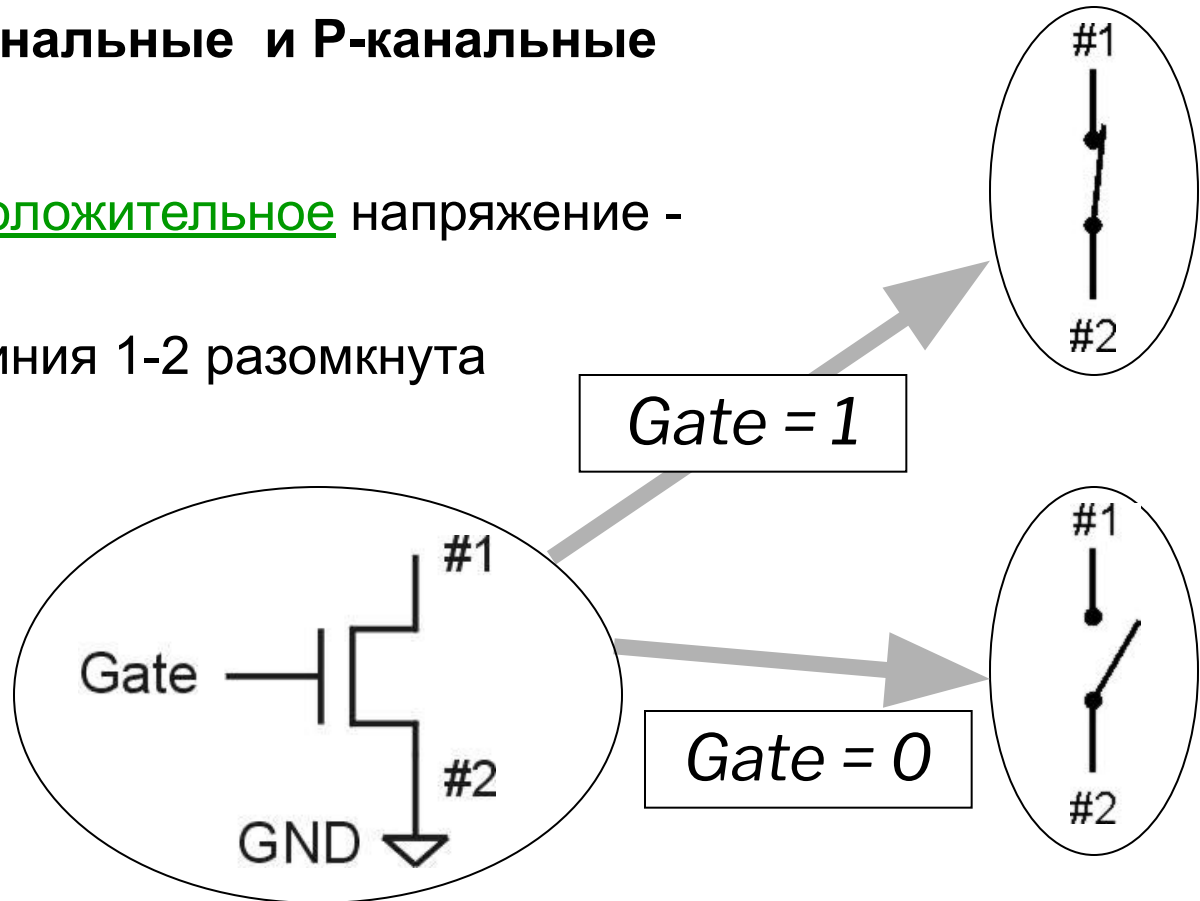
МОП (MOS) = **Металл-окисел-полупроводник (Metal Oxide Semiconductor)**

- два типа: N-канальные и P-канальные

## N-канальный

на затворе (Gate) положительное напряжение -  
линия 1-2 замкнута

на затворе 0 В — линия 1-2 разомкнута

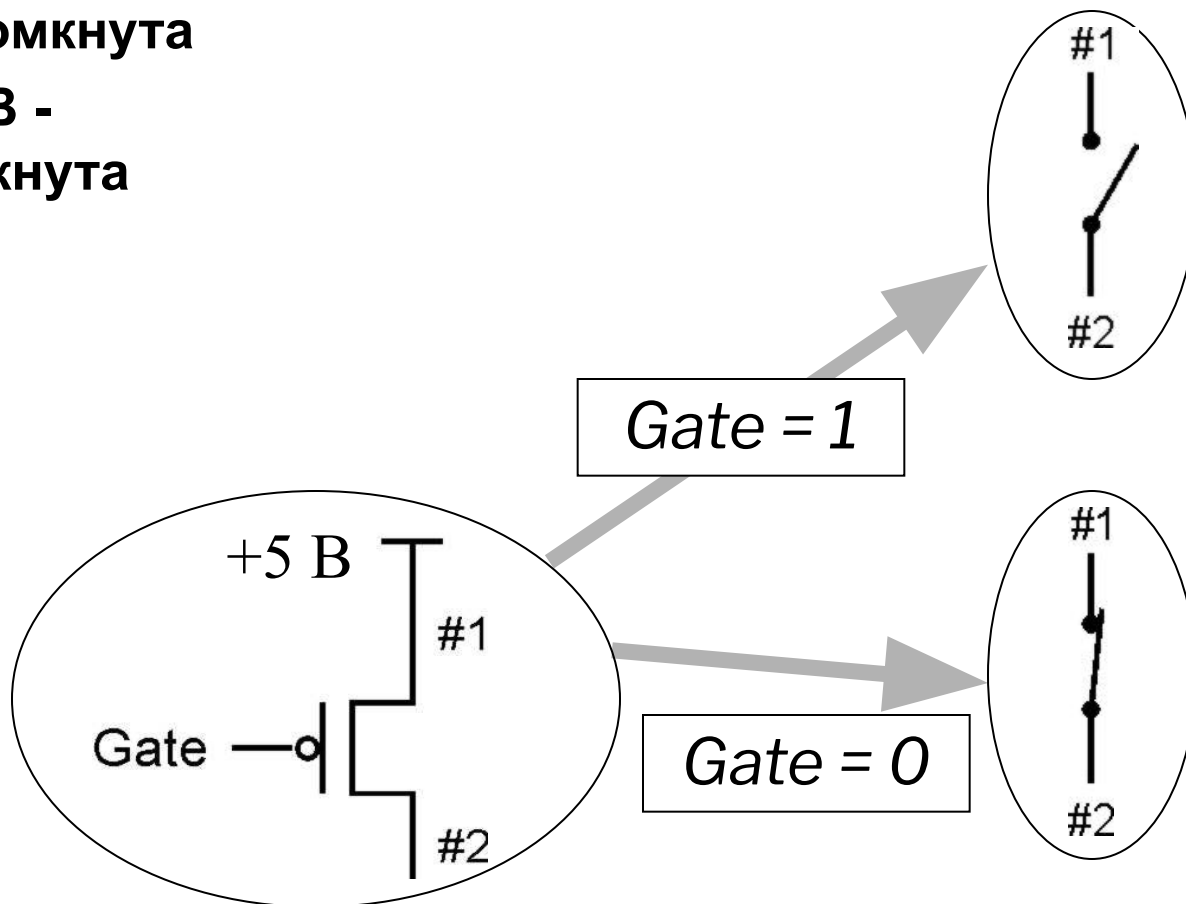


вывод 2 подключен  
к земле (0 В).

