

Цифровая схемотехника и архитектура компьютера, второе издание

Дэвид М. Харрис и Сара Л. Харрис

Цифровая схемотехника и архитектура компьютера

Эти слайды предназначены для преподавателей, которые читают лекции на основе учебника «Цифровая схемотехника и архитектура компьютера» авторов Дэвида Харриса и Сары Харрис. Бесплатный русский перевод второго издания этого учебника можно загрузить с сайта компании Imagination Technologies:

<https://community.imgtec.com/downloads/digital-design-and-computer-architecture-russian-edition-second-edition>

Процедура регистрации на сайте компании Imagination Technologies описана на странице:

<http://www.silicon-russia.com/2016/08/04/harris-and-harris-2/>

Благодарности

Перевод данных слайдов на русский язык был выполнен командой сотрудников университетов и компаний из России, Украины, США в составе:

- Александр Барабанов - доцент кафедры компьютерной инженерии факультета радиофизики, электроники и компьютерных систем Киевского национального университета имени Тараса Шевченко, кандидат физ.-мат. наук, Киев, Украина;
- Антон Брюзгин - начальник отдела АО «Вибро-прибор», Санкт-Петербург, Россия.
- Евгений Короткий - доцент кафедры конструирования электронно-вычислительной аппаратуры факультета электроники Национального технического университета Украины «Киевский Политехнический Институт», руководитель открытой лаборатории электроники Lamra, кандидат технических наук, Киев, Украина;
- Евгения Литвинова – заместитель декана факультета компьютерной инженерии и управления, доктор технических наук, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники, Харьков, Украина;
- Юрий Панчул - старший инженер по разработке и верификации блоков микропроцессорного ядра в команде MIPS I6400, Imagination Technologies, отделение в Санта-Кларе, Калифорния, США;
- Дмитрий Рожко - инженер-программист АО «Вибро-прибор», магистр Санкт-Петербургского государственного автономного университета аэрокосмического приборостроения (ГУАП), Санкт-Петербург, Россия;
- Владимир Хаханов – декан факультета компьютерной инженерии и управления, проректор по научной работе, доктор технических наук, профессор кафедры автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники, Харьков, Украина;
- Светлана Чумаченко – заведующая кафедрой автоматизации проектирования вычислительной техники Харьковского национального университета радиоэлектроники, доктор технических наук, профессор, Харьков, Украина.

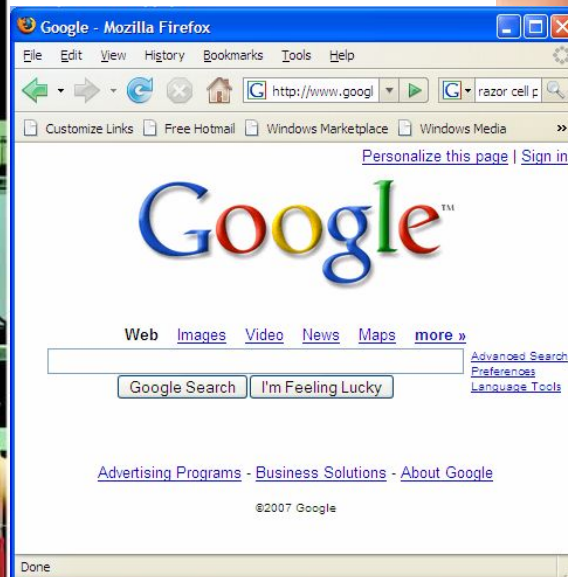


Глава1 :: Темы

- Микроэлектронная промышленность
- План игры
- Искусство управления сложностью
- Цифровая абстракция
- Системы счисления
- Логические элементы
- Логические уровни
- КМОП транзисторы
- Энергопотребление

Полупроводниковая микроэлектроника

- Микропроцессоры кардинально изменили наш мир
 - Сотовые телефоны, интернет, достижения в медицинской сфере
- Объем продаж полупроводниковой промышленности вырос с 21 миллиарда долларов в 1985 году до 300 миллиардов долларов в 2011



ОТНУЛЯ К
ЕДИНИЦЕ

План игры

- Цель курса:
 - Понять, что происходит внутри корпуса компьютера
 - Изучить основные принципы цифровой схемотехники
 - Научиться разрабатывать проекты увеличивающейся сложности
 - Научиться проектировать микропроцессоры

ОТ НУЛЯ К
ЕДИНИЦЕ

Искусство управления



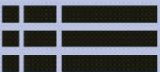
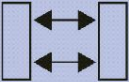
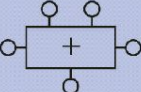
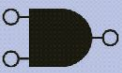
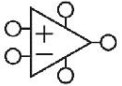
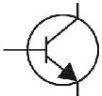

- Абстракция
- Дисциплина
- Три базовых принципа
 - Иерархичность
 - Модульность
 - Регулярность

ОТ НУЛЯ К
ЕДИНИЦЕ

Абстракция

- Исключение из рассмотрения деталей, которые в данном контексте неважны

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

Дисциплина

- Намеренное ограничение выбора возможных проектных решений

- Пример: Цифровая дисциплина

- Использование дискретных значений напряжений вместо непрерывных
- Цифровые системы проще проектировать, чем аналоговые – можно создать более сложные устройства
- Аналоговые предшественники были вытеснены цифровыми системами:
 - например, цифровые камеры, цифровое телевидение, сотовые телефоны, компакт-диски

Три базовых принципа

- **Иерархичность**

- Система разделяется на модули и подмодули

- **Модульность**

- Каждый модуль имеет четко определенные функции и интерфейсы

- **Регулярность**

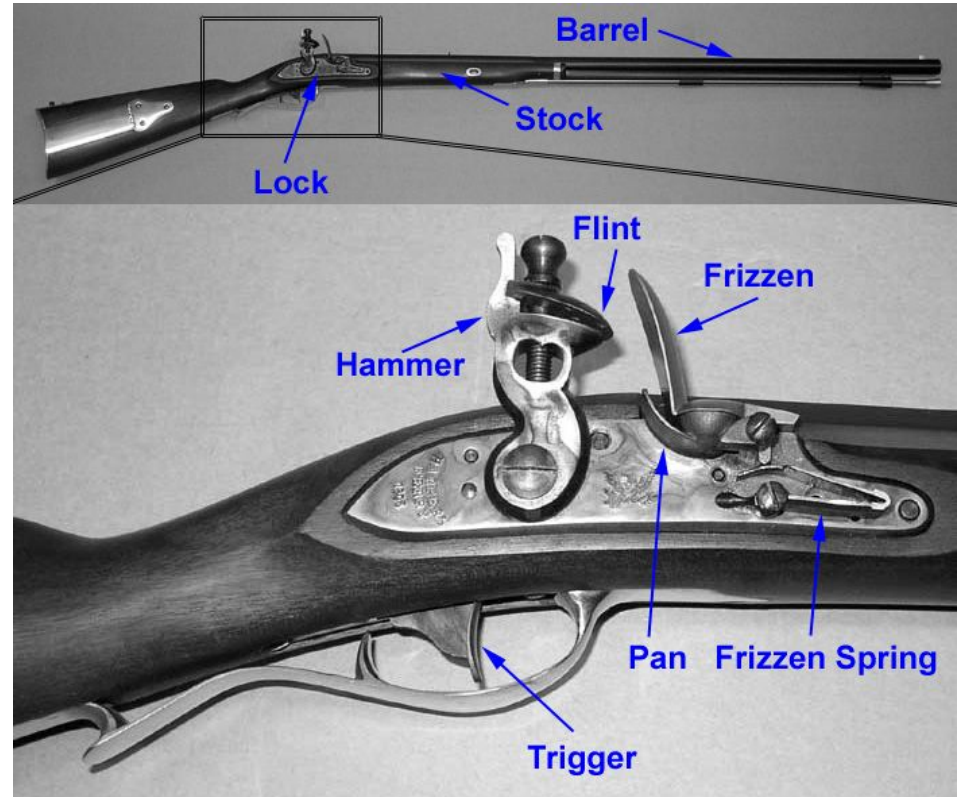
- Поощрение единообразия, что позволяет многократно использовать модули

ОТ НУЛЯ К
ЕДИНИЦЕ

Пример: Кремневое ружье

• Иерархичность

- Три главные модуля: ствол, ударно-спусковой механизм и приклад с цевьем
- Подмодули ударно-спускового механизма: крючок, курок, кремль и т.д.



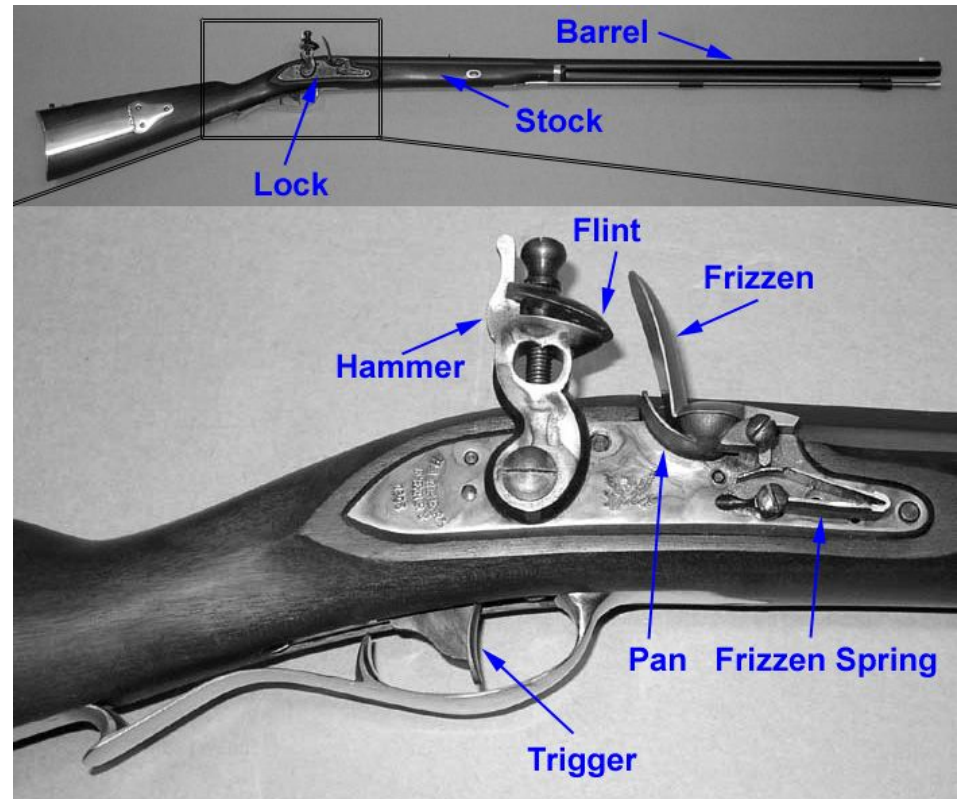
Пример: Кремневое ружье

- **Модульность**

- **Функции приклада и цевья:** служить базой для установки ствола и ударно-спускового механизма
- **Интерфейс приклада и цевья:** длина и расположение посадочных мест

