



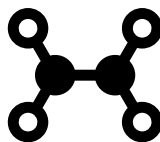
# ОСНОВЫ ЭЛЕКТРОНИКИ

## ЗАНЯТИЕ 12





|                                                   |           |
|---------------------------------------------------|-----------|
| <b>ОСНОВЫ ЦИФРОВОЙ ЛОГИКИ .....</b>               | <b>3</b>  |
| Логические функции и законы алгебры логики .....  | 3         |
| Логические вентили.....                           | 4         |
| Унарные операции .....                            | 5         |
| Логический вентиль НЕ (NOT) .....                 | 5         |
| Буфер (логический повторитель).....               | 5         |
| Бинарные операции .....                           | 5         |
| Логический вентиль И (AND GATE) .....             | 6         |
| Логический вентиль ИЛИ (OR GATE) .....            | 6         |
| Элемент «И-НЕ» (NAND GATE) .....                  | 7         |
| Элемент «ИЛИ-НЕ» (NOR GATE) .....                 | 8         |
| Исключающее ИЛИ (XOR GATE) .....                  | 8         |
| ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ (XNOR GATE).....               | 9         |
| <b>КОМБИНАЦИОННЫЕ ЛОГИЧЕСКИЕ УСТРОЙСТВА .....</b> | <b>10</b> |
| Типовые комбинационные цифровые устройства.....   | 10        |
| Сумматор .....                                    | 10        |
| Шифратор .....                                    | 12        |
| Дешифратор .....                                  | 13        |
| Мультиплексор .....                               | 15        |
| Демультимплексор .....                            | 17        |
| <b>ПОСЛЕДОВАТЕЛЬНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА.....</b>  | <b>18</b> |
| Триггер .....                                     | 18        |
| Регистр.....                                      | 21        |
| Счетчик импульсов .....                           | 22        |
| Цифровые запоминающие устройства (ЗУ).....        | 25        |
| Оперативные запоминающие устройства.....          | 26        |
| Постоянные запоминающие устройства .....          | 27        |



# ОСНОВЫ ЦИФРОВОЙ ЛОГИКИ

Все цифровые устройства построены на элементах, которые выполняют различные логические операции. Для формального описания логической стороны процессов в цифровых устройствах используется алгебра логики (АЛ).

Алгебра логики оперирует двоичными переменными, которые условно обозначаются как логический ноль (лог. 0) и логическая единица (лог. 1). Между обычной, привычной нам алгеброй и алгеброй логики существуют существенные различия в отношении количества и характера операций, а также законов, которым они подчиняются.

## ЛОГИЧЕСКИЕ ФУНКЦИИ И ЗАКОНЫ АЛГЕБРЫ ЛОГИКИ

Логическая, или булева, функция от двоичных переменных принимает только два значения: логический ноль (лог. 0) и логическая единица (лог. 1). Для задания логической функции (ЛФ) обычно используются два способа: аналитический (запись формулой) и табличный.

При табличном способе строится таблица истинности, в которой перечисляются все возможные сочетания значений аргументов и соответствующие им значения логической функции. Поскольку число таких сочетаний ограничено, таблица истинности позволяет определить значение функции для любых значений аргументов.

Количество возможных наборов из  $n$  аргументов составляет  $2^n$ , и количество различных функций составляет  $2^{2^n}$ . Так, для одного аргумента существует 4 различных логических функции, для двух аргументов их уже 16, и так далее.

Аналитический способ представления логической функции предполагает ее запись в виде математического выражения, которое связывает аргументы логической функции с использованием определенных математических (логических) операций. Это выражение имеет вид  $f(x_1, x_2, \dots, x_n)$ , где  $x_1, x_2, \dots, x_n$  - аргументы логической функции, а  $f(x_1, x_2, \dots, x_n)$  - выражение, определяющее значение функции в зависимости от значений аргументов.

В аналитическом представлении логической функции используются логические операции, такие как логическое И (AND), логическое ИЛИ (OR), логическое НЕ (NOT), их комбинации и другие. Эти операции позволяют строить сложные логические выражения, которые описывают поведение функции при различных значениях аргументов.

# ЛОГИЧЕСКИЕ ВЕНТИЛИ

Логические вентили (рис. 1) представляют собой простейшие цифровые схемы. Они получают один или несколько двоичных сигналов на входе и генерируют новый двоичный сигнал на выходе. Взаимосвязь между входными сигналами и выходным сигналом логического вентиля может быть описана с помощью таблицы истинности или уравнения Булевой логики. В таблице истинности (рис. 2) слева указываются значения входных сигналов, а справа представлено значение соответствующего выходного сигнала. Каждая строка в такой таблице соответствует одной из возможных комбинаций входных сигналов.