# P-канальный МОП-транзистор

## P-канальный дополняет N-канальный

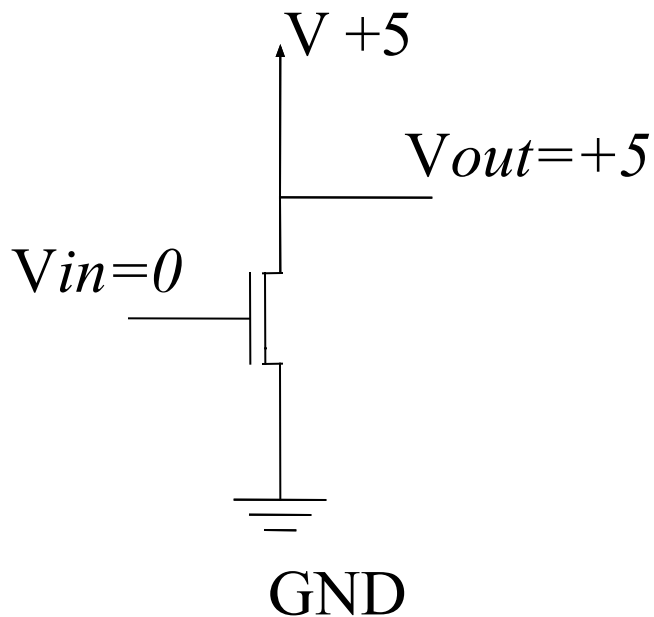
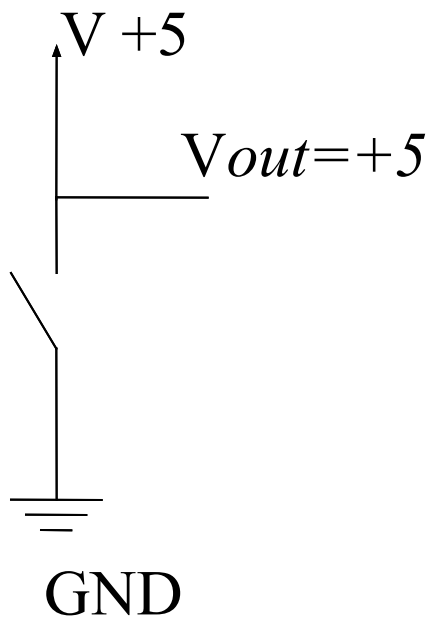
- на затворе положительное напряжение - линия 1-2 разомкнута
- на затворе 0 В - линия 1-2 замкнута



Вывод 1 подключен  
к питанию +5 В

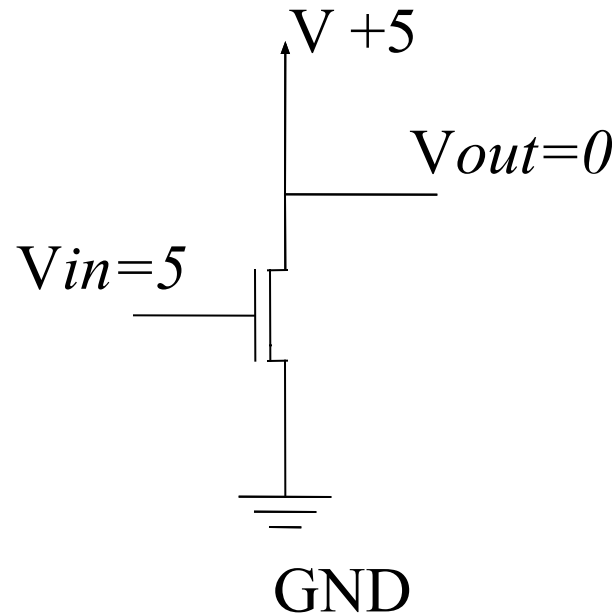
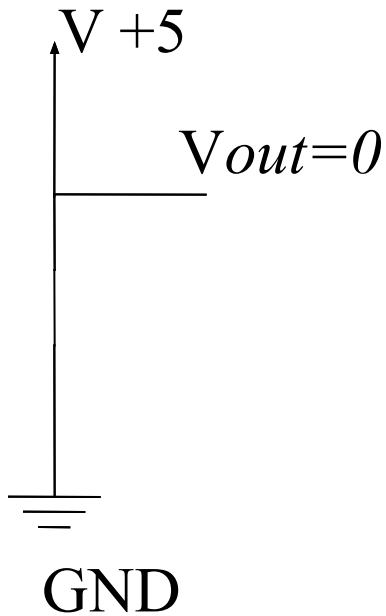
# Логические вентили

Как использовать МОП-транзисторы для реализации логической операции НЕ



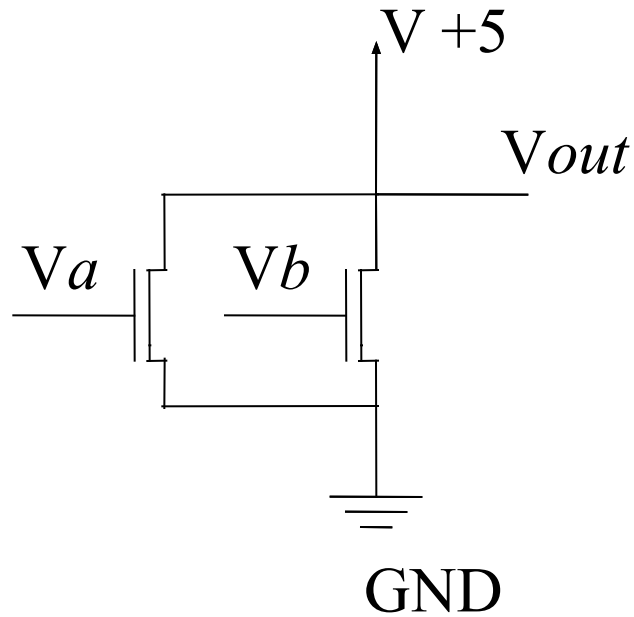
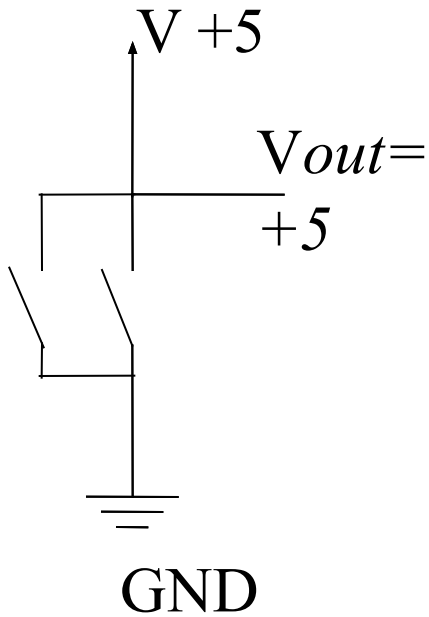
# Логические вентили