- **Регулярность**

- Взаимозаменяемые детали



Цифровая абстракция

ОТ НУЛЯ К

ЕДИНИЦЕ

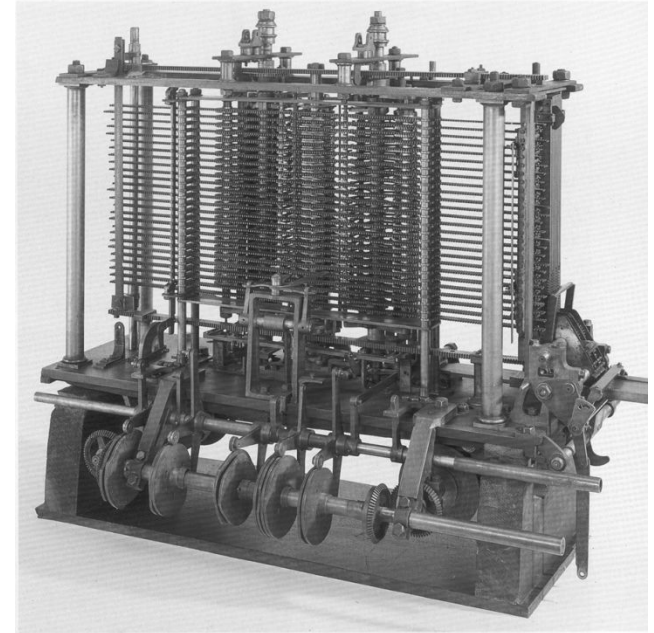
- Большинство физических величин **непрерывны**
 - Потенциал проводника
 - Частота колебаний
 - Положение тела
- Цифровая абстракция рассматривает **дискретное множество** возможных значений

Аналитическая машина

ОТ НУЛЯ К

ЕДИНИЦЕ

- Спроектирована Чарльзом Бэббиджем в 1834 – 1871 годах
- Считается первым цифровым компьютером
- Построена из механических шестеренок, каждая шестеренка представляла дискретную величину (0-9)
- Бэббидж не дожил до окончания работ над машиной



- **Два дискретные значения:**

- 1 и 0

- 1, Истина, Большая величина

- 0, Ложь, Малая величина

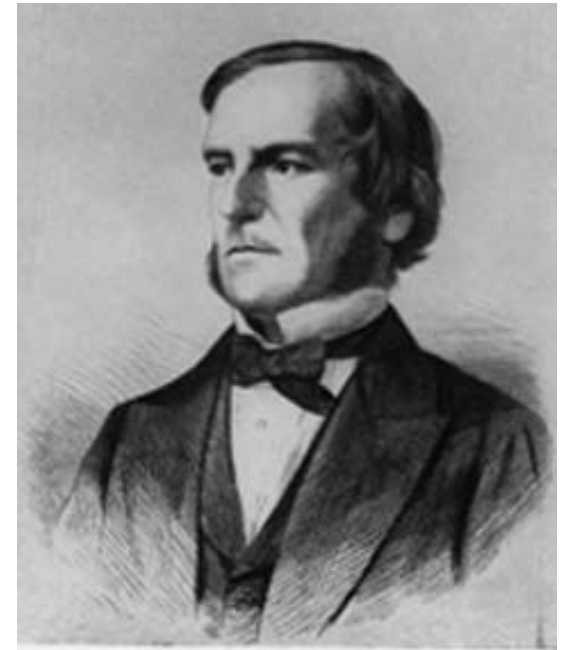
- **1 и 0:** Величина напряжения, угол поворота шестеренки, уровень жидкости и т.д.

- Цифровые схемы используют значение **напряжения** для представления 0 и 1

- **Bit (Bit):** Двоичная цифра (*Binary digit*)

Джордж Буль, 1815-1864

- Родился в семье небогатого ремесленника
- Самостоятельно изучал математику и стал преподавателем Королевского колледжа в Ирландии.
- Написал работу *Исследование законов мышления* (1854)
- Ввел двоичные переменные
- Ввел три основных логических оператора: И, ИЛИ, НЕ (AND, OR, NOT)



GEORGE BOOLE

Scanned at the American
Institute of Physics

СИСТЕМЫ СЧИСЛЕНИЯ

- Десятичные числа

1's column
10's column
100's column
1000's column

$$5374_{10} =$$

- Двоичные числа

1's column
2's column
4's column
8's column

$$1101_2 =$$

ОТ НУЛЯ К
ЕДИНИЦЕ

СИСТЕМЫ СЧИСЛЕНИЯ

- Десятичные числа

1's column
10's column
100's column
1000's column

$$5374_{10} = 5 \times 10^3 + 3 \times 10^2 + 7 \times 10^1 + 4 \times 10^0$$

five three seven four
thousands hundreds tens ones

- Двоичные числа

1's column
2's column
4's column
8's column

$$1101_2 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$$

one one no one
eight four two one

ОТ НУЛЯ К
ЕДИНИЦЕ



Степени числа 2

- $2^0 =$

- $2^1 =$

- $2^2 =$

- $2^3 =$

- $2^4 =$

- $2^5 =$

- $2^6 =$

- $2^7 =$

- $2^8 =$

- $2^9 =$

- $2^{10} =$

- $2^{11} =$

- $2^{12} =$

- $2^{13} =$

- $2^{14} =$

- $2^{15} =$

ОТ НУЛЯ К

ЕДНИЦЕ

Степени числа 2

- $2^0 = 1$

- $2^1 = 2$

- $2^2 = 4$

- $2^3 = 8$

- $2^4 = 16$

- $2^5 = 32$

- $2^6 = 64$

- $2^7 = 128$

- Желательно запомнить до 2^9

- $2^8 = 256$

- $2^9 = 512$

- $2^{10} = 1024$

- $2^{11} = 2048$

- $2^{12} = 4096$

- $2^{13} = 8192$

- $2^{14} = 16384$

- $2^{15} = 32768$

ОТ НУЛЯ К

ЕДИНИЦЕ

- Преобразование двоичного числа в десятичное:

- Преобразовать 10011_2 в десятичное число

- Преобразование десятичного числа в двоичное:

- Преобразовать 47_{10} в двоичное число

- Преобразование десятичного числа в двоичное:

- Преобразовать 10011_2 в десятичное число
- $16 \times 1 + 8 \times 0 + 4 \times 0 + 2 \times 1 + 1 \times 1 = 19_{10}$

- Преобразование десятичного числа в двоичное:

- Преобразовать 47_{10} в двоичное число
- $32 \times 1 + 16 \times 0 + 4 \times 1 + 2 \times 1 + 1 \times 1 = 101111_2$



Двоичные числа и их диапазоны

ОТ НУЛЯ К
ЕДИНИЦЕ

- N -разрядное десятичное число
 - Сколько значений?
 - Диапазон?
 - Пример: Трехразрядное десятичное число
- N -битовое двоичное число
 - Сколько значений?
 - Диапазон:
 - Пример: Трехразрядное двоичное число

Двоичные числа и их диапазоны

ОТ НУЛЯ К
ЕДИНИЦЕ

- N -разрядное десятичное число
 - Сколько значений? 10^N
 - Диапазон? $[0, 10^N - 1]$
 - Пример: Трехразрядное десятичное число
 - $10^3 = 1000$ возможных значений
 - Диапазон: $[0, 999]$
- N -битовое двоичное число
 - Сколько значений? 2^N
 - Диапазон: $[0, 2^N - 1]$
 - Пример: Трехразрядное двоичное число
 - $2^3 = 8$ возможных значений
 - Диапазон: $[0, 7] = [\text{от } 000_2 \text{ до } 111_2]$

Шестнадцатеричные числа

ОТ НУЛЯ К
ЕДИНИЦЕ

Шестнадцатеричная цифра	Десятичный эквивалент	Двоичный эквивалент
0	0	
1	1	
2	2	
3	3	
4	4	
5	5	
6	6	
7	7	
8	8	
9	9	
A	10	
B	11	
C	12	
D	13	
E	14	
F	15	

Шестнадцатеричные числа

ОТ НУЛЯ К
ЕДИНИЦЕ

Шестнадцатеричная цифра	Десятичный эквивалент	Двоичный эквивалент
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

Шестнадцатеричные числа

- Основание 16
- Компактная запись двоичных чисел

ОТ НУЛЯ К
ЕДИНИЦЕ

Преобразование шестнадцатеричных чисел в двоичные

- Преобразование шестнадцатеричных чисел в двоичные:

- Преобразовать $4AF_{16}$ (также записывается $0x4AF$) в двоичное число

- Преобразование шестнадцатеричных чисел в двоичные:

- Преобразовать $0x4AF$ в десятичное число

ОТ НУЛЯ К
ЕДИНИЦЕ

• Преобразование шестнадцатеричных чисел

в двоичные:

- Преобразовать $4AF_{16}$ (также записывается $0x4AF$) в двоичное число
- $0100\ 1010\ 1111_2$

• Преобразование шестнадцатеричных чисел

в десятичные:

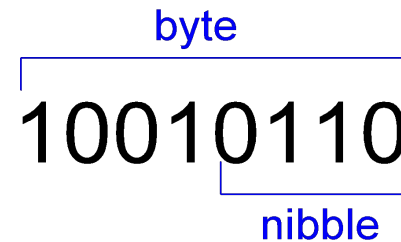
- Преобразовать $4AF_{16}$ в десятичное число
- $16^2 \times 4 + 16^1 \times 10 + 16^0 \times 15 = 1199_{10}$

Биты, байты, полубайты...

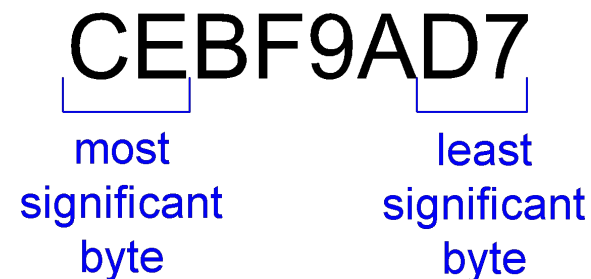
- БИТЫ



- Байты и
- полубайты (nibble)



- Байты



ОТ НУЛЯ К
ЕДИНИЦЕ



Большие степени 2

- $2^{10} = 1$ кило ≈ 1000 (1024)
- $2^{20} = 1$ мега ≈ 1 миллион (1,048,576)
- $2^{30} = 1$ гига ≈ 1 миллиард (1,073,741,824)

ОТ НУЛЯ К
ЕДИНИЦЕ

Вычисление степеней 2

- Чему равно 2^{24} ?
- Сколько значений может представить 32-битовая переменная?

ОТ НУЛЯ К
ЕДИНИЦЕ

Вычисление степеней 2

- Чему равно 2^{24} ?

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

- Сколько значений может представить 32-битовая переменная?

- $2^2 \times 2^{30} \approx 4$ миллиарда

ОТ НУЛЯ К
ЕДИНИЦЕ

Сложение

- Десятичное

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

- Двоичное

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

ОТ НУЛЯ К
ЕДИНИЦЕ

Примеры сложения двоичных чисел

- Сложите следующие 4-битовые двоичные числа

$$\begin{array}{r} 1001 \\ + 0101 \\ \hline \end{array}$$

- Сложите следующие 4-битовые двоичные числа

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

ОТ НУЛЯ К
ЕДИНИЦЕ

Примеры сложения двоичных чисел

- Сложите следующие 4-битовые двоичные числа

$$\begin{array}{r} 1 \\ 1001 \\ + 0101 \\ \hline 1110 \end{array}$$

- Сложите следующие 4-битовые двоичные числа

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

Переполнение!