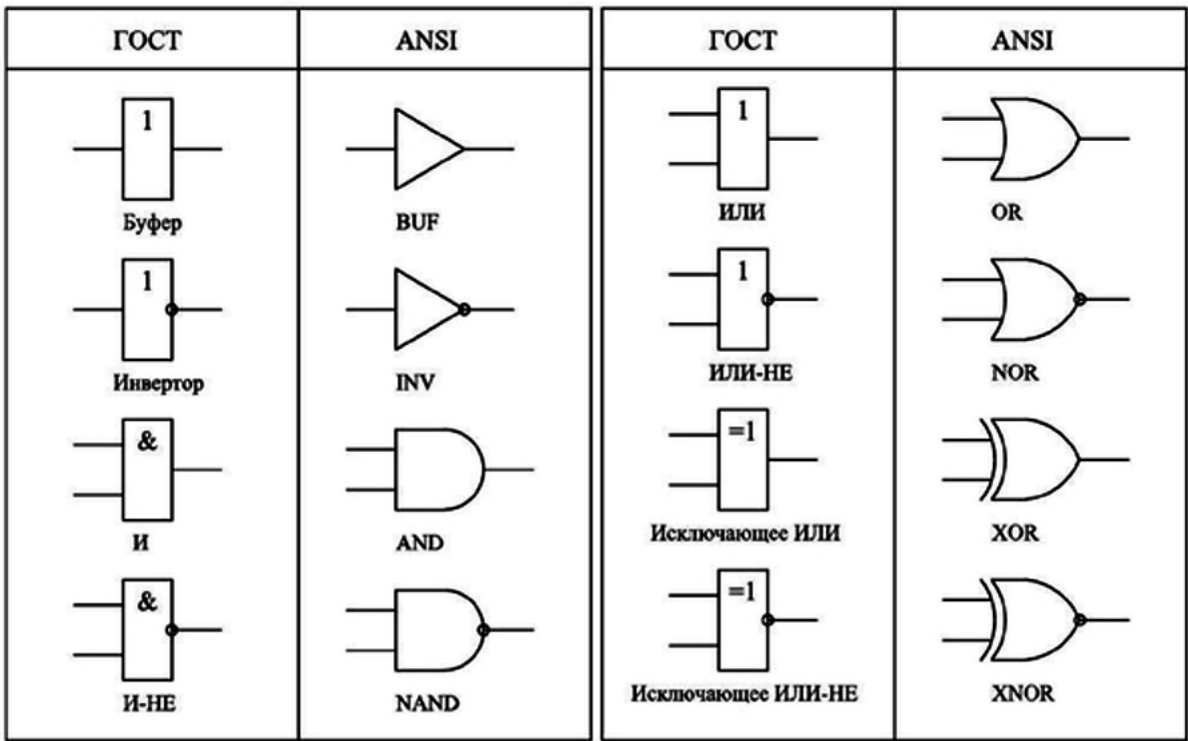


Рис. 1

| Входы |   | Выходы |     |                 |      |        |                    |
|-------|---|--------|-----|-----------------|------|--------|--------------------|
| A     | B | И      | ИЛИ | Исключающее ИЛИ | И-НЕ | ИЛИ-НЕ | Исключающее ИЛИ-НЕ |
| 0     | 0 | 0      | 0   | 0               | 1    | 1      | 1                  |
| 0     | 1 | 0      | 1   | 1               | 1    | 0      | 0                  |
| 1     | 0 | 0      | 1   | 1               | 1    | 0      | 0                  |
| 1     | 1 | 1      | 1   | 0               | 0    | 0      | 1                  |

Рис. 2

Электрическая схема, спроектированная для выполнения конкретной логической операции с входными данными, называется логическим элементом. В данном контексте входные данные интерпретируются как различные уровни напряжения, а результат логической операции на выходе также представляется в виде напряжения определенного уровня. Это основной принцип работы цифровых схем, используемых в цифровой электронике.

## УНАРНЫЕ ОПЕРАЦИИ

Логические операции с одним операндом называются **унарными**. В основном применяются две операции: операция отрицания (инверсии) и повторения. Операция отрицания имеет большую значимость, так как повторитель может быть собран из двух инверторов, а инвертор из повторителей не собрать.

### ЛОГИЧЕСКИЙ ВЕНТИЛЬ *НЕ* (*NOT*)

Функцией этого элемента является инверсия. Иными словами, если на вход поступает лог. 1, то на выходе будет лог. 0, и наоборот. Кругик на выходе этого логического элемента, который часто называют "пузырем" (bubble) в англоязычной литературе, указывает на инверсию сигнала.

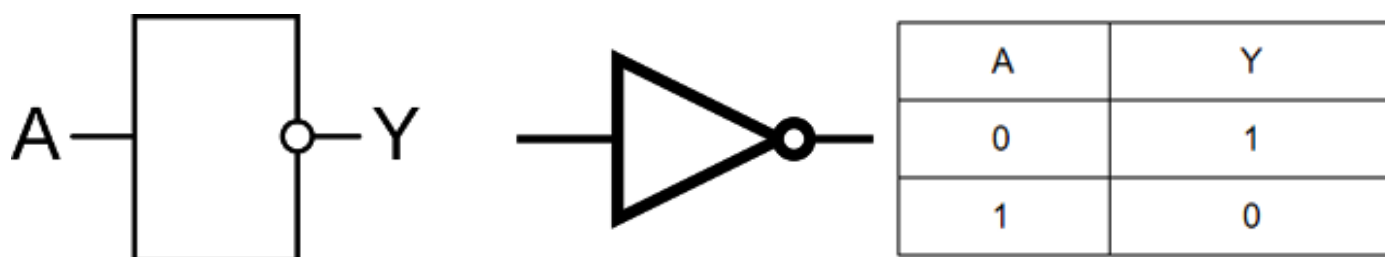


Рис. 3

Логическое выражение:  $Y = \bar{A}$

### БУФЕР (ЛОГИЧЕСКИЙ ПОВТОРИТЕЛЬ)

Другим примером логического вентиля с одним входом является буфер. Буфер просто копирует входной сигнал на выход, например, это может быть нужно для передачи большего тока последующему потребителю, чем это может дать источник сигнала, или для быстрой передачи сигнала сразу нескольким логическим элементам. В логических схемах буфер обозначается треугольником.

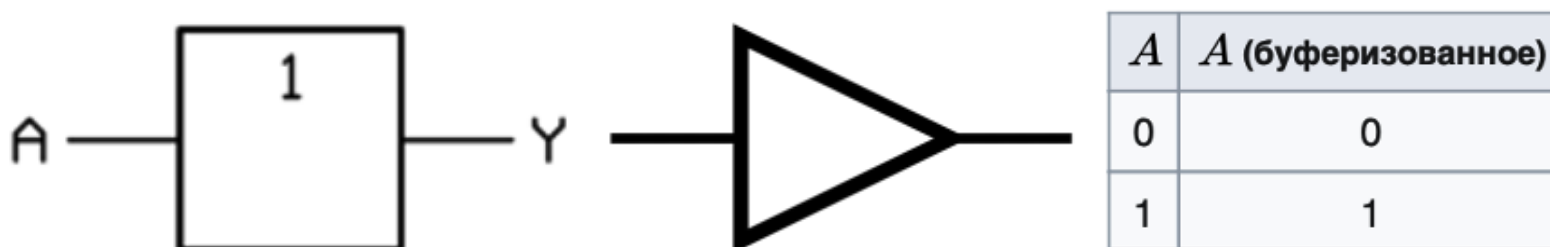


Рис. 4

Логическое выражение:  $Y = A$

## БИНАРНЫЕ ОПЕРАЦИИ

Логические операции с двумя операндами называются **бинарными**. Логические вентили с двумя входными сигналами позволяют выполнять более широкий спектр функциональных задач. Кроме того, если имеется большое количество входных сигналов, их обработку часто можно разделить на последовательность операций с двумя сигналами.

## ЛОГИЧЕСКИЙ ВЕНТИЛЬ И (AND GATE)

Этот элемент выполняет операцию "И", и на его выходе будет лог. 1 только в том случае, если оба входных сигнала А и В имеют значение лог. 1. В противном случае, выходной сигнал Y будет иметь значение лог. 0.