Как использовать МОП-транзисторы для реализации логической операции НЕ



# Логические вентили МОП

## ИЛИ-НЕ $!(a \vee b)$

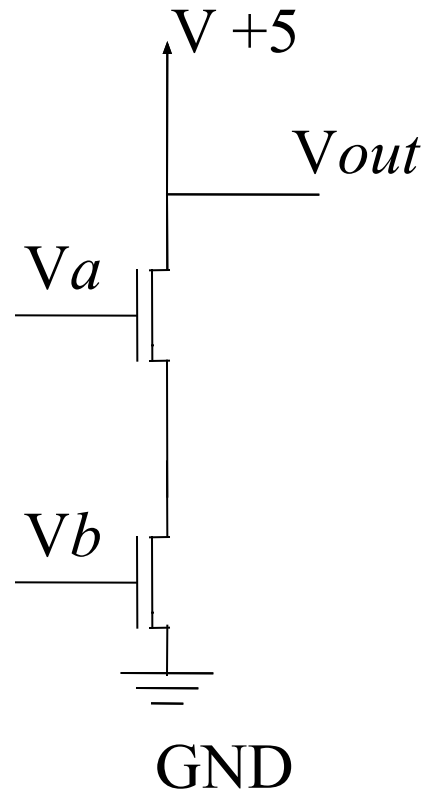
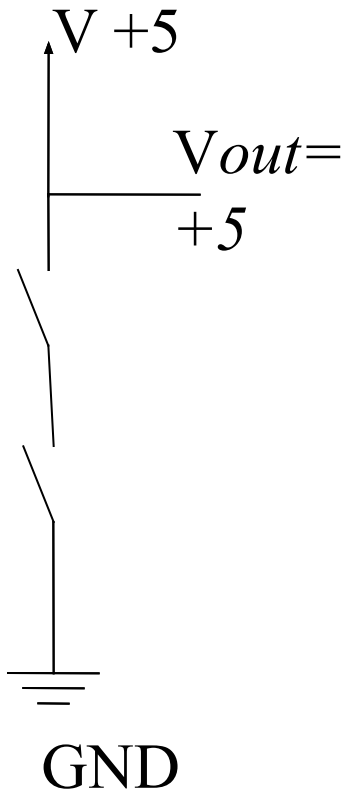


a	b	$!(a \vee b)$
0	0	1
0	1	0
1	0	0
1	1	0



# Логические вентили МОП

## И-НЕ $!(a\&b)$



a	b	$!(a\&b)$
0	0	1
0	1	1
1	0	1
1	1	0

# Инвертор (НЕ) на технологии КМОП

Устраняет проблему тока на землю

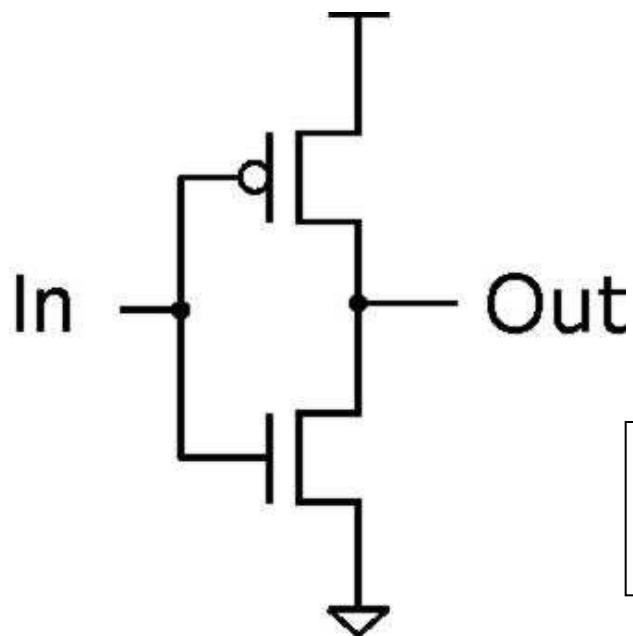
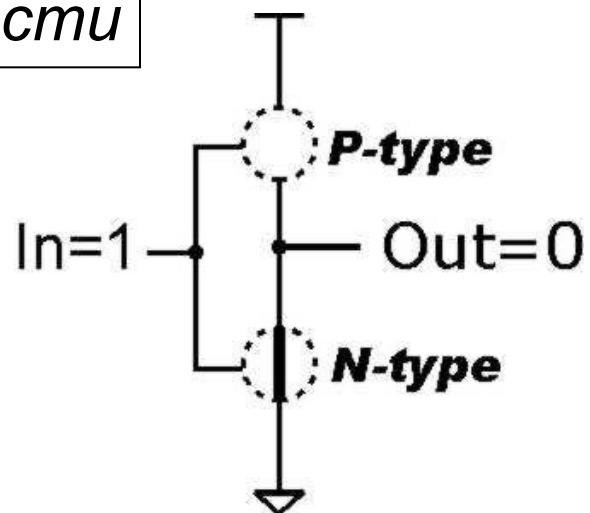
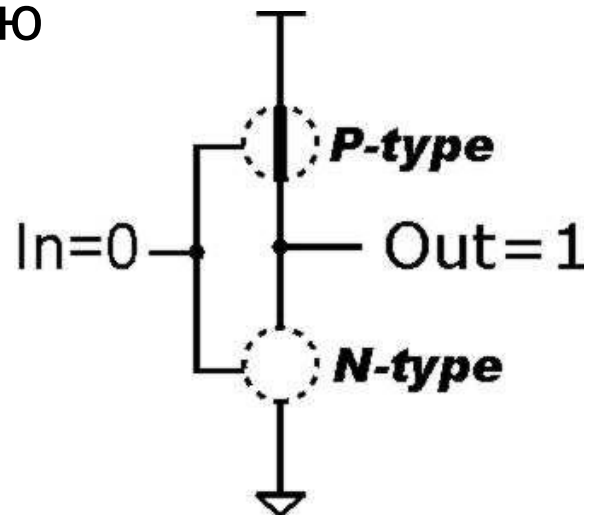