Переполнение

- Цифровые системы работают с **фиксированным количеством разрядов**
- Переполнение: когда результат слишком большой, чтобы поместится в доступном количестве разрядов
- Вспомните пример сложения $11 + 6$

Двоичные числа со знаком

- Числа в прямом коде
- Числа в дополнительном коде

ОТ НУЛЯ К
ЕДИНИЦЕ

Числа в прямом коде

- Один знаковый бит, $N-1$ битов величины
- Знаковый бит является старшим (самым левым) битом

– Положительные числа:

знаковый бит = 1

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

– Отрицательные числа:

знаковый бит = 0

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

- Пример, 4-битовое представление числа ± 6 в прямом коде

+6 =

-6 =

- Диапазон N -битового числа в прямом коде:



Числа в прямом коде

- Один знаковый бит, $N-1$ битов величины
- Знаковый бит является старшим (самым левым) битом

– Положительные числа:

знаковый бит = 1

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

– Отрицательные числа:

знаковый бит = 0

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

- Пример, 4-битовое представление числа ± 6 в прямом коде

$$+6 = \mathbf{0110}$$

$$-6 = \mathbf{1110}$$

- Диапазон N -битового числа в прямом коде:
 $[-(2^{N-1}-1), 2^{N-1}-1]$

Числа в прямом коде

- Недостатки:

- Стандартный способ сложения не работает, например, $-6 + 6$:

$$\begin{array}{r} 1110 \\ +0110 \\ \hline \end{array}$$

10100 (не правильно!)

- Два представления числа 0 (± 0):

1000

0000

ОТ НУЛЯ К
ЕДИНИЦЕ

Числа в дополнительном

- Не имеет проблем прямого кода
 - Алгоритм сложения работает
 - Единственное представление 0

ОТ НУЛЯ К
ЕДИНИЦЕ

Числа в дополнительном

- Старший бит имеет вес -2^{N-1}

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

- Наибольшее положительное n -битовое число
- Наибольшее (по модулю) отрицательное n -битовое число
- Старший бит, как и ранее, показывает знак (1 = отрицательное, 0 = положительное)
- Диапазон N -битового числа в дополнительном коде:

ОТ НУЛЯ К
ЕДИНИЦЕ

Числа в дополнительном

- Старший бит имеет вес -2^{N-1}

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

- Наибольшее положительное 4-битовое число **0111**
- Наибольшее (по модулю) отрицательное 4-битовое число **1000**
- Старший бит, как и ранее, показывает знак (1 = отрицательное, 0 = положительное)
- Диапазон N -битового числа в дополнительном коде:
 $[-(2^{N-1}-1), 2^{N-1}-1]$



Операция “Дополнение до двух”

- Изменение знака числа, представленного в дополнительном коде

• Метод:

1. Инвертировать биты
2. Добавить 1

• Пример: Изменить знак $3_{10} = 0011_2$

Операция “Дополнение до двух”

- Изменение знака числа, представленного в дополнительном коде

• Метод:

1. Инвертировать биты
2. Добавить 1

• Пример: Изменить знак $3_{10} = 0011_2$

1. $\underline{1100}$

2. $+ 1$

$1101 = -3_{10}$

ОТНУЛЯ К
ЕДИНИЦЕ

Примеры вычислений с числами в дополнительном коде

- Найти представление в дополнительном коде числа $6_{10} = 0110_2$
- Чему равно десятичное представление числа 1001_2 ?

ОТ НУЛЯ К
ЕДИНИЦЕ

Примеры вычислений с числами в дополнительном коде

ОТНУЛЯ К
ЕДИНИЦЕ

- Найти представление в дополнительном коде $6_{10} = 0110_2$

1. 1001

2. + 1

$$1010 = -6_{10}$$

- Чему равно десятичное представление числа в дополнительном коде 1001_2 ?

1. 0110

2. + 1

$$\text{---}0111_2 = 7_{10}, \text{ следовательно } 1001_2 = -7_{10}$$

Сложение чисел в дополнительном коде

ОТ НУЛЯ К
ЕДИНИЦЕ

- Сложить числа $6 + (-6)$ с использованием дополнительного кода

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

- Сложить числа $-2 + 3$ с использованием дополнительного кода

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

Сложение чисел в дополнительном коде

ОТ НУЛЯ К
ЕДИНИЦЕ

- Сложить числа $6 + (-6)$ с использованием дополнительного кода

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

- Сложить числа $-2 + 3$ с использованием дополнительного кода

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

Увеличение количества бит

ОТ НУЛЯ К
ЕДИНИЦЕ

- Увеличить количество бит с N до M ($M > N$):
 - Знаковое расширение
 - Дополнение нулями

Знаковое расширение

- Знаковый бит копируется во все новые старшие биты
- Значение числа не изменяется
- **Пример 1:**
 - 4-битовое представление $3 = 0011$
 - 8-битовое представление: 00000011
- **Пример 2:**
 - 4-битовое представление $-5 = 1011$
 - 8-битовое представление: 11111011

Дополнение нулями

- Все новые старшие биты принимают нулевое значение
- Значение отрицательных чисел изменяется

- **Пример 1:**

– 4-битовая величина = $0011_2 = 3_{10}$

– 8-битовая величина после дополнения нулями:
 $00000011 = 3_{10}$

- **Пример 2:**

– 4-битовая величина = $1011_2 = -5_{10}$

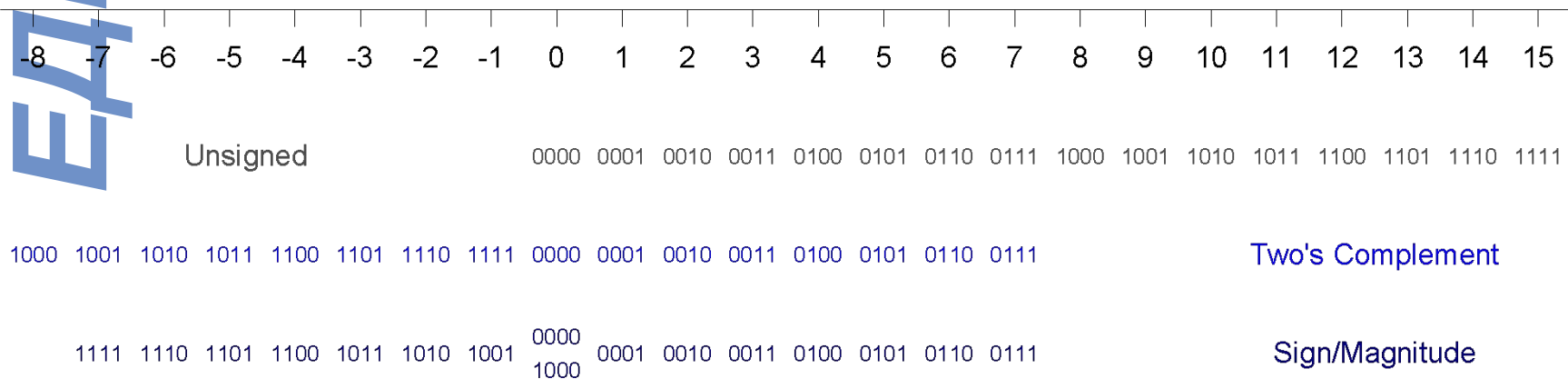
– 8-битовая величина после дополнения нулями:
 $00001011 = 11_{10}$



Сравнение способов представления двоичных чисел

Представление	Диапазон
Числа без знака	$[0, 2^N-1]$
Числа в прямом коде	$[-(2^{N-1}-1), 2^{N-1}-1]$
Числа в дополнительном коде	$[-2^{N-1}, 2^{N-1}-1]$

Пример: 4-битовое представление:



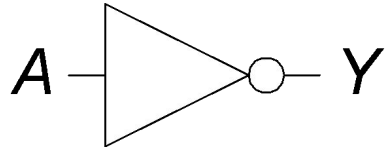
- **Выполняют логические функции**
 - Инверсия (НЕ), И (AND), ИЛИ (OR), И-НЕ (NAND), ИЛИ-НЕ (NOR), и т.д.
- **С одним входом**
 - Элемент НЕ, буфер
- **С двумя входами**
 - И, ИЛИ, И-НЕ, ИЛИ-НЕ, Исключающее ИЛИ, Исключающее ИЛИ-НЕ
- **С несколькими входами**

Логические элементы с одним

входом

ОТ НУЛЯ К
ЕДИНИЦЕ

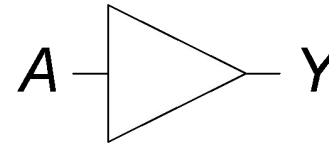
NOT



$$Y = \bar{A}$$

A	Y
0	1
1	0

BUF



$$Y = A$$

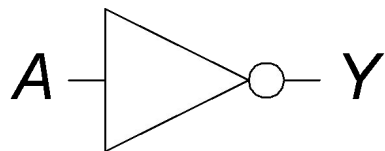
A	Y
0	0
1	1

Логические элементы с одним

ВХОДОМ

ОТ НУЛЯ К
ЕДИНИЦЕ

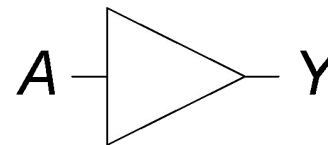
NOT



$$Y = \bar{A}$$

A	Y
0	1
1	0

BUF



$$Y = A$$

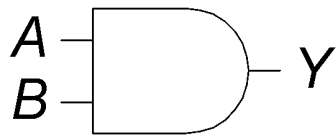
A	Y
0	0
1	1

Логические элементы с двумя

ВХОДАМИ

ОТ НУЛЯ К
ЕДИНИЦЕ

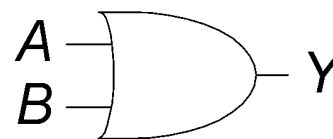
AND



$$Y = AB$$

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

OR



$$Y = A + B$$

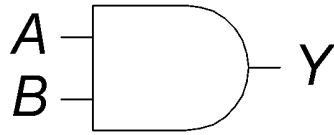
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Логические элементы с двумя

ВХОДАМИ

ОТ НУЛЯ К
ЕДИНИЦЕ

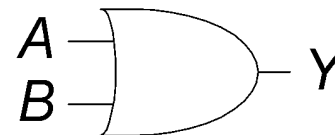
AND



$$Y = AB$$

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

OR



$$Y = A + B$$

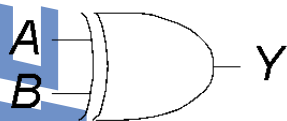
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

Прочие логические элементы с двумя входами

ОТ НУЛЯ К

ЕДИНИЦЕ

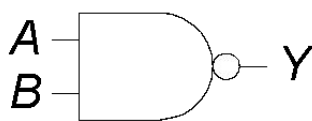
XOR



$$Y = A \oplus B$$

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

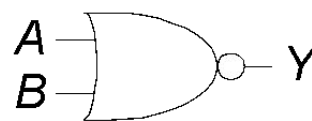
NAND



$$Y = \overline{AB}$$

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

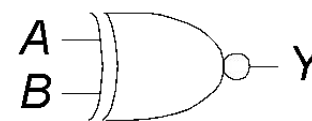
NOR



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

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

XNOR



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

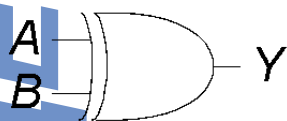
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	1

Прочие логические элементы с двумя входами

ОТ НУЛЯ К

ЕДИНИЦЕ

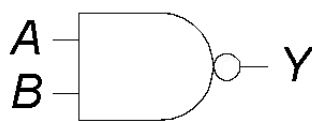
XOR



$$Y = A \oplus B$$

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

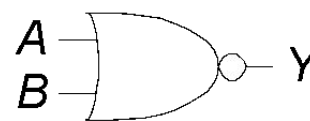
NAND



$$Y = \overline{AB}$$

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

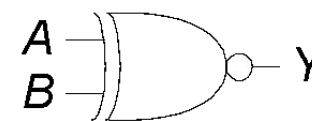
NOR



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

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

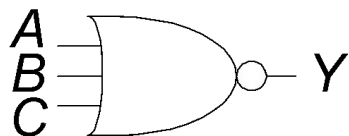
XNOR



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

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

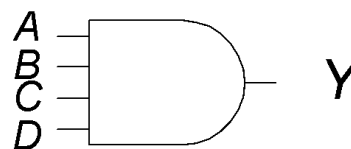
NOR3



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

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	

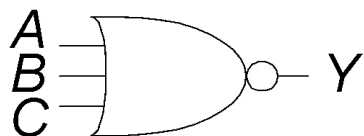
AND4



$$Y = ABCD$$

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	

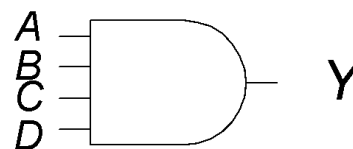
NOR3



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

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

AND4



$$Y = ABCD$$

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

- Многовходовый элемент XOR: Контроль четности

Логические уровни

- Дискретные уровни напряжения представляют 1 и 0
- Например:
 - 0 = земля (GND) или 0 В
 - 1 = V_{DD} или 5 В
- Как трактовать напряжение 4.99 В? Это 0 или 1?
- Как трактовать напряжение 3.2 В?