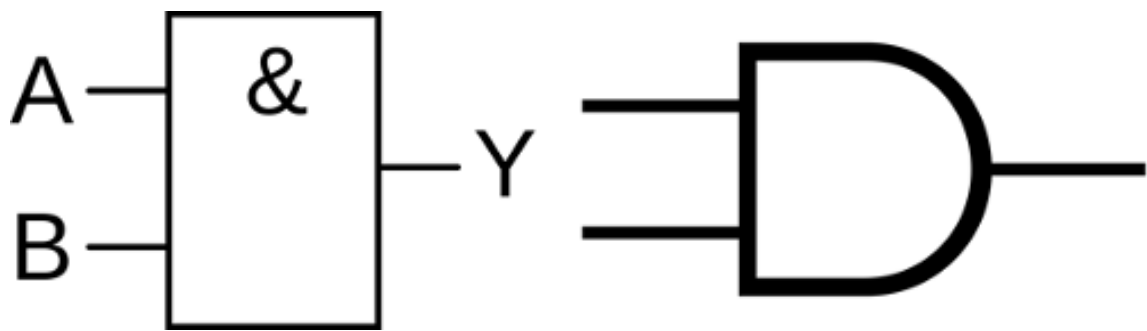


Рис. 5

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Таблица истинности AND

Логическое выражение:  $Y = A \cdot B = A \cap B$

## ЛОГИЧЕСКИЙ ВЕНТИЛЬ ИЛИ (OR GATE)

Этот элемент выполняет операцию "ИЛИ", и на его выходе будет лог. 1, если хотя бы один из двух входных сигналов А или В имеет значение лог. 1.

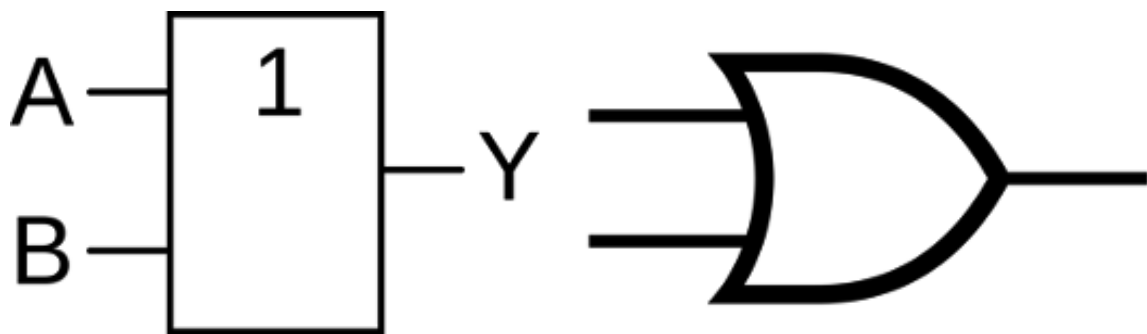


Рис. 6

| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

Таблица истинности OR

Логическое выражение:  $Y = A + B = A \cup B$

Символ  $\cup$  читается как "объединение". Разработчики цифровых систем часто используют простой символ "+". Математически это выражение можно интерпретировать как "Y равно A или B".

## ЭЛЕМЕНТ «И-НЕ» (NAND GATE)

Элемент "И-НЕ" работает как операция «И» (рис. 5), но инвертирует свой выходной сигнал. Таким образом, если у элемента "И" на выходе был бы лог. 0, то у элемента "И-НЕ" будет лог. 1, и наоборот.

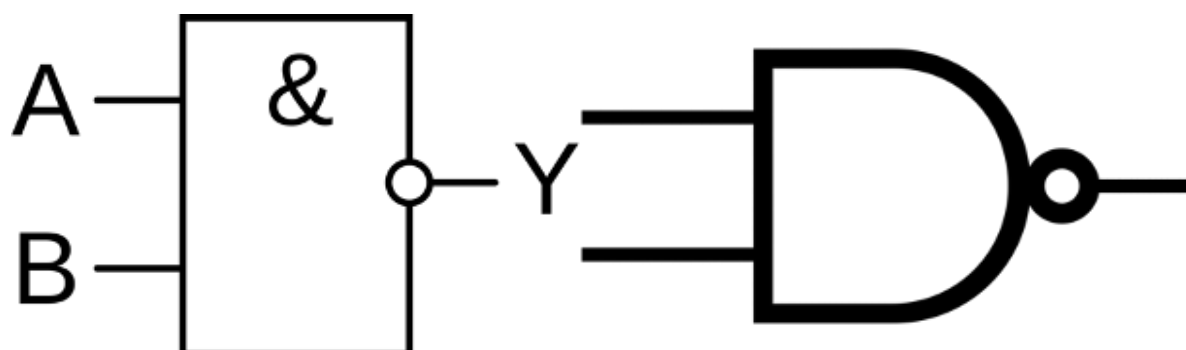


Рис. 7

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Таблица истинности NAND

Логическое выражение:  $Y = \overline{A \cdot B}$

**ЭЛЕМЕНТ «ИЛИ-НЕ» (NOR GATE)**

А это уже знакомый элемент «ИЛИ», но уже с инвертором на выходе. При появлении лог. 1 на любом из входов или на обоих, то на выходе установится лог. 0, и наоборот – лог. 1 на выходе говорит о том, что на всех входах низкий логический уровень.

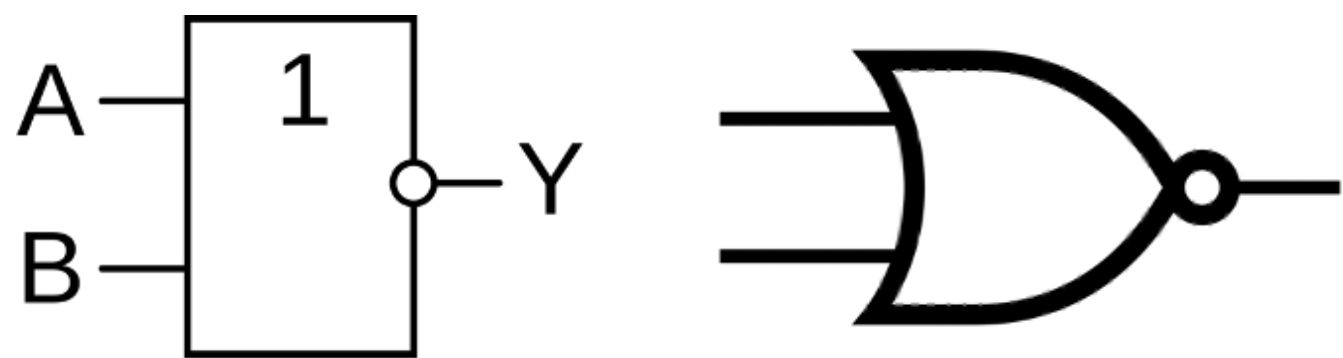


Рис. 8

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

Таблица истинности NOR

Логическое выражение:  $Y = \overline{A \cdot B}$

**ИСКЛЮЧАЮЩЕЕ ИЛИ (XOR GATE)**

«Исключающее ИЛИ» (XOR) для двух входов A и B даёт на выходе Y лог. 1 только в том случае, когда значения этих входов разнятся, а вот если они совпадают, результатом на выходе будет лог. 0