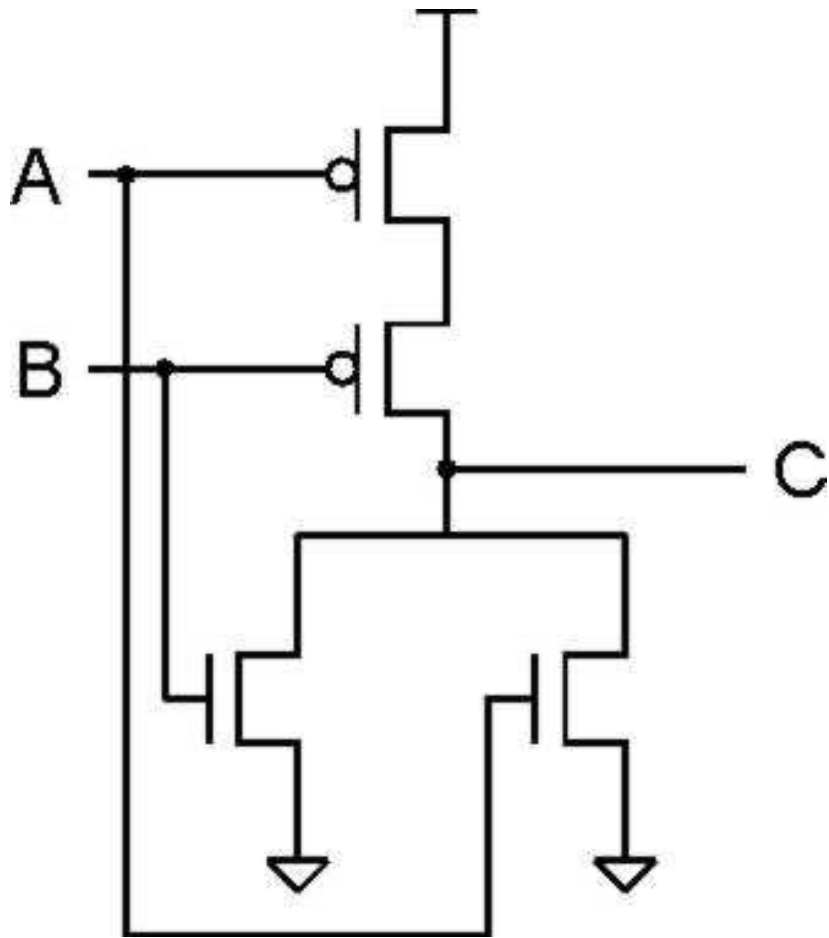
таблица истинности

In	Out	In	Out
0 В	5 В	0	1
5 В	0 В	1	0

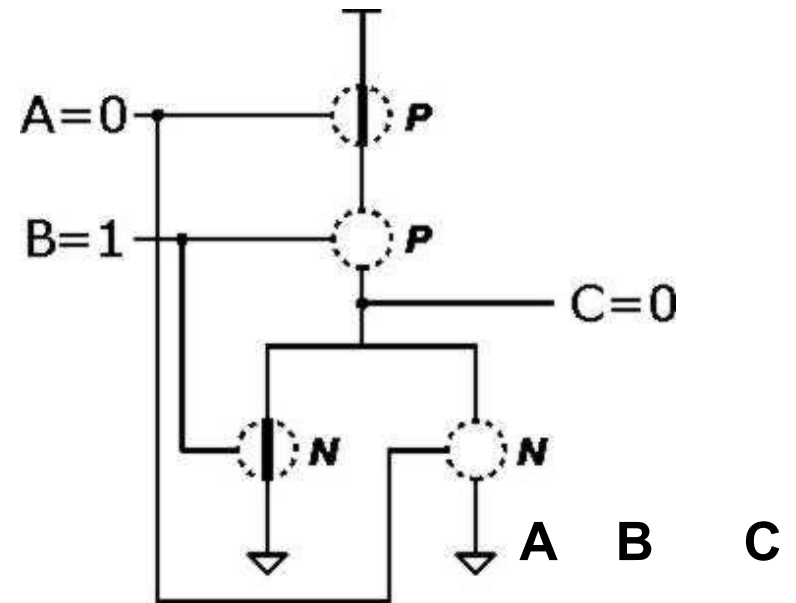


10

# Вентиль ИЛИ-НЕ !(AVB)

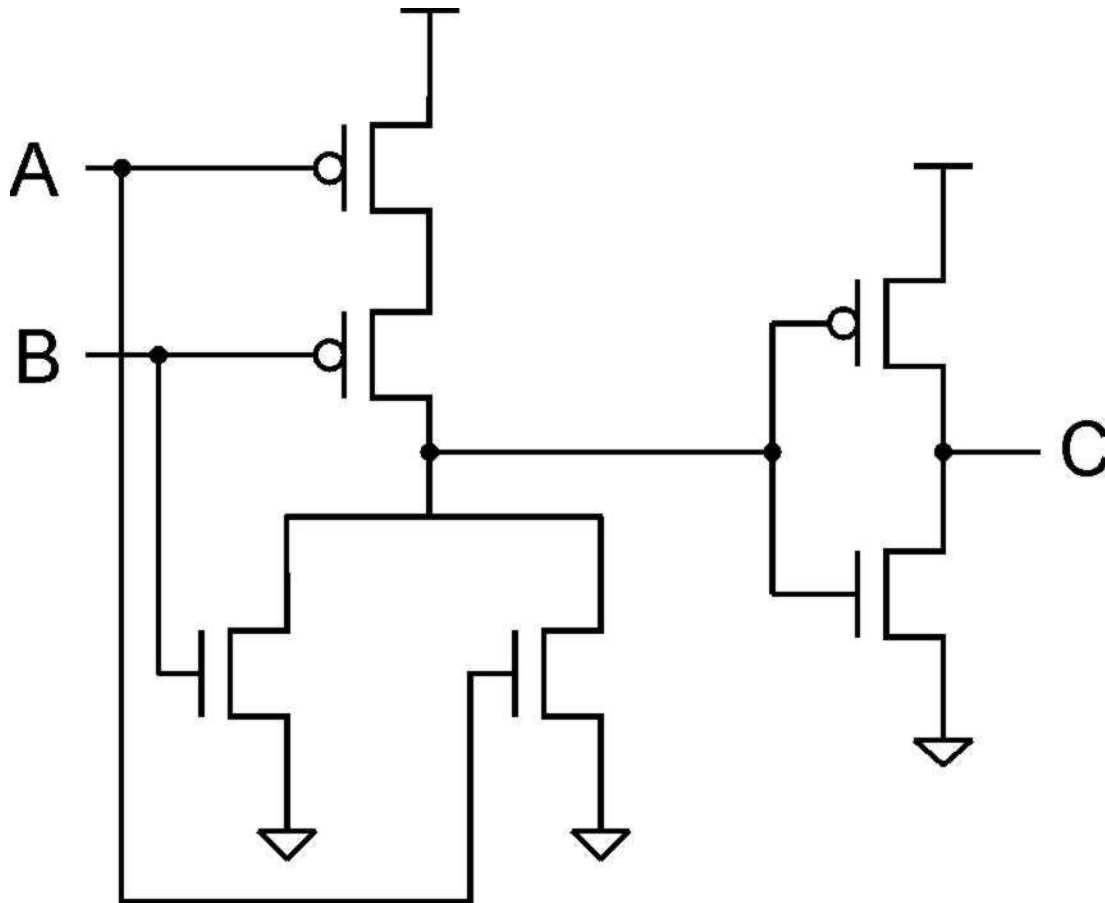


Сверху последовательная структура,  
снизу - параллельная