Логические уровни

- Диапазон напряжений для 1 и 0
- Разные диапазоны для входов и выходов обеспечивают работу схем при наличии помех и шумов

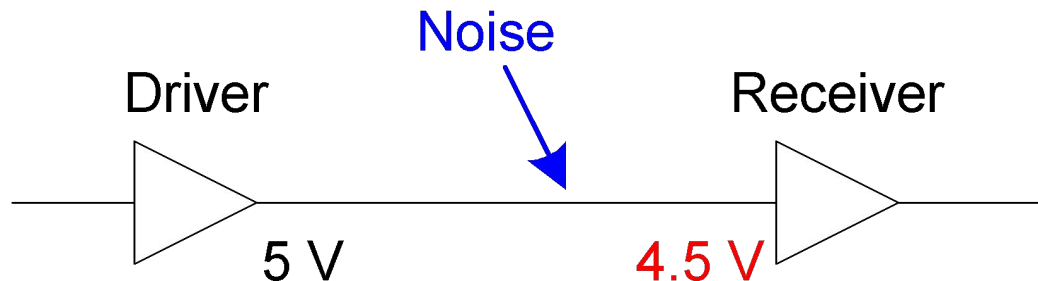
ОТ НУЛЯ К
ЕДИНИЦЕ

Что такое шум?

ОТ НУЛЯ К
ЕДИНИЦЕ

Что такое шум?

- **Любая помеха искажающая сигнал**
 - Например, сопротивление проводников, помехи источника питания, наводки от соседних проводников и т.д.
- **Пример:** элемент (его выходной каскад) выдает 5 В, но из-за сопротивления длинного проводника на приемник поступает 4.5 В

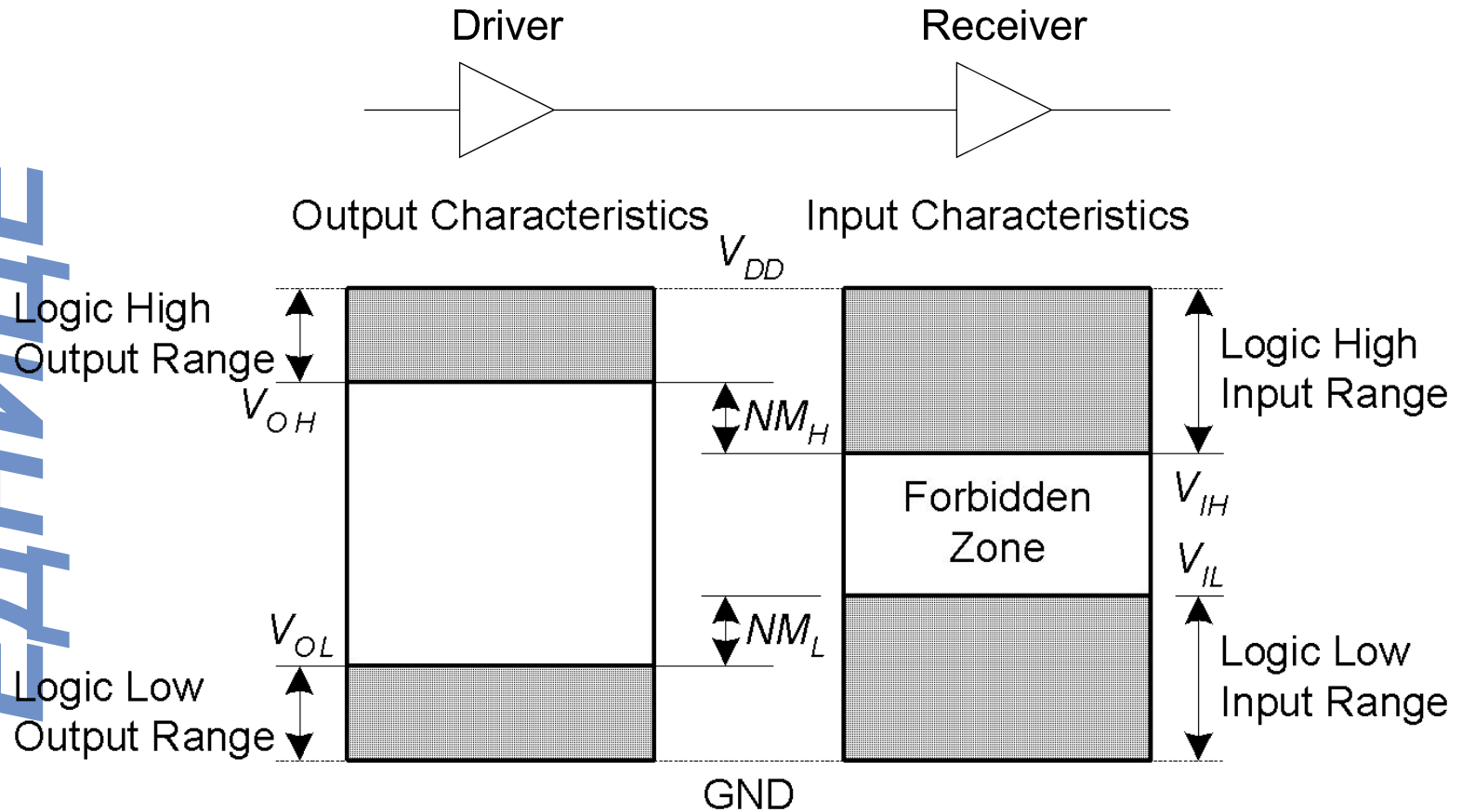


Статическая дисциплина

- Если на вход элемента поступают корректные логические значения, на его выходе формируются корректные выходные сигналы
- Для представления дискретных величин используется ограниченный диапазон напряжений