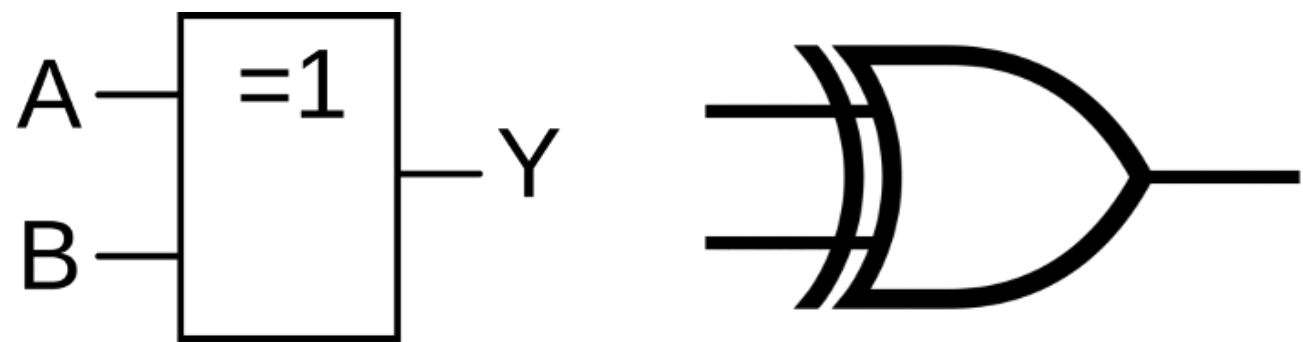


Рис. 9

Логическое выражение:  $Y = A \oplus B$



| A | B | Y |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Таблица истинности XOR

## ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ (XNOR GATE)

Логический элемент ИСКЛЮЧАЮЩЕЕ ИЛИ-НЕ, иногда называемый элементом контроля по четности (parity gate), выдает на выход сигнал лог. 1, если нечетное количество входных сигналов имеют значение лог. 1. Для двухвходового варианта выходной сигнал принимает значение логической единицы, если уровни входных сигналов совпадают (оба логических нуля или обе логические единицы). Стоит нарушить

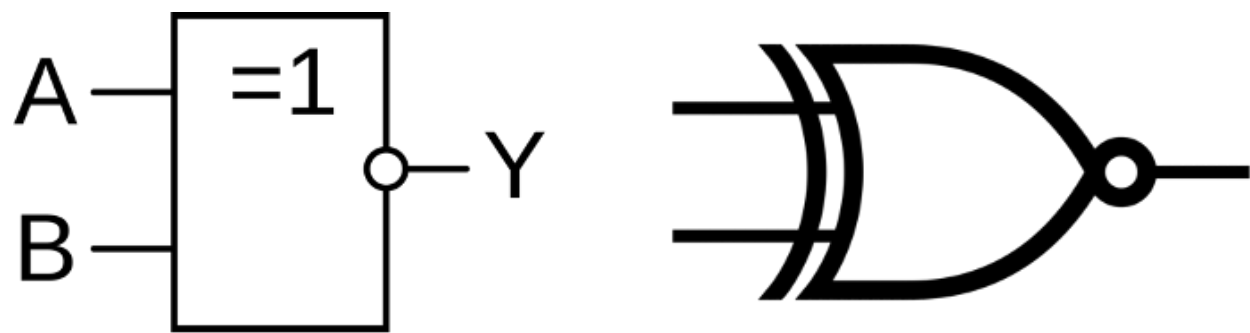


Рис. 10

| A | B | Y |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Таблица истинности XNOR

это условие, сигнал на выходе элемента примет значение логического нуля.

Логическое выражение:  $Y = A \sim B$

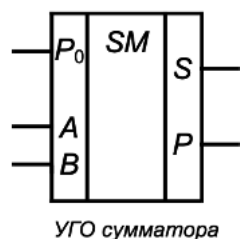
# КОМБИНАЦИОННЫЕ ЛОГИЧЕСКИЕ УСТРОЙСТВА

На основе логических принципов и элементов из предыдущего раздела построены комбинационные логические устройства, состояние которых полностью зависит от входящих логических сигналов. Устройство называют комбинационным, если его выходные сигналы в некоторый момент времени однозначно определяются входными сигналами, имеющими место в этот момент времени.

## ТИПОВЫЕ КОМБИНАЦИОННЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

При разработке сложных устройств широко используются не только отдельные логические элементы, которые реализуют элементарные булевы функции, но также их комбинации в виде типовых структур, которые действуют как единое целое и реализуются в виде интегральных микросхем. В таких структурах на вход могут подаваться информационные логические сигналы, а также сигналы управления. Сигналы управления могут определять, например, порядок передачи информационных входных сигналов на выход или выполнять функцию сигналов синхронизации. Внутренняя структура комбинационных логических устройств (КЦУ) часто описывается в справочниках, и для разработчика важно знать таблицу истинности и принцип преобразования входных сигналов в выходные.

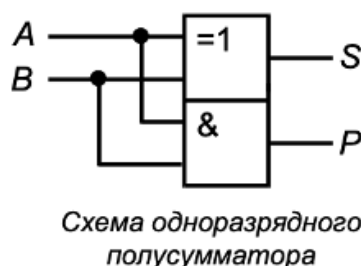
## СУММАТОР



Сумматор — это логическая электронная схема, предназначенная для выполнения операции сложения двоичных чисел. Сумматор является главной составляющей процессора.

Одноразрядный сумматор может обычно иметь три входа: A и B, которые представляют собой слагаемые, и  $C_{in}$  (иногда обозначается как  $P_0$ ), который представляет перенос из предыдущего разряда. На выходе у одноразрядного сумматора обычно два сигнала: S, который представляет собой сумму, и P, который представляет собой перенос, который может передаваться следующему разряду.

A, B – одноразрядные числа;  
S – сумма;  
P – перенос.