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

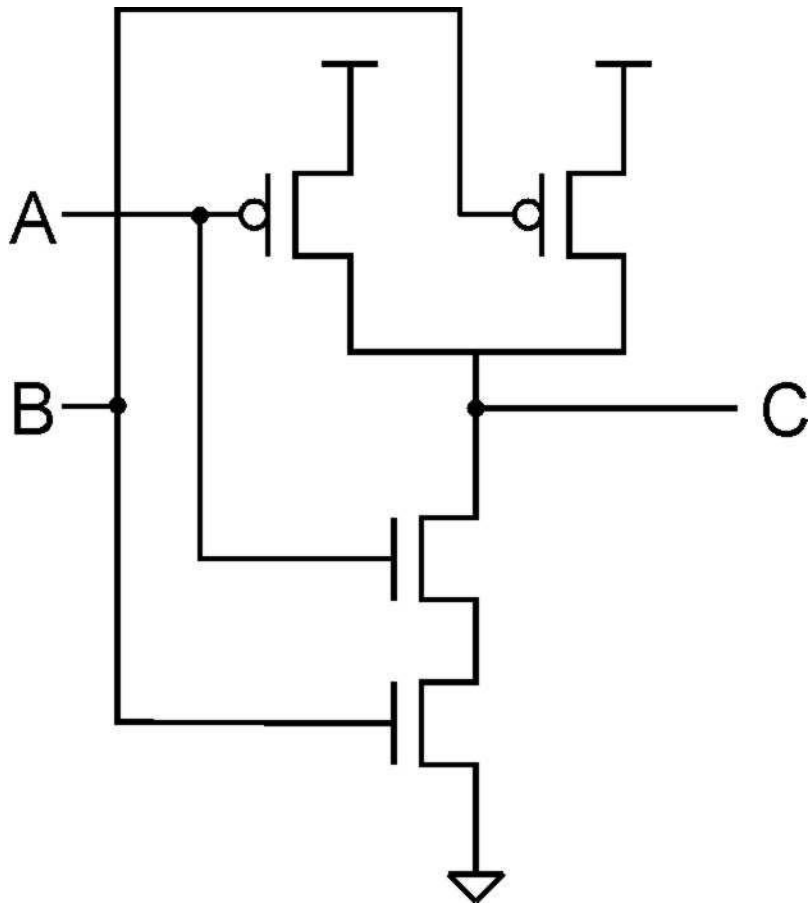
# Вентиль ИЛИ



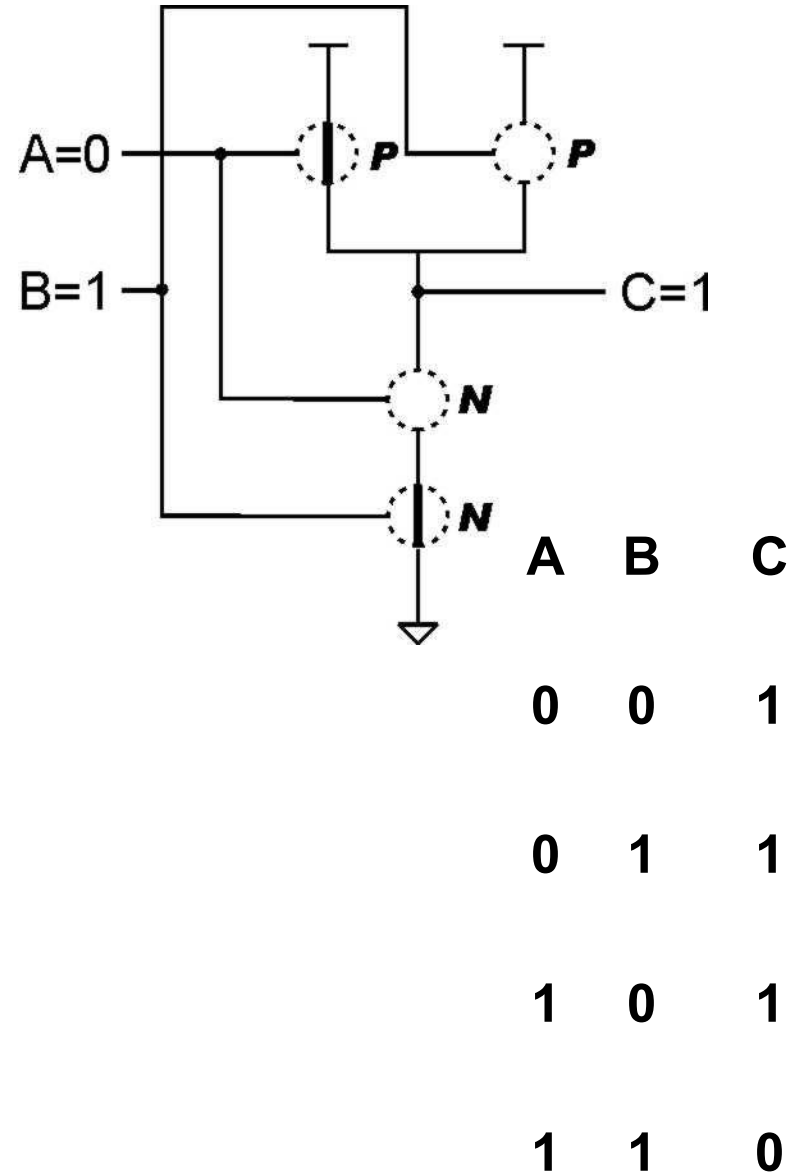
A	B	C
0	0	0
0	1	1
1	0	1
1	1	1

*ИЛИ-НЕ с инвертером*

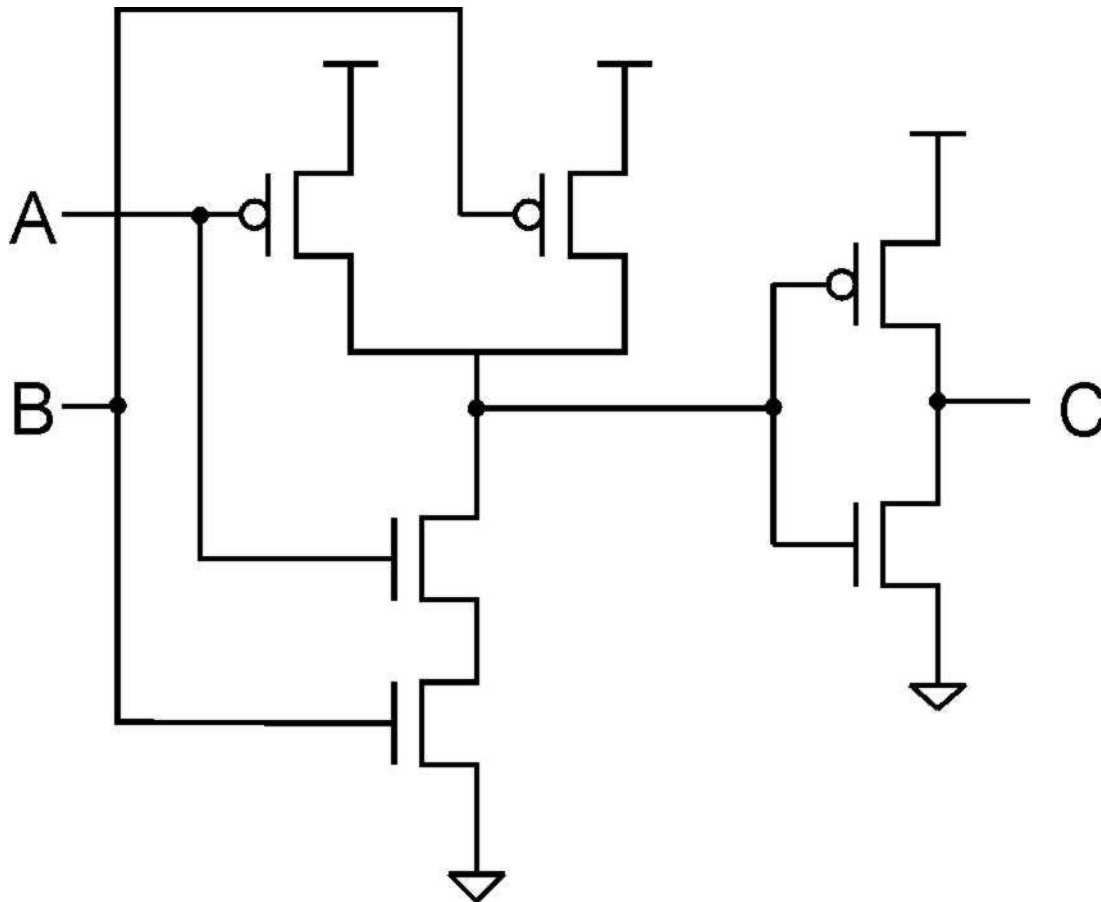
# Вентиль И-НЕ (NAND)