Логические уровни

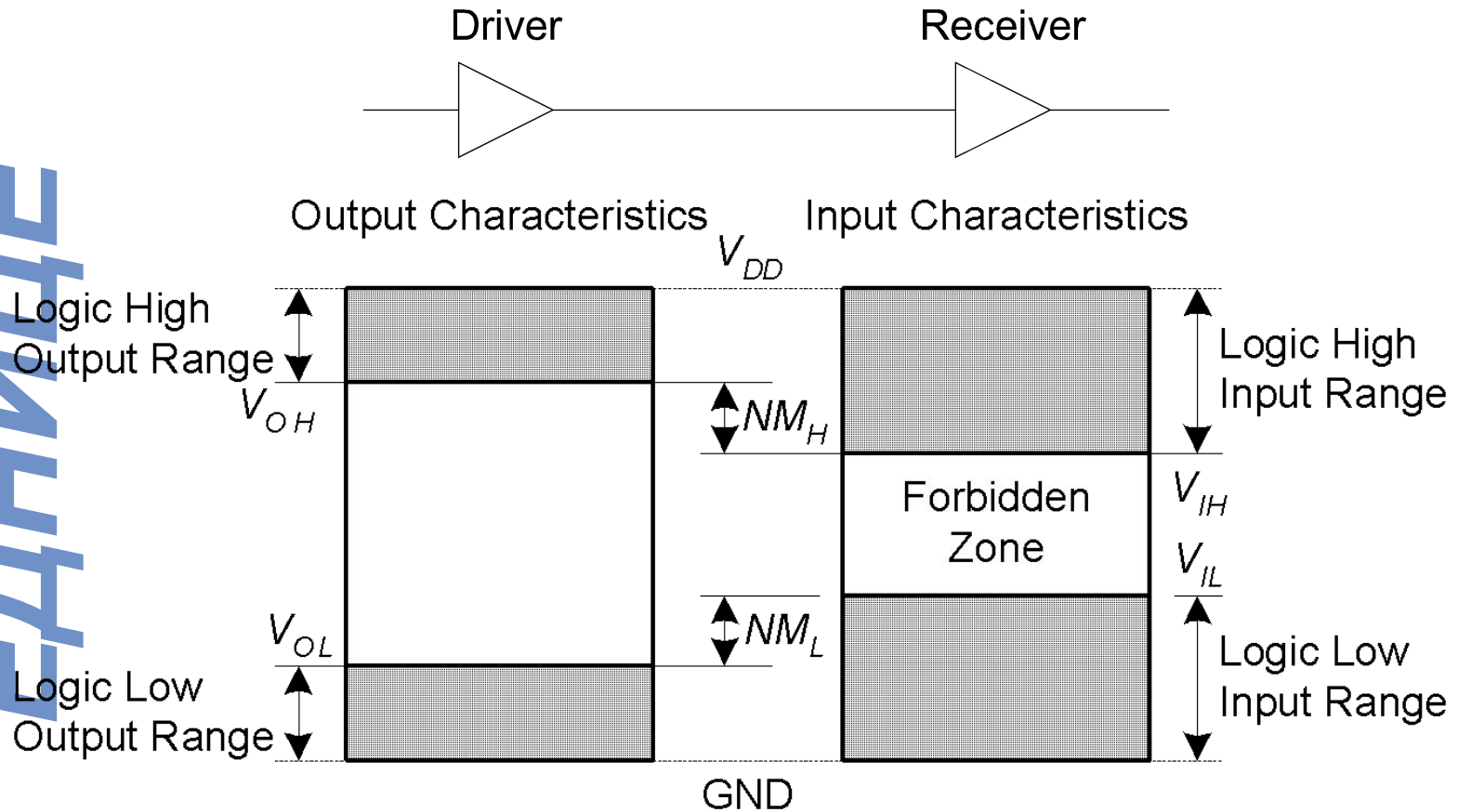
ОТ НУЛЯ К
ЕДИНИЦЕ



Допустимые уровни шумов

ОТ НУЛЯ К

ЕДИНИЦЕ



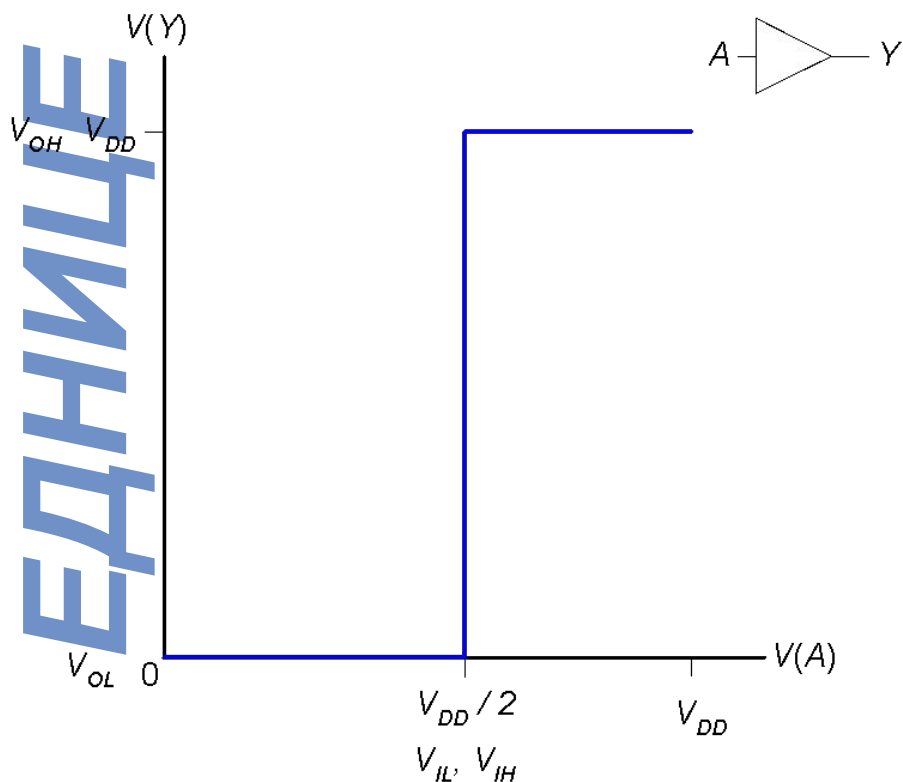
$$NM_H = V_{OH} - V_{IH}$$

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

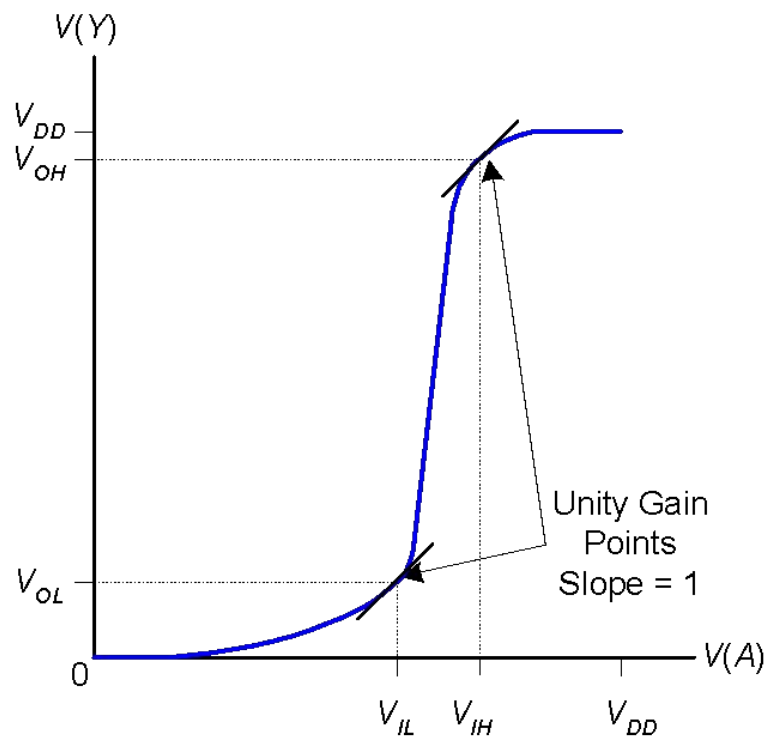
Передаточная характеристика на постоянном токе

ОТ НУЛЯ К ЕДИНИЦЕ

Идеальный буфер:



Реальный буфер:



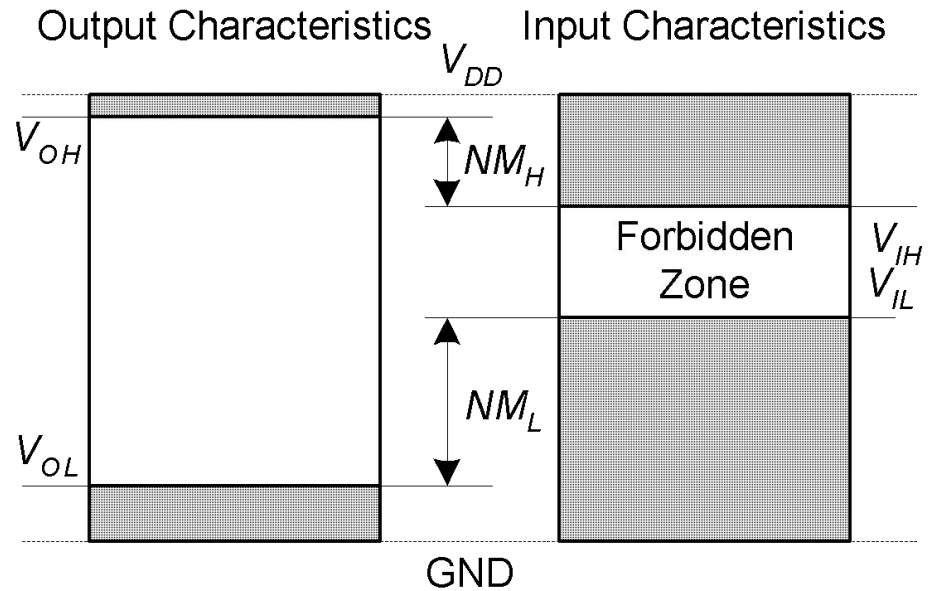
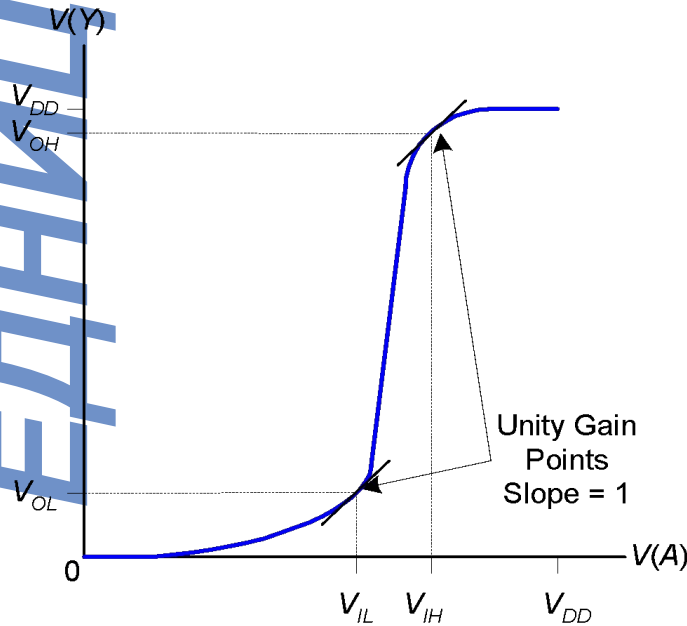
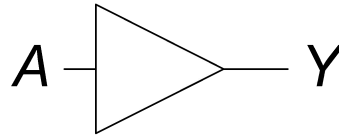
$$NM_H = NM_L = V_{DD}/2$$

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

Передаточная характеристика на постоянном токе

ОТ НУЛЯ К

ЕДНИЦЕ



Изменение V_{DD}

- В 1970 и 1980 годы, $V_{DD} = 5 \text{ В}$
- В следующие годы V_{DD} уменьшается
 - Уменьшается нагрев транзисторов
 - Уменьшается энергопотребление
- 3.3 В, 2.5 В, 1.8 В, 1.5 В, 1.2 В, 1.0 В, ...
- При соединении микросхем с разными напряжениями питания нужно быть очень осторожным

Микросхемы работают, пока они содержат волшебный дым

Доказательство:

- Если волшебный дым покидает микросхему, она перестает работать

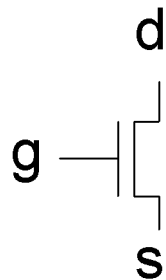


Примеры логических

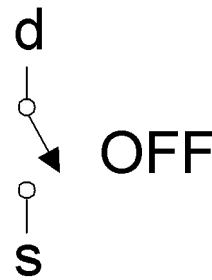
Логические семейства	V_{DD}	V_{IL}	V_{IH}	V_{OL}	V_{OH}
TTL	5 (4.75 - 5.25)	0.8	2.0	0.4	2.4
КМОП	5 (4.5 - 6)	1.35	3.15	0.33	3.84
LVTTL	3.3 (3 - 3.6)	0.8	2.0	0.4	2.4
LVC MOS	3.3 (3 - 3.6)	0.9	1.8	0.36	2.7

Транзисторы

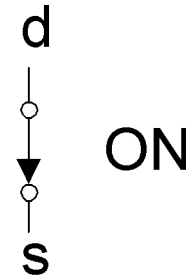
- Логические элементы состоят из транзисторов
- Трехходовый управляемый напряжением выключатель
 - Соединение двух входов зависит от напряжения на третьем
 - d и s соединены (ON) когда g равно 1



$g = 0$



$g = 1$

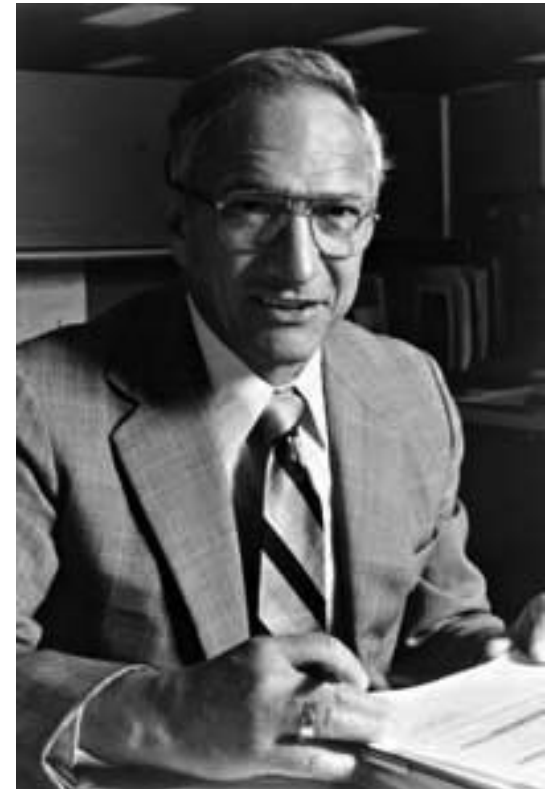


Роберт Нойс, 1927-1990

ОТ НУЛЯ К

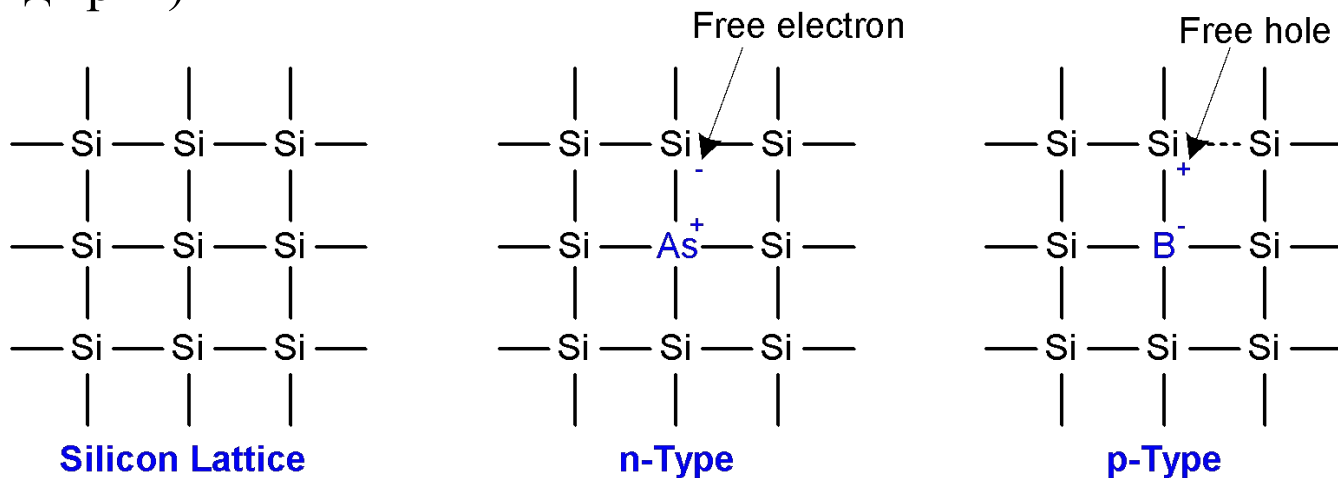
ЕДИНИЦЕ

- Прозвище - “Мэр Силиконовой долины”
- Со-основатель Fairchild Semiconductor в 1957 году
- Со-основатель Intel в 1968 году
- Одни из изобретателей интегральной микросхемы