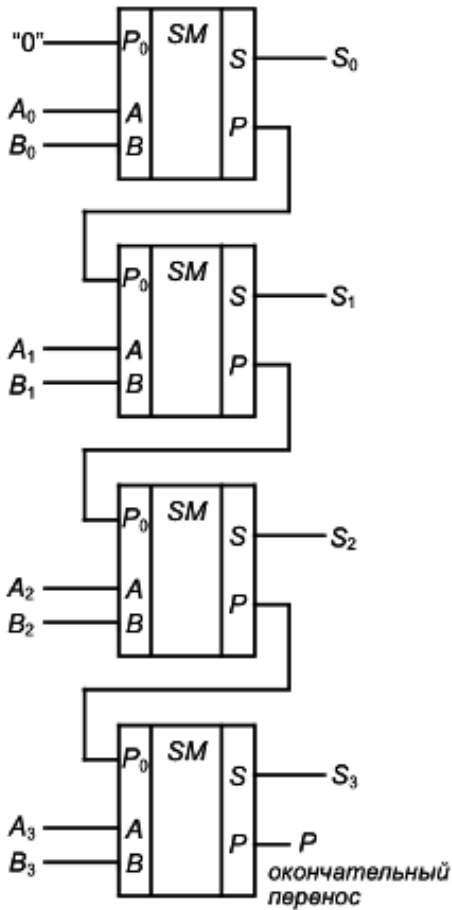
Логическое выражение для S:  $S = \overline{A}B + A\overline{B} = A \oplus B$

Логическое выражение для P:  $P = A \cdot B$

| A | B | P <sub>0</sub> | P | S |
|---|---|----------------|---|---|
| 0 | 0 | 0              | 0 | 0 |
| 0 | 1 | 0              | 0 | 1 |
| 1 | 0 | 0              | 0 | 1 |
| 1 | 1 | 0              | 1 | 0 |
| 0 | 0 | 1              | 0 | 1 |
| 0 | 1 | 1              | 1 | 0 |
| 1 | 0 | 1              | 1 | 0 |
| 1 | 1 | 1              | 1 | 1 |

Таблица сложения

Процессоры, как правило, выполняют операции с многоразрядными двоичными числами. Например, чтобы сложить 1012 и 1102, необходим многоразрядный сумматор. В нем каждому разряду соответствует одноразрядный сумматор, а выходной перенос младшего разряда подключается к входу старшего разряда. Полный сумматор, в отличие от полусумматора, учитывает результат предыдущего сложения и для этого имеет вход для переноса из предыдущего разряда.

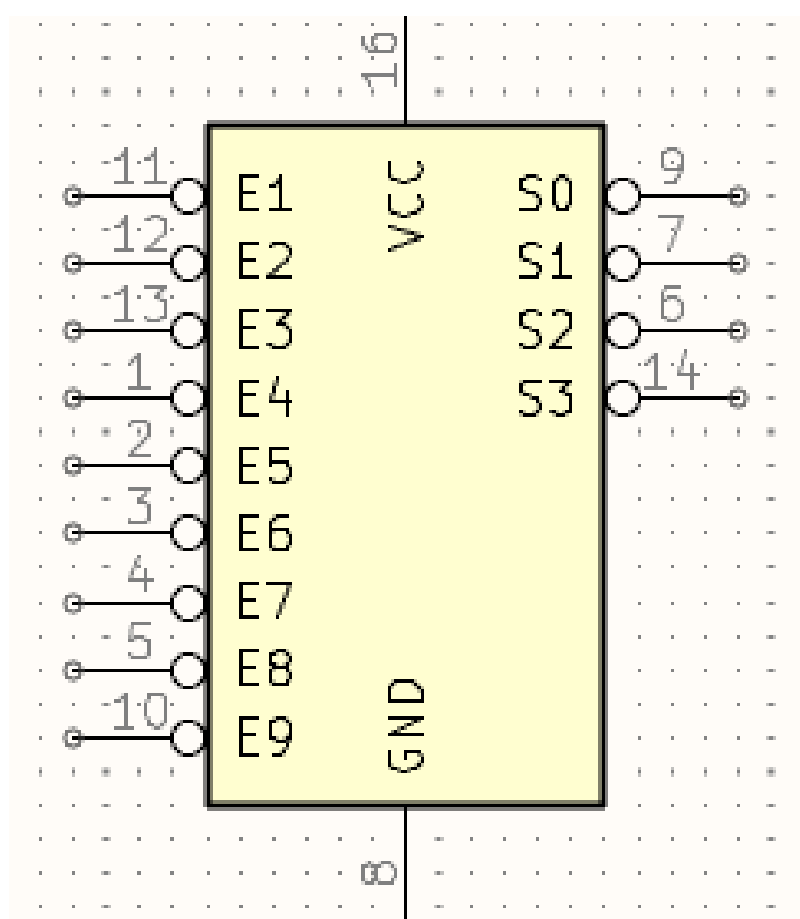


Многоразрядный сумматор

Многоразрядные сумматоры строятся на основе одnorазрядных полных сумматоров. Параллельный сумматор предназначен для сложения двух 4-х разрядных чисел. Для увеличения разрядности сумматора требуется окончательный перенос. В представленной схеме перенос выполняется последовательно из разряда в разряд, что занимает определенное время. При суммировании многоразрядных чисел это время становится значительным и определяет общее время на выполнение операции сложения. С целью сокращения времени суммирования вместо последовательного переноса используется параллельный перенос. Такой метод реализован, например, в 4-х разрядном сумматоре SN74LS283. Интегральные микросхемы доступны в вариантах с одним, двумя и четырьмя разрядами для выполнения двоичного сложения.

## ШИФРАТОР

Шифратор — это комбинационное устройство, предназначенное для преобразования десятичных чисел в двоичную систему счисления. Каждому входу в шифраторе может быть сопоставлено десятичное число, и на выходе формируется соответствующий двоичный код, представленный набором логических сигналов.



УГО шифратора

Из таблицы ниже видно, что на выходах S0, S1, S2, S3 формируется двоичный код, соответствующий номеру входной линии (E1, E2, ..., E9), на которую поступает входной сигнал. Если на нескольких входных линиях обычного шифратора поступают сигналы одновременно, это может вызвать неопределенность на выходах. Шифратор также иногда называют "кодером" (от английского слова "Coder") и применяют, например, для перевода десятичных чисел, введенных с клавиатуры, в соответствующий двоичный код. На



практике чаще используется приоритетный шифратор, который позволяет одновременно подавать входной сигнал на несколько входов. Однако на выходе будет сформирован двоичный код, соответствующий наибольшему (старшему) номеру из входов, на которые подан сигнал.