Сверху параллельная структура  
снизу - последовательная



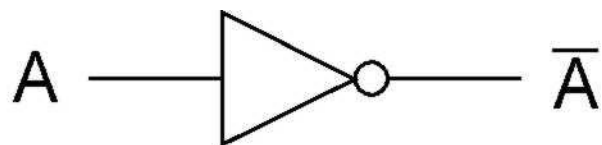
# Вентиль И



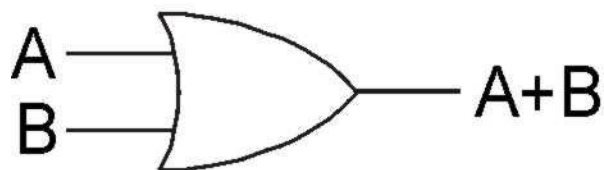
A	B	C
0	0	0
0	1	0
1	0	0
1	1	1

*И-НЕ с инвертером*

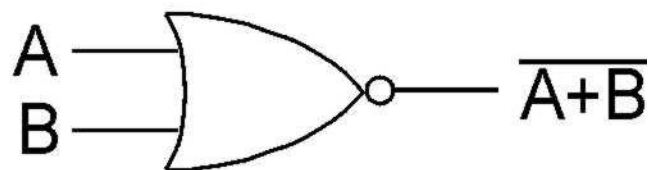
## Обозначение вентилей



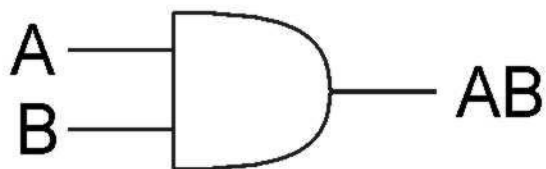
*NOT*



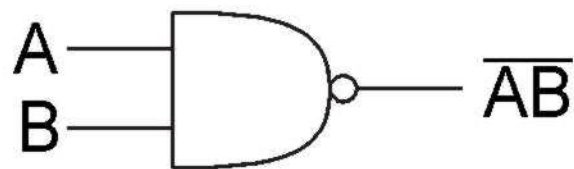
*OR*



*NOR*

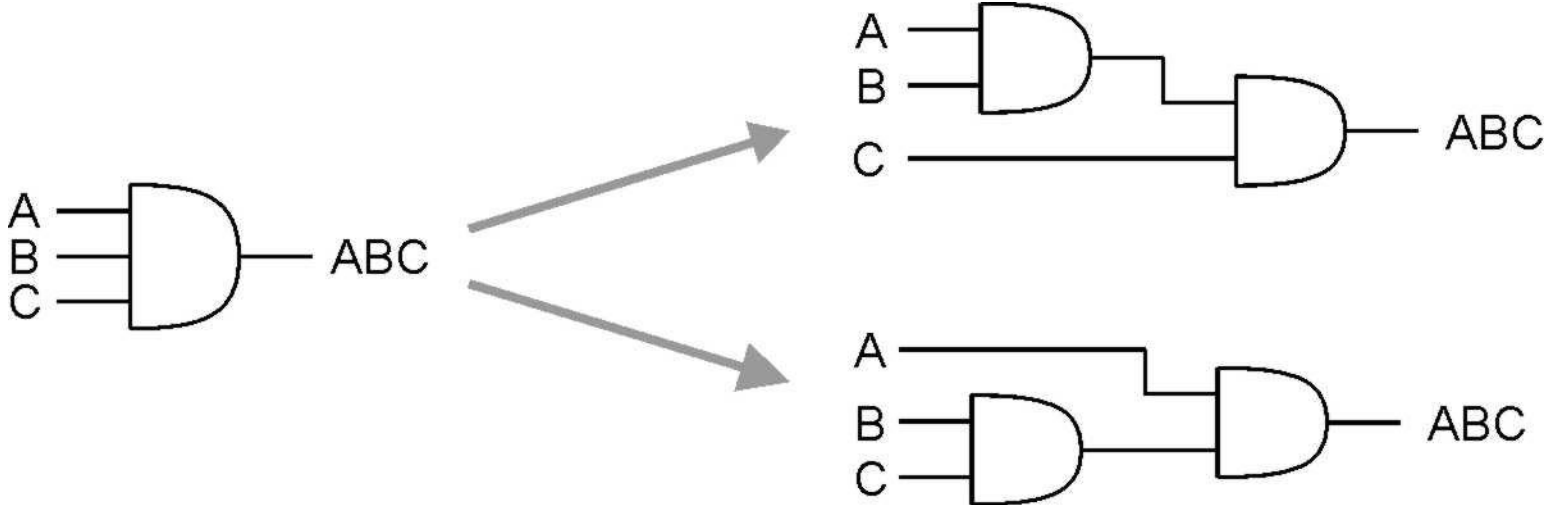


*AND*



*NAND*

# Вентили с несколькими входами



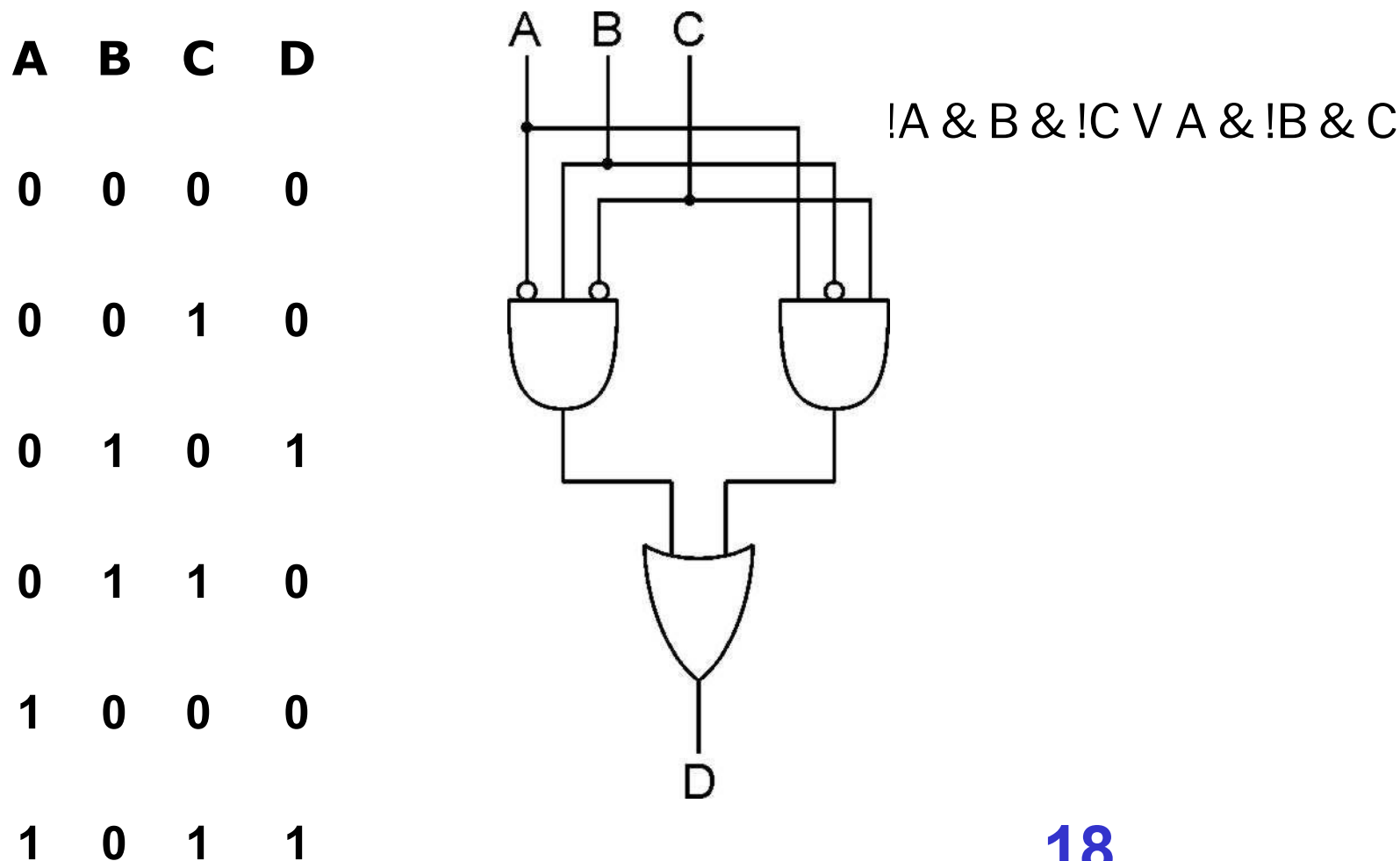


## Упражнение 1

Реализуем 3-операндный вентиль ИЛИ-НЕ на КМОП

## Полный набор логических функций

Легко видеть (тм), что имеемый набор вентиляей позволяет реализовать любую таблицу истинности



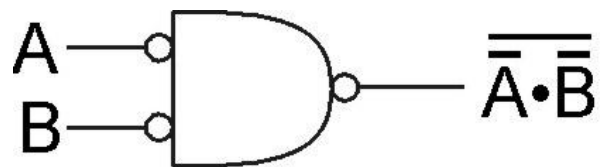
## Упражнение 2

Реализуем следующую таблицу истинности

<b>A</b>	<b>B</b>	<b>C</b>
0	0	0
0	1	1
1	0	1
1	1	0

# Закон Де-Моргана

Вентиль И превращается в ИЛИ  
инвертированием входов и выходов