Кремний

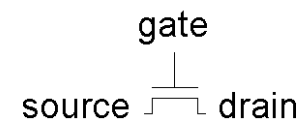
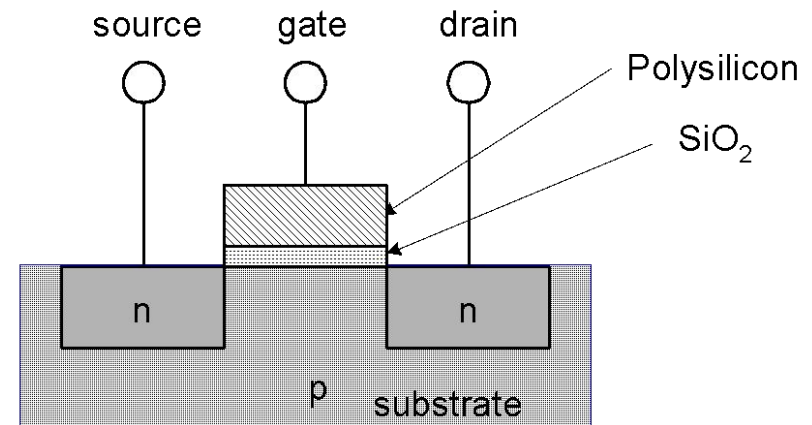
- Транзисторы создаются из полупроводникового материала, кремния
- Чистый кремний плохой проводник (свободные носители заряда отсутствуют)
- Легированный кремний хороший проводник (есть свободные носители заряда)
 - n-типа (свободные носители заряда отрицательные (*negative*), электроны)
 - p-типа (свободные носители заряда положительные (*positive*), дырки)



МОП транзисторы

- **Метал-оксид-полупроводник (МОП) транзисторы:**

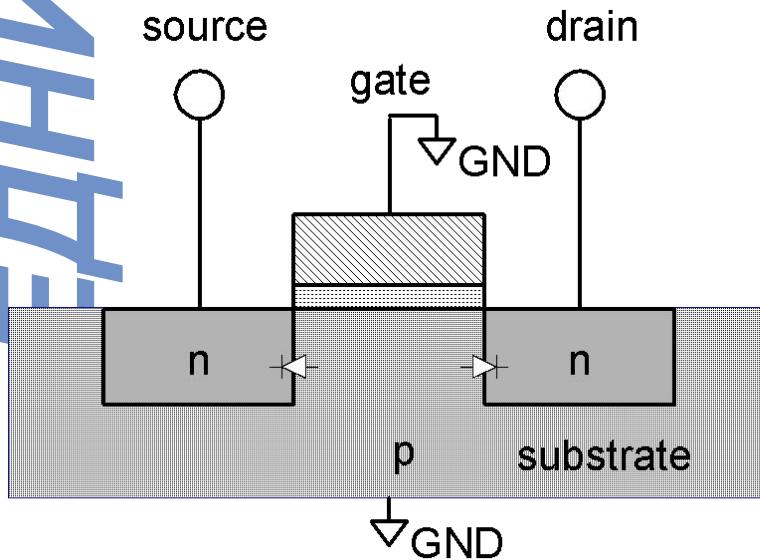
- Поликремниевый (используется как металл) затвор
- Оксидный (диоксид кремния) изолятор
- Легированный кремний



Транзисторы: n-МОП

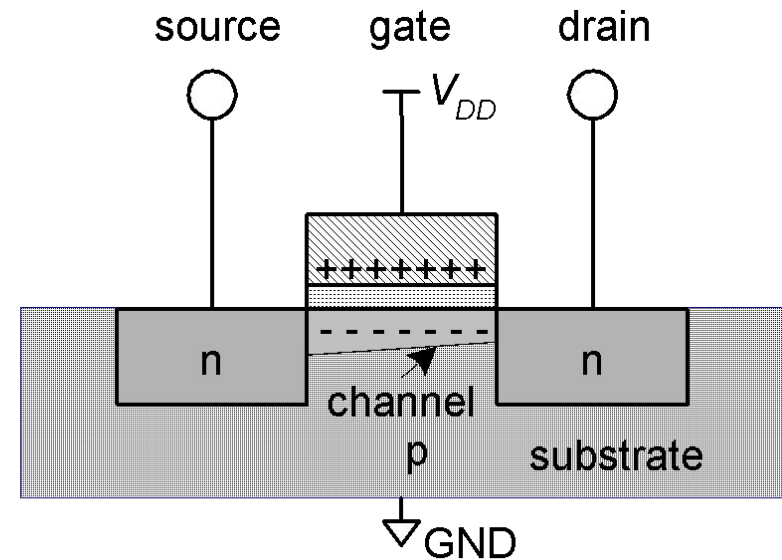
Gate = 0

OFF (исток и сток не соединены)



Gate = 1

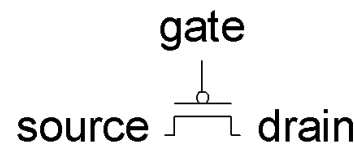
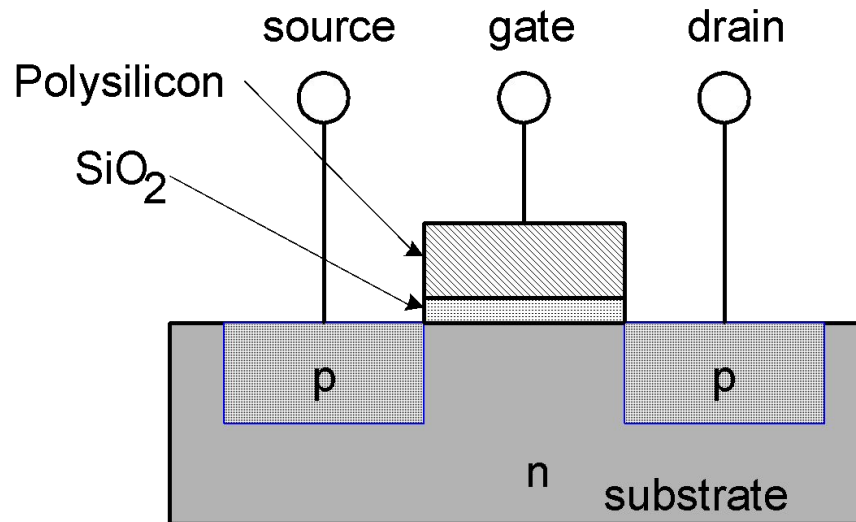
ON (исток и сток соединены)



ОТ НУЛЯ К
ЕДИНИЦЕ

Транзисторы: р-МОП

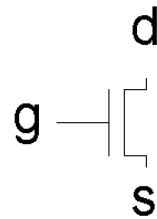
- р-МОП транзистор работает противоположным образом
 - ON, когда Gate = 0
 - OFF, когда Gate = 1



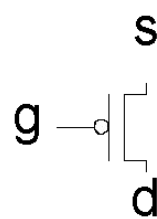
Работа транзистора

ОТ НУЛЯ К
ЕДИНИЦЕ

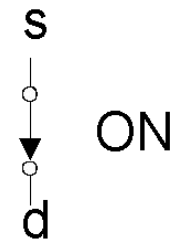
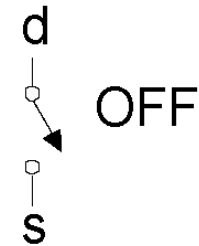
nMOS



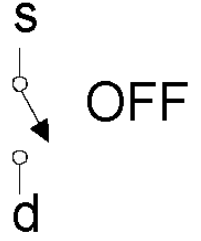
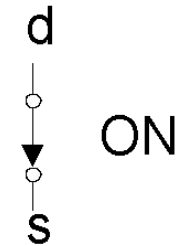
pMOS



$g = 0$

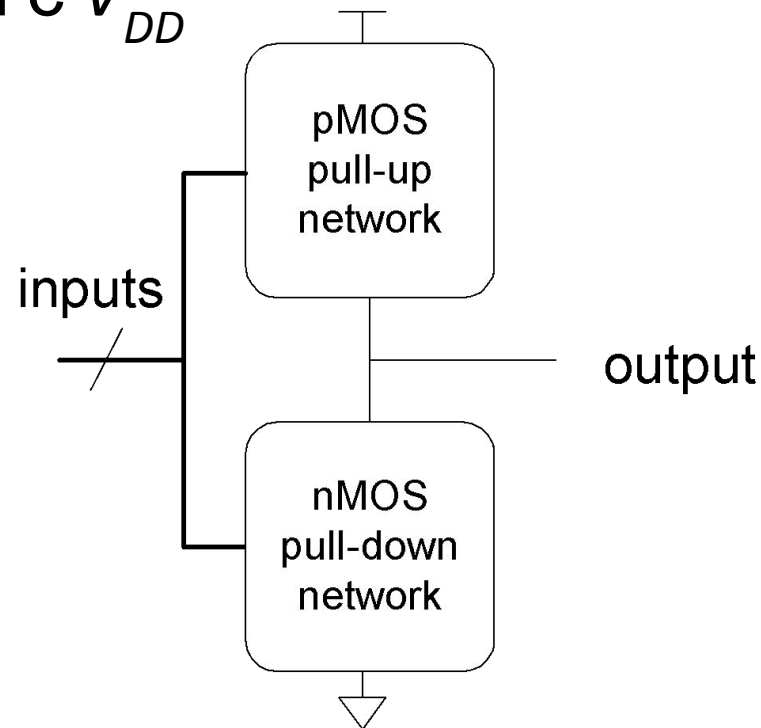


$g = 1$



Работа транзистора

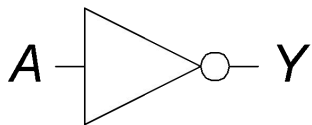
- **n-МОП:** Хорошо передают 0, т.е. исток соединен с GND
- **p-МОП:** Хорошо передают 1, т.е. исток соединен с V_{DD}



Логические элементы КМОП: Логический элемент НЕ:

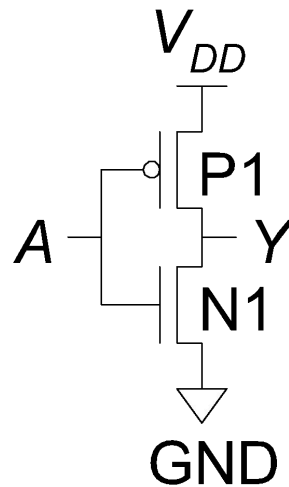
ОТ НУЛЯ К
ЕДИНИЦЕ

NOT



$$Y = \overline{A}$$

A	Y
0	1
1	0

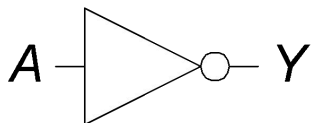


A	P1	N1	Y
0			
1			

Логические элементы КМОП: Логический элемент НЕ:

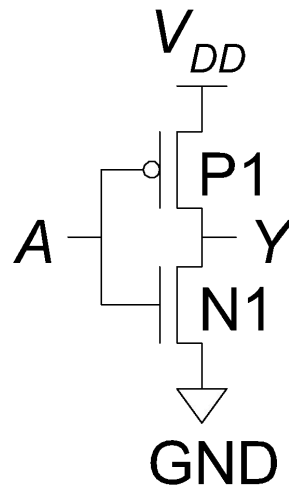
ОТ НУЛЯ К
ЕДИНИЦЕ

NOT



$$Y = \overline{A}$$

A	Y
0	1
1	0

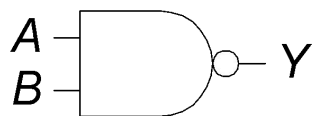


A	P1	N1	Y
0	Вкл.	Выкл.	1
1	Выкл.	Вкл.	0

Логические элементы КМОП: Логический элемент И-НЕ:

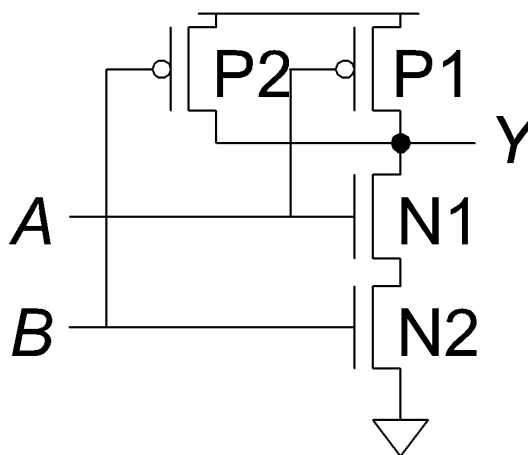
ОТ НУЛЯ К
ЕДИНИЦЕ

NAND



$$Y = \overline{AB}$$

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

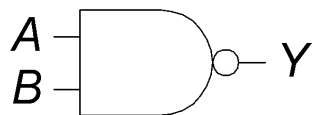


A	B	P1	P2	N1	N2	Y
0	0					
0	1					
1	0					
1	1					

Логические элементы КМОП: Логический элемент И-НЕ:

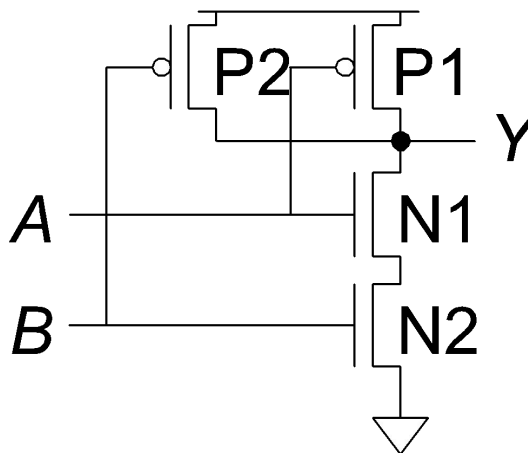
ОТ НУЛЯ К ЕДИНИЦЕ

NAND



$$Y = \overline{AB}$$

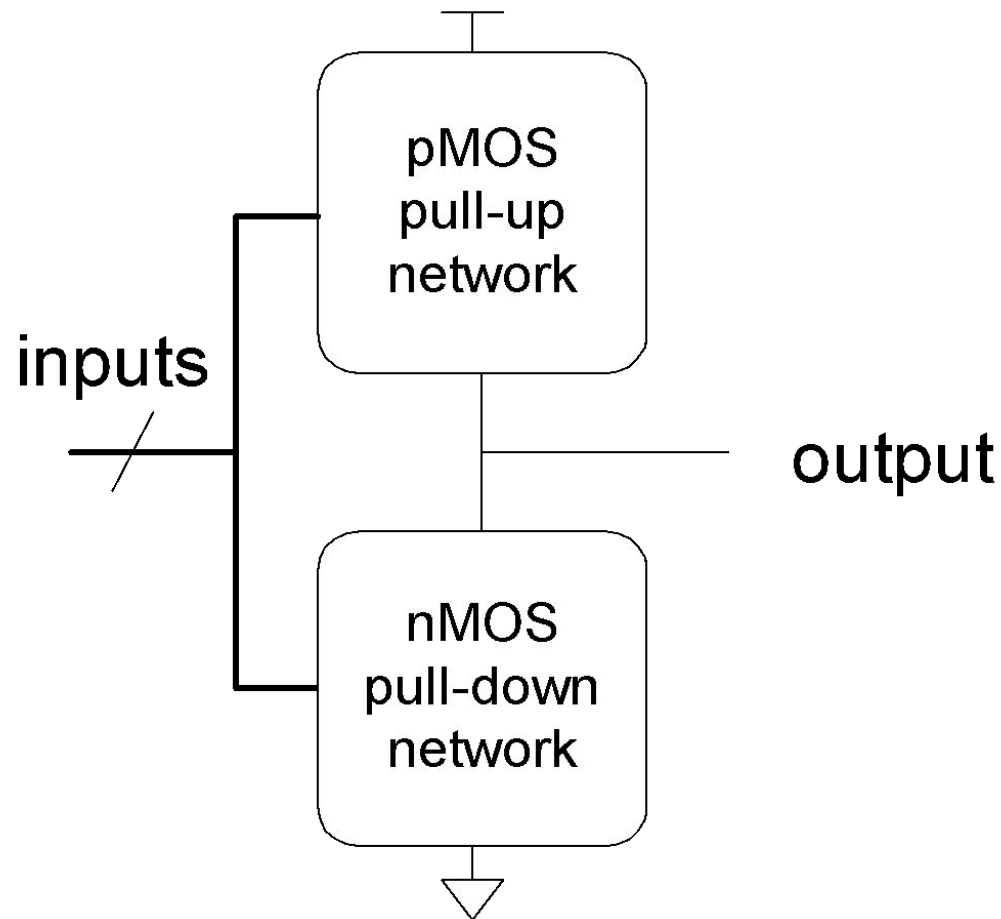
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0



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 элемента КМОП

ОТ НУЛЯ К
ЕДИНИЦЕ



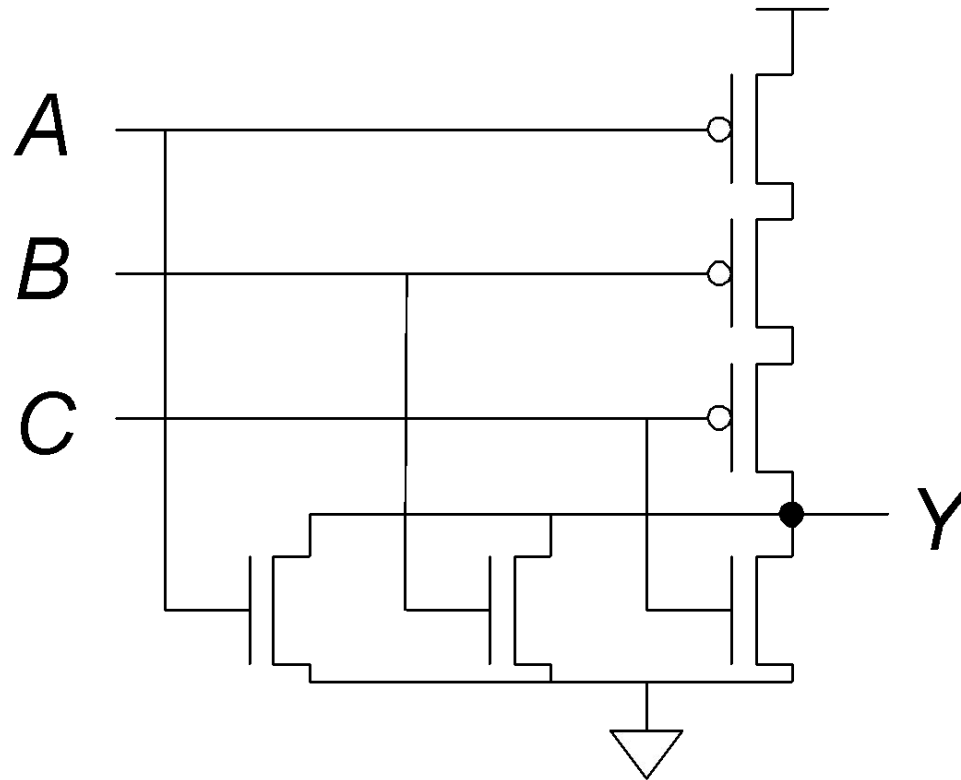
Логический элемент ИЛИ-НЕ

Как построить элемент ИЛИ-НЕ?

ОТ НУЛЯ К
ЕДИНИЦЕ

Элемент ИЛИ-НЕ с тремя входами

ОТ НУЛЯ К
ЕДИНИЦЕ



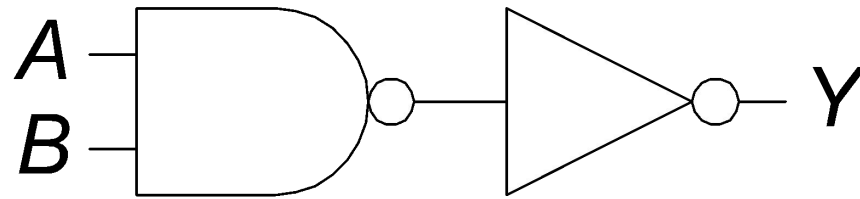
Другие элементы КМОП

Как построить элемент И с двумя входами?