Другими словами, приоритетный шифратор работает так, будто сигнал подается только на один из входов, и вход с наибольшим номером имеет приоритет. Приоритетные шифраторы широко применяются, например, в телефонных клавиатурах. Примером приоритетного шифратора является микросхема SN74LS148.

| Входы |    |    |    |    |    |    |    |    | Выходы |    |    |    |
|-------|----|----|----|----|----|----|----|----|--------|----|----|----|
| E1    | E2 | E3 | E4 | E5 | E6 | E7 | E8 | E9 | S3     | S2 | S1 | S0 |
| 1     | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1      | 1  | 1  | 1  |
| X     | X  | X  | X  | X  | X  | X  | X  | 0  | 0      | 1  | 1  | 0  |
| X     | X  | X  | X  | X  | X  | X  | 0  | 1  | 0      | 1  | 1  | 1  |
| X     | X  | X  | X  | X  | X  | 0  | 1  | 1  | 1      | 0  | 0  | 0  |
| X     | X  | X  | X  | X  | 0  | 1  | 1  | 1  | 1      | 0  | 0  | 1  |
| X     | X  | X  | X  | 0  | 1  | 1  | 1  | 1  | 1      | 0  | 1  | 0  |
| X     | X  | X  | 0  | 1  | 1  | 1  | 1  | 1  | 1      | 0  | 1  | 1  |
| X     | X  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1      | 1  | 0  | 0  |
| X     | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1      | 1  | 0  | 1  |
| 0     | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1      | 1  | 1  | 0  |

1- высокий уровень, 0- низкий уровень, X- не важно. Так как входы и выходы данного шифратора инверсные, то активным уровнем и на входе, и на выходе будет являться логический 0!

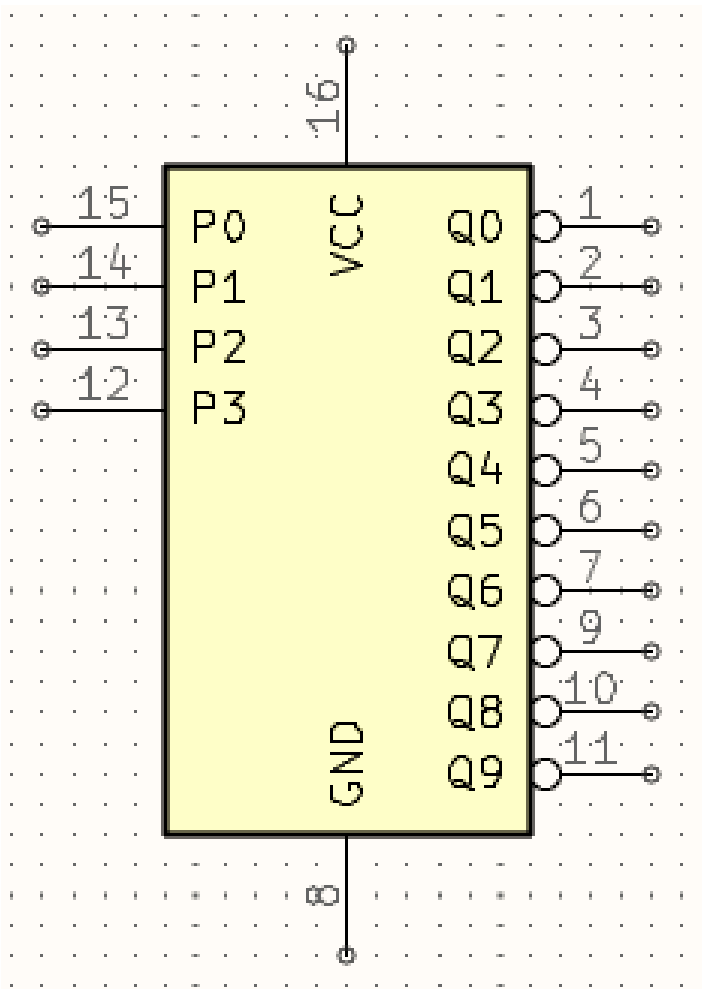
## ДЕШИФРАТОР

Дешифратором называется комбинационное устройство, которое выполняет преобразование n-разрядного двоичного кода в логический сигнал, который появляется на выходе с десятичным номером, соответствующим этому двоичному коду.

Дешифратор всегда активирует только один выход. Это легко заметить в таблице истинности, расположенной ниже, так как активируется тот выход, адрес которого установлен на входах дешифратора.

Дешифраторы широко применялись в цифровой аппаратуре. Однако в настоящее время они редко используются для индикации, так как их роль в этой области заменена другими

съемными решениями. В основном дешифраторы используются как часть более сложных цифровых модулей.