<b>A</b>	<b>B</b>	$\overline{A}$	$\overline{B}$	$\overline{A} \cdot \overline{B}$	$\overline{\overline{A} \cdot \overline{B}}$
0	0	1	1	1	0
0	1	1	0	0	1
1	0	0	1	0	1
1	1	0	0	0	1

## Итоги

**МОП-транзисторы реализуют логические функции как переключатели**

- **N-канальные:** подключаются к земле, запитываются (логической 1) чтобы сбросить напряжение до 0
- **P-канальные:** подключаются к +V, запитываются (логическим 0) чтобы поднять напряжение до 1

**Основные вентили: НЕ, ИЛИ-НЕ, И-НЕ**

- реализуют логические функции И, ИЛИ, НЕ и т.п.

# Реализация функций на вентилях

Рассмотренные примеры реализации логических функций — комбинаторные схемы

*Комбинаторная логическая схема*

- выход зависит только от текущих входных данных
- нет памяти состояний

*Последовательная логическая схема*

- выход зависит от цепочки входных комбинаций (прошлых и настоящей)
- хранит информацию о предыдущих входных данных (состояние)

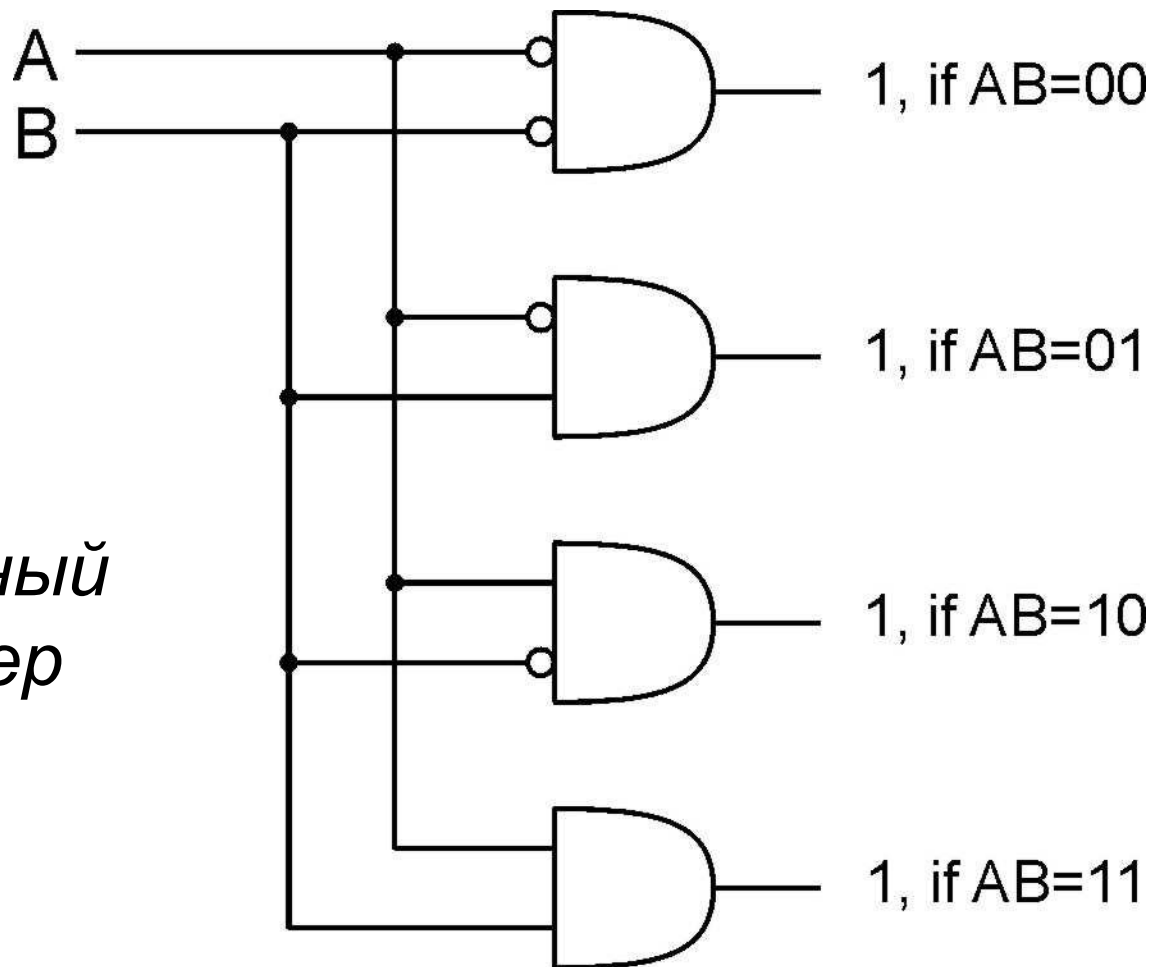
Рассмотрим примеры комбинаторных схем, затем перейдём к последовательным, хранящим информацию о состояниях