ОТ НУЛЯ К
ЕДИНИЦЕ

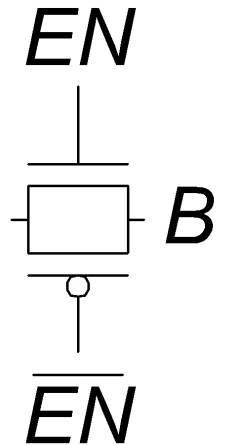
Элемент И с двумя входами

ОТ НУЛЯ К
ЕДИНИЦЕ



Передаточный логический элемент

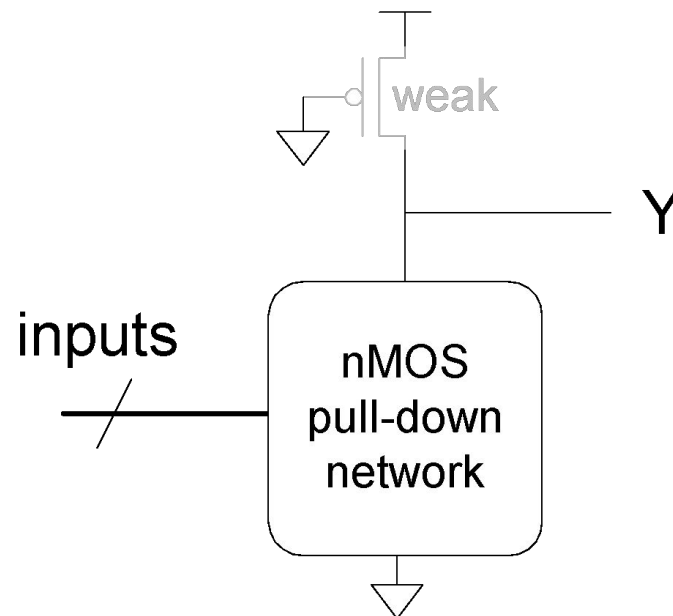
- n-МОП плохо передают 1
- p-МОП плохо передают 0
- Передаточный логический элемент A — лучший выключатель
 - хорошо передает и 0 и 1
- Когда $EN = 1$, выключатель замкнут (ON):
 - $\overline{EN} = 1$ и A соединен с B
- Когда $EN = 0$, выключатель разомкнут (OFF):
 - A и B не соединены



Элементы Псевдо-n-МОП

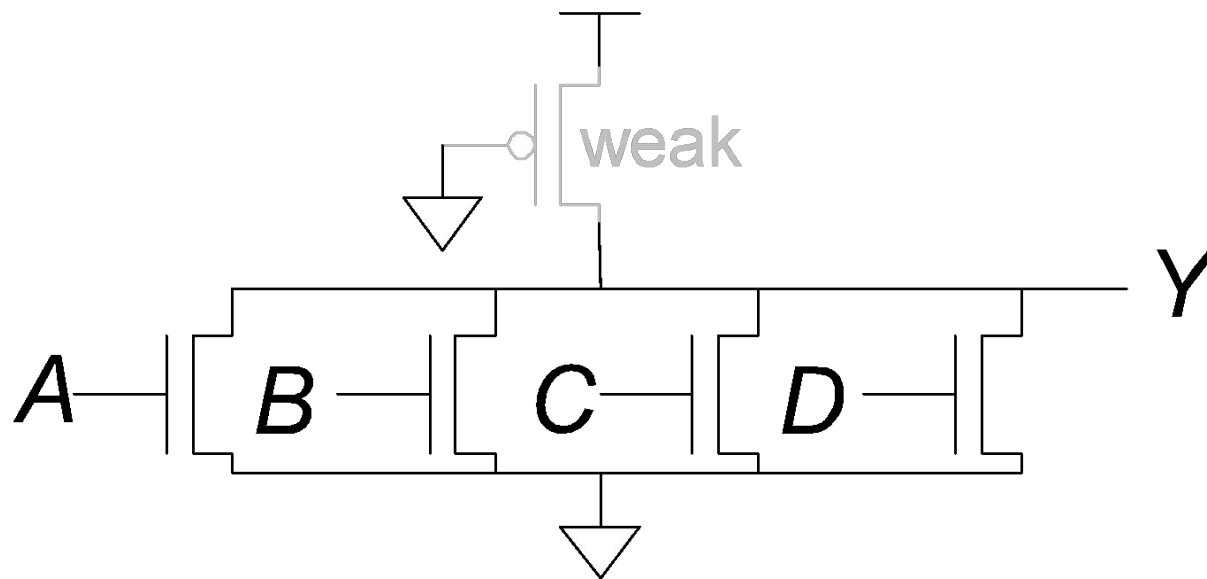
ОТ НУЛЯ К
ЕДИНИЦЕ

- Заменить подтягивающую цепь слабым всегда включенным p-МОП транзистором
- p-МОП транзистор: подтягивает выход к высокому напряжению, только если n-МОП цепь не тянет его к низкому напряжению



Пример элемента Псевдо-n-МОП

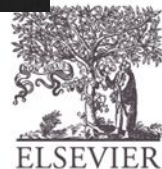
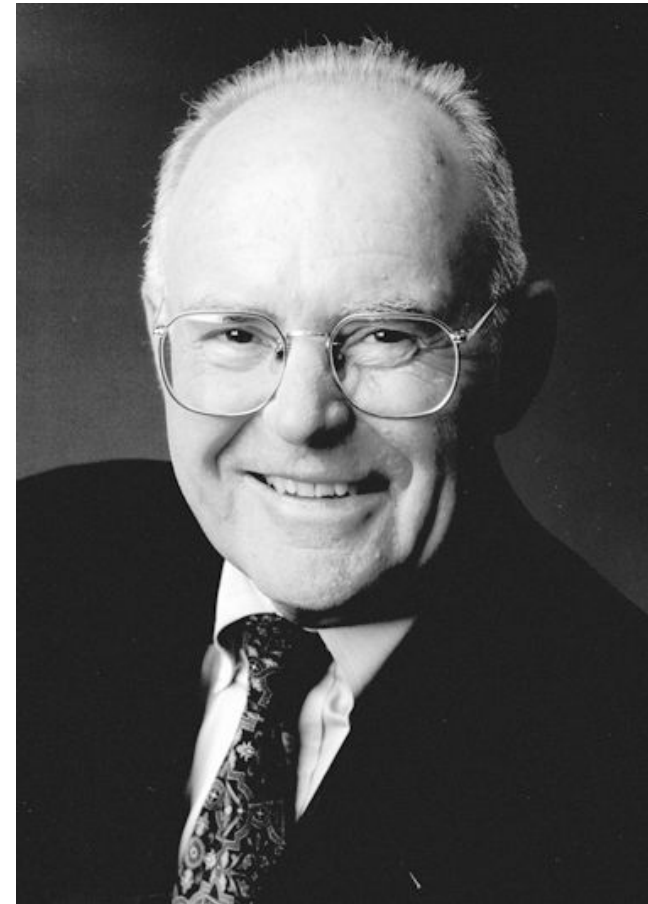
Псевдо-n-МОП элемент NOR4



ОТ НУЛЯ К
ЕДИНИЦЕ

Гордон Мур, 1929-

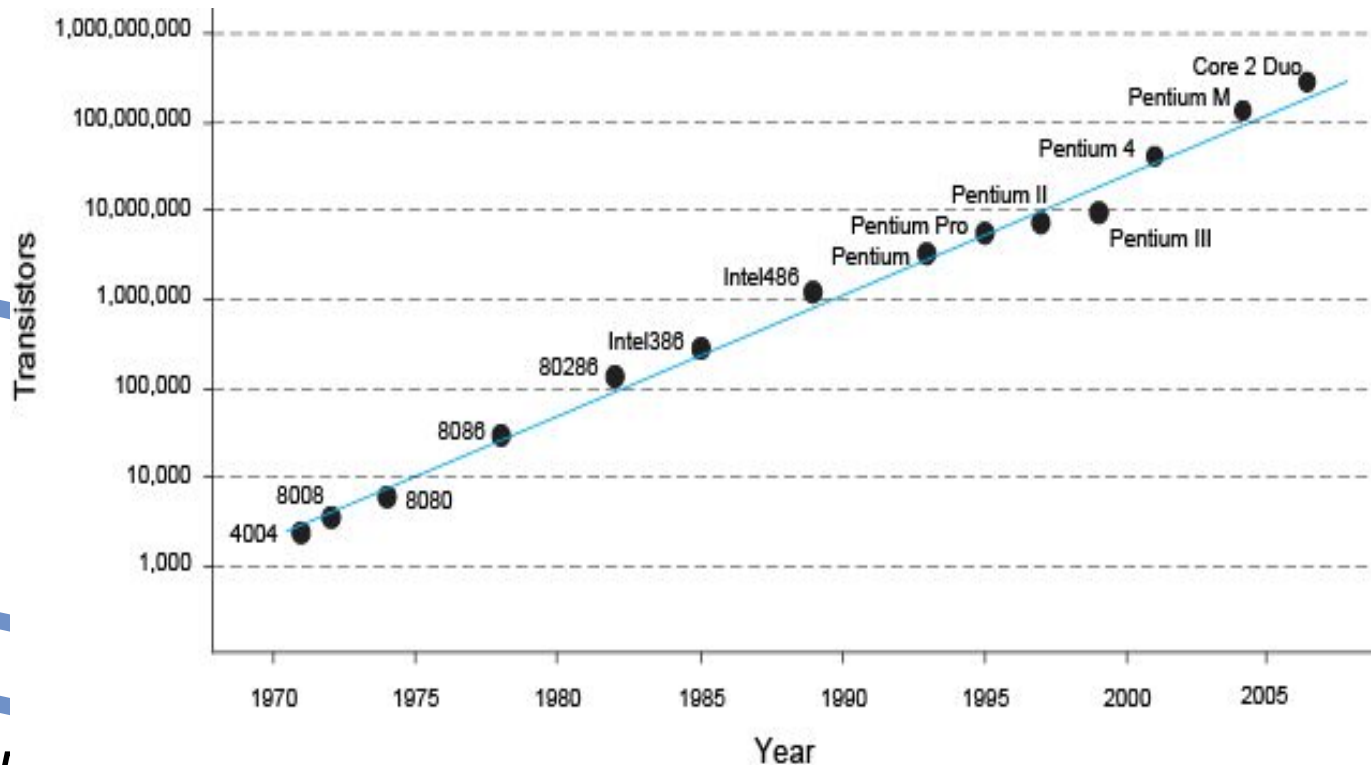
- Со-основатель (вместе с Робертом Нойсом) Intel в 1968 году
- **Закон Мура:** количество транзисторов на микросхеме удваивается каждый год (наблюдался в 1965 году)
- С 1975 года количество транзисторов удваивается каждые два года



Закон Мура:

ОТ НУЛЯ К

ЕДИНИЦЕ



• “Если автомобильная промышленность получила бы такому же циклу развития, как и компьютерная, Rolls-Royce стоил бы сейчас \$100, на одном галлоне бензина проезжал бы миллион миль и взрывался бы раз в году...”

– Robert Cringley

Энергопотребление

- Мощность = Потребление энергии в единицу времени
 - Динамическая потребляемая мощность
 - Статическая потребляемая мощность

ОТ НУЛЯ К
ЕДИНИЦЕ

- **Мощность идет на зарядку емкостей затворов транзисторов**
 - Для зарядки конденсатора емкостью C до напряжения V_{DD} необходима энергия CV_{DD}^2
 - Ток переключается с частотой f : транзистор переключается (от 0 в 1 или наоборот) с такой частотой
 - Конденсатор заряжается $f/2$ раз за секунду (разрядка из 1 в 0 не требует энергии)
- Динамическая потребляемая

МОЩНОСТЬ:

Статическая потребляемая

МОЩНОСТИ

- Мощность, потребляемая, когда элементы не переключаются
- Обусловлена *токами покоя (токами утечки)*, I_{DD}
- Статическая потребляемая мощность:

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

Пример оценки

- Оцените мощность, потребляемую беспроводным переносным компьютером

- $V_{DD} = 1.2 \text{ В}$

- $C = 20 \text{ нФ}$

- $f = 1 \text{ ГГц}$

- $I_{DD} = 20 \text{ мА}$

ОТ НУЛЯ К
ЕДИНИЦЕ

Пример оценки

- Оцените мощность, потребляемую беспроводным переносным компьютером

- $V_{DD} = 1.2 \text{ В}$

- $C = 20 \text{ нФ}$

- $f = 1 \text{ ГГц}$

- $I_{DD} = 20 \text{ мА}$

$$\begin{aligned} P &= \frac{1}{2} C V_{DD}^2 f + I_{DD} V_{DD} \\ &= \frac{1}{2} (20 \text{ нФ}) (1.2 \text{ В})^2 (1 \text{ GHz}) + \\ &\quad (20 \text{ mA}) (1.2 \text{ В}) \\ &= \mathbf{14.4 \text{ W}} \end{aligned}$$