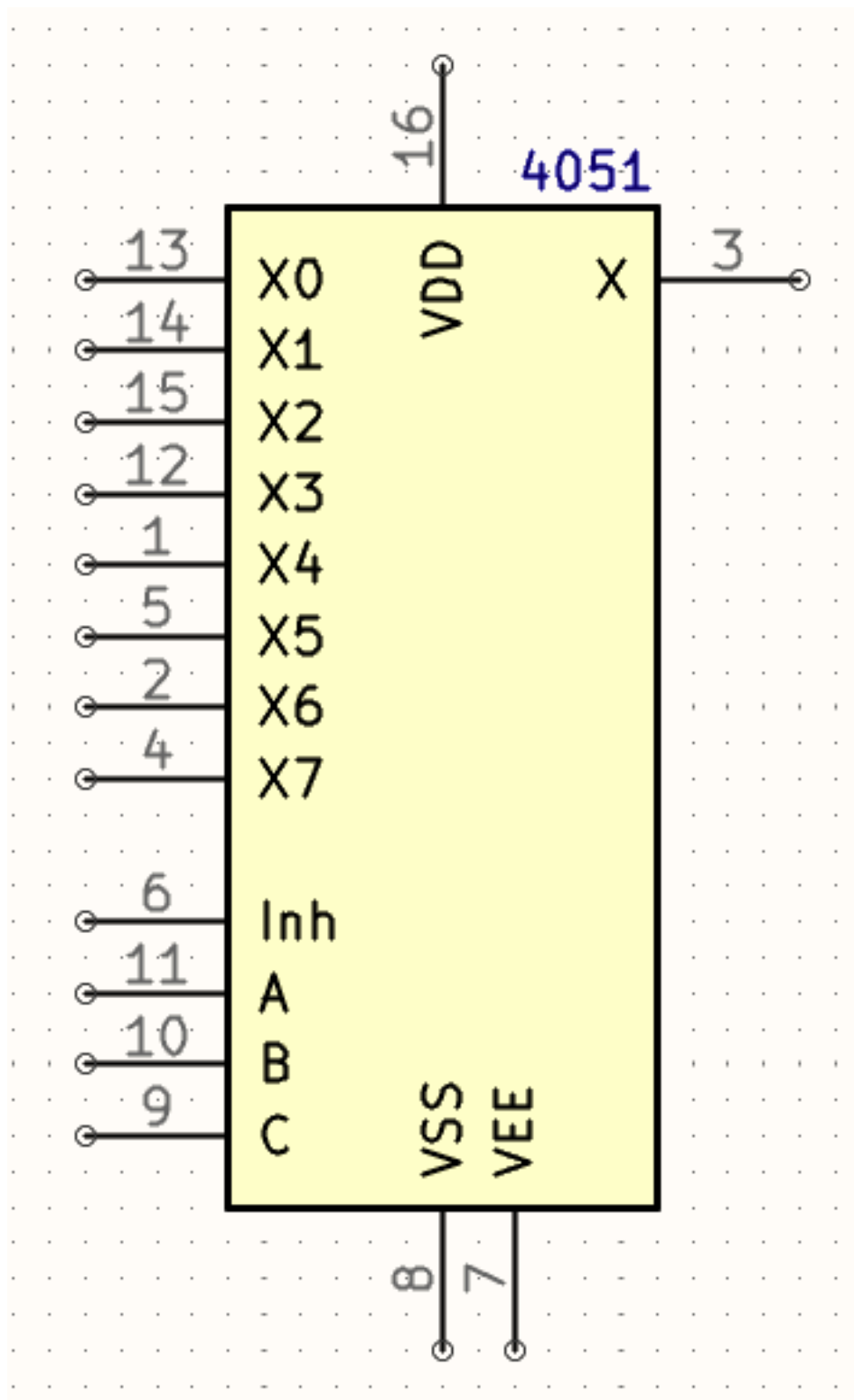
УГО дешифратора

| Входы |    |    |    | Выходы |    |    |    |    |    |    |    |    |    |
|-------|----|----|----|--------|----|----|----|----|----|----|----|----|----|
| P3    | P2 | P1 | P0 | Q0     | Q1 | Q2 | Q3 | Q4 | Q5 | Q6 | Q7 | Q8 | Q9 |
| 0     | 0  | 0  | 0  | 0      | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| 0     | 0  | 0  | 1  | 1      | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| 0     | 0  | 1  | 0  | 1      | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |
| 0     | 0  | 1  | 1  | 1      | 1  | 1  | 0  | 1  | 1  | 1  | 1  | 1  | 1  |
| 0     | 1  | 0  | 0  | 1      | 1  | 1  | 1  | 0  | 1  | 1  | 1  | 1  | 1  |
| 0     | 1  | 0  | 1  | 1      | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1  | 1  |
| 0     | 1  | 1  | 0  | 1      | 1  | 1  | 1  | 1  | 1  | 0  | 1  | 1  | 1  |
| 0     | 1  | 1  | 1  | 1      | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 1  | 1  |
| 1     | 0  | 0  | 0  | 1      | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  | 1  |
| 1     | 0  | 0  | 1  | 1      | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 0  |

Таблица истинности дешифратора

1- высокий уровень, 0- низкий уровень. Выходы дешифратора инверсные, активным уровнем является логический 0, входы не инверсные, соответственно активным уровнем является логическая 1!