# Декодер

$n$  ВХОДОВ,  $2^n$  ВЫХОДОВ

- для каждого варианта ВХОДА один и только один ВЫХОД принимает 1

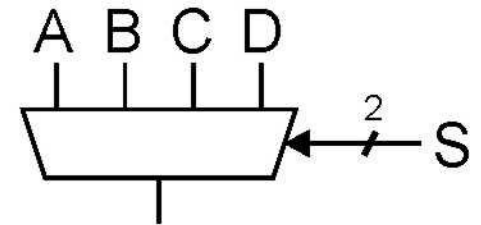
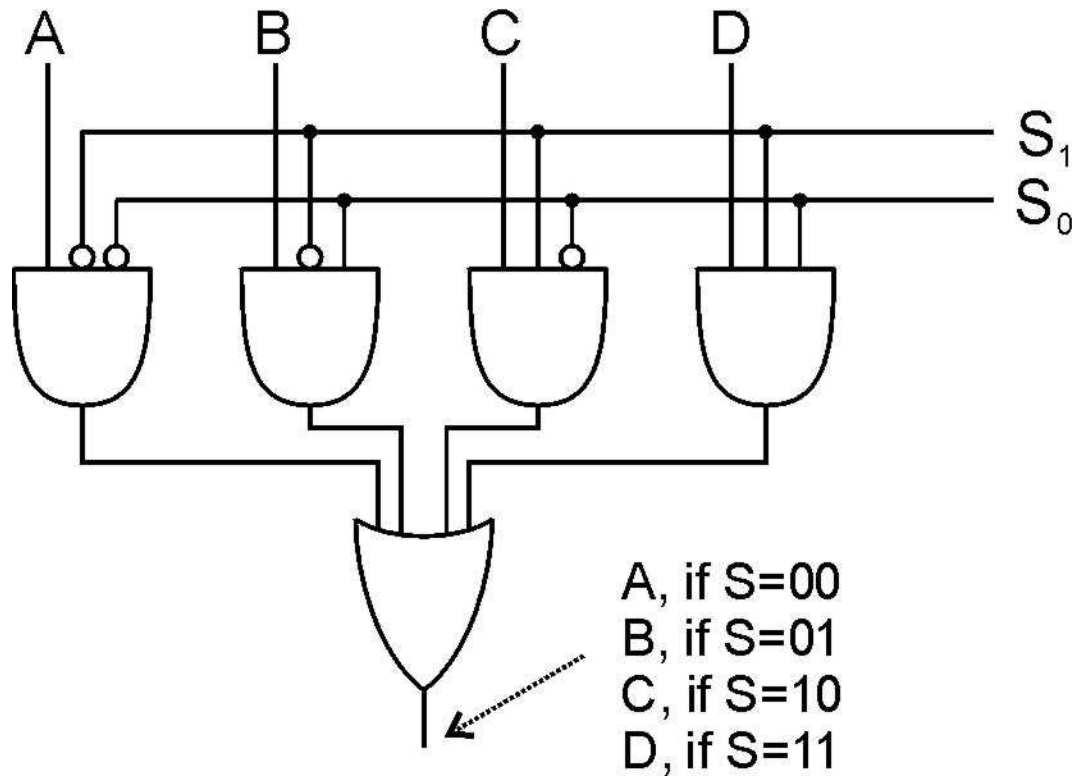
2-  
разрядный  
декодер



# Мультиплексор (MUX)

$n$ -разрядный селектор  $2^n$  входов, один выход

- output equals one of the inputs, depending on selector

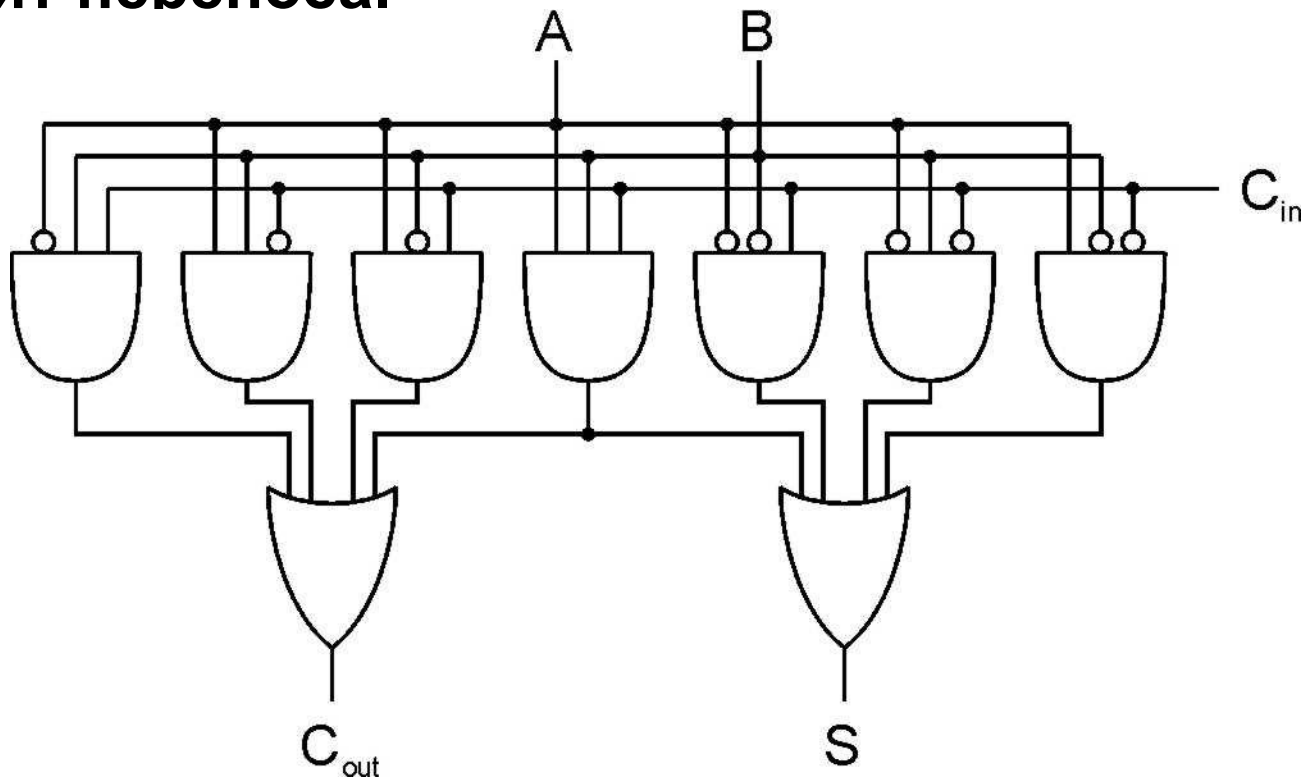


*4-to-1 MUX*



# Полный сумматор

2-разрядное сложение с переносом,  
на выходе 1-разрядная сумма и  
бит переноса.



A	B	$C_{in}$	S	$C_{out}$
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1

# 4-разрядный сумматор