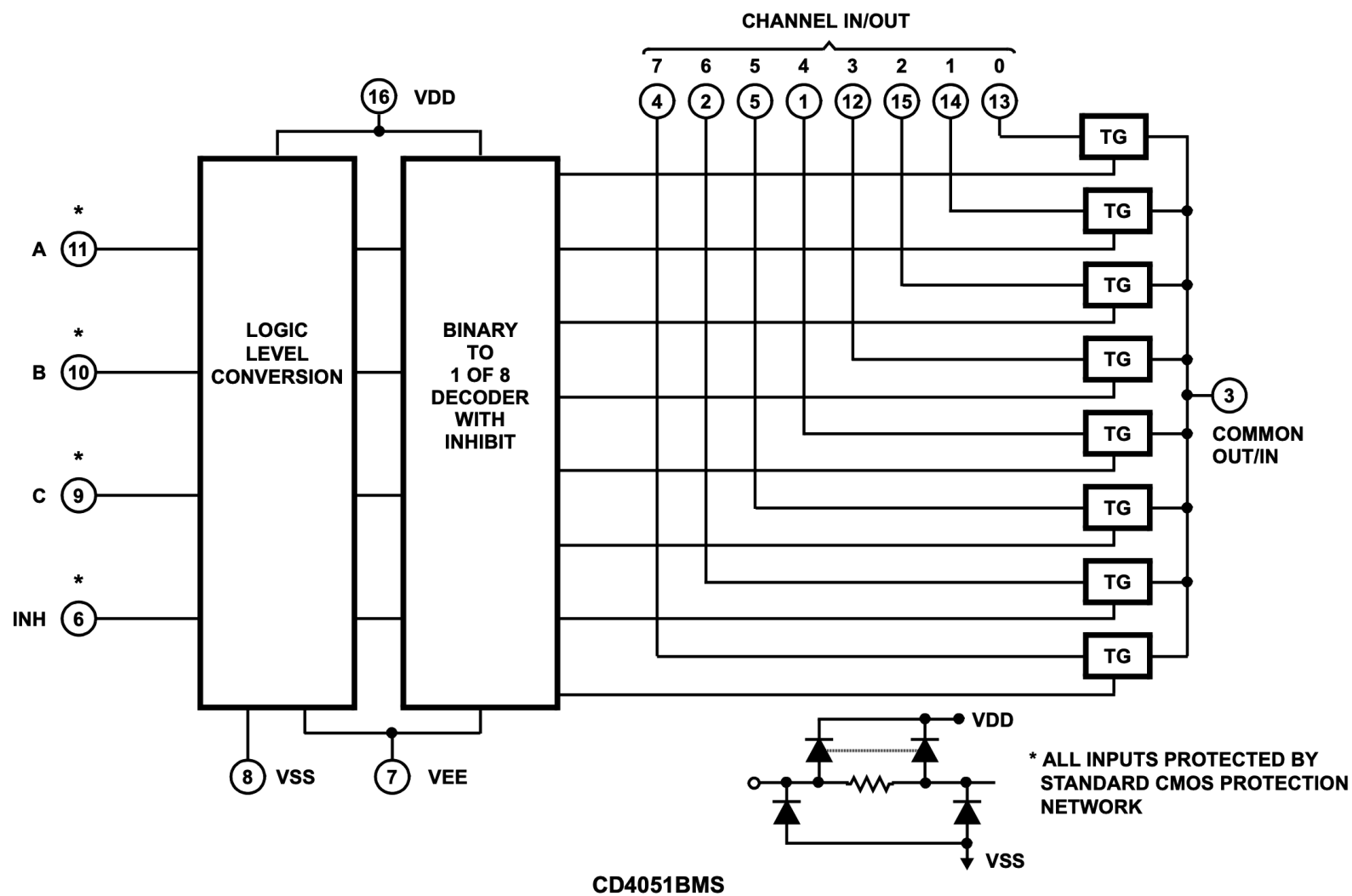
## МУЛЬТИПЛЕКСОР



УГО мультимплексор CD4051B

Мультиплексор — это устройство, которое имеет несколько сигнальных входов, один или более управляющих входов и один выход. Мультиплексор позволяет переключать сигнал с одного из входов на выход, и выбор нужного входа осуществляется путем подачи соответствующей комбинации управляющих сигналов.

Подключение к различным источникам сигналов может быть осуществлено последовательно или по выбору. Выбор информационного входа производится через адресные входы, обычно обозначаемых как А, В, С и т.д., путем задания двоичного соответствующего кода: 000, 100, 011 и т.д.



Функциональная схема

| INPUT STATES     |          |          |          | "ON" CHANNEL(S) |
|------------------|----------|----------|----------|-----------------|
| <b>CD4051BMS</b> |          |          |          |                 |
| <b>INHIBIT</b>   | <b>C</b> | <b>B</b> | <b>A</b> |                 |
| 0                | 0        | 0        | 0        | 0               |
| 0                | 0        | 0        | 1        | 1               |
| 0                | 0        | 1        | 0        | 2               |
| 0                | 0        | 1        | 1        | 3               |
| 0                | 1        | 0        | 0        | 4               |
| 0                | 1        | 0        | 1        | 5               |
| 0                | 1        | 1        | 0        | 6               |
| 0                | 1        | 1        | 1        | 7               |
| 1                | X        | X        | X        | NONE            |

Таблица истинности

Мультиплексор может быть использован для организации передачи сигналов от разных источников через один и выход, но это происходит не одновременно. Выход



мультиплексора представляет собой канал с временным разделением сигналов. Это позволяет эффективно передавать информацию от разных источников через общий канал, но каждый источник может передавать данные только в определенный момент времени.

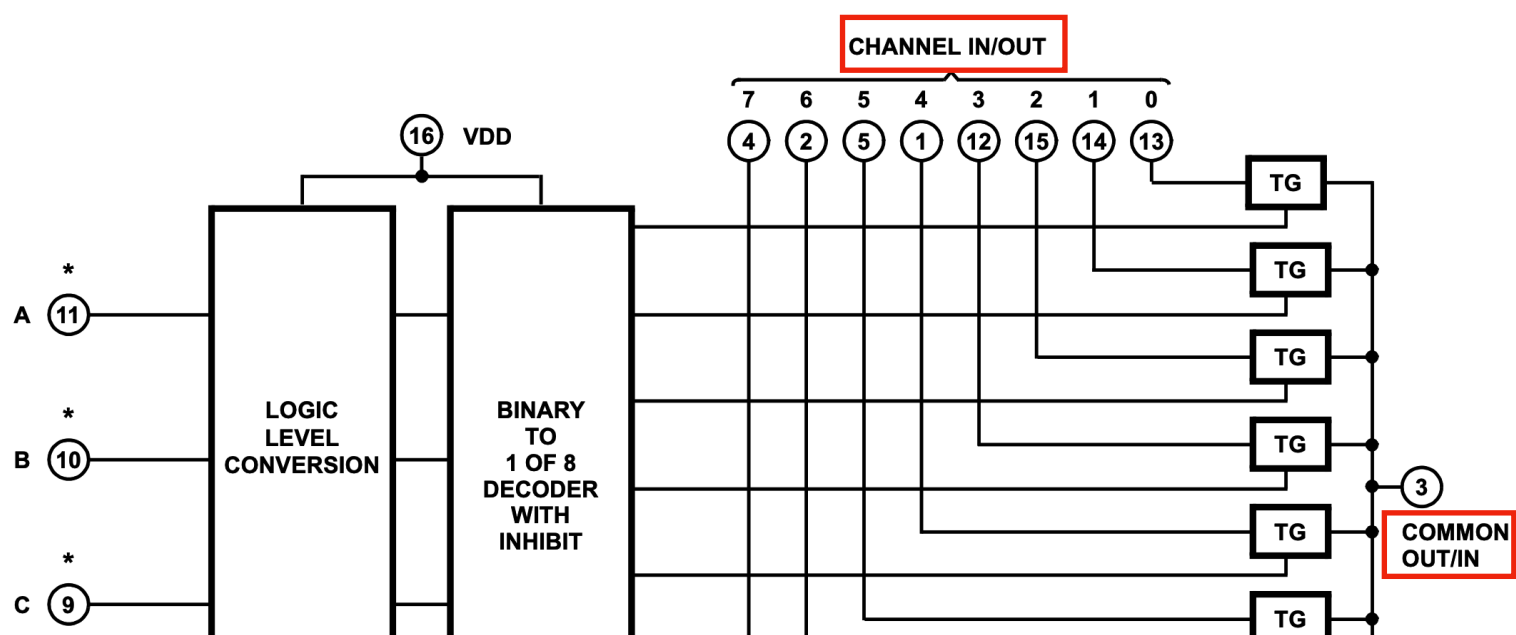
После того как информация была передана одним каналом связи, ее необходимо разделить между соответствующими приемниками. Эту задачу решает демультиплексор, который принимает входящий сигнал и направляет его на соответствующий выход или приемник в зависимости от управляющих сигналов.

## ДЕМУЛЬТИПЛЕКСОР

Демультиплексор — это логическое устройство, предназначенное для переключения сигнала с одного информационного входа на один из информационных выходов. Функции демультиплексоров схожи с функциями дешифраторов.

Дешифратор можно рассматривать как демультиплексор, у которого информационный вход поддерживает состояние выходов в активном состоянии, а адресные входы выполняют роль входов дешифратора. Таким образом, демультиплексор и дешифратор оба выполняют функцию переключения сигнала на один из выходов на основе управляющих входных сигналов, и различие между ними зависит от спецификации их использования в конкретных схемах.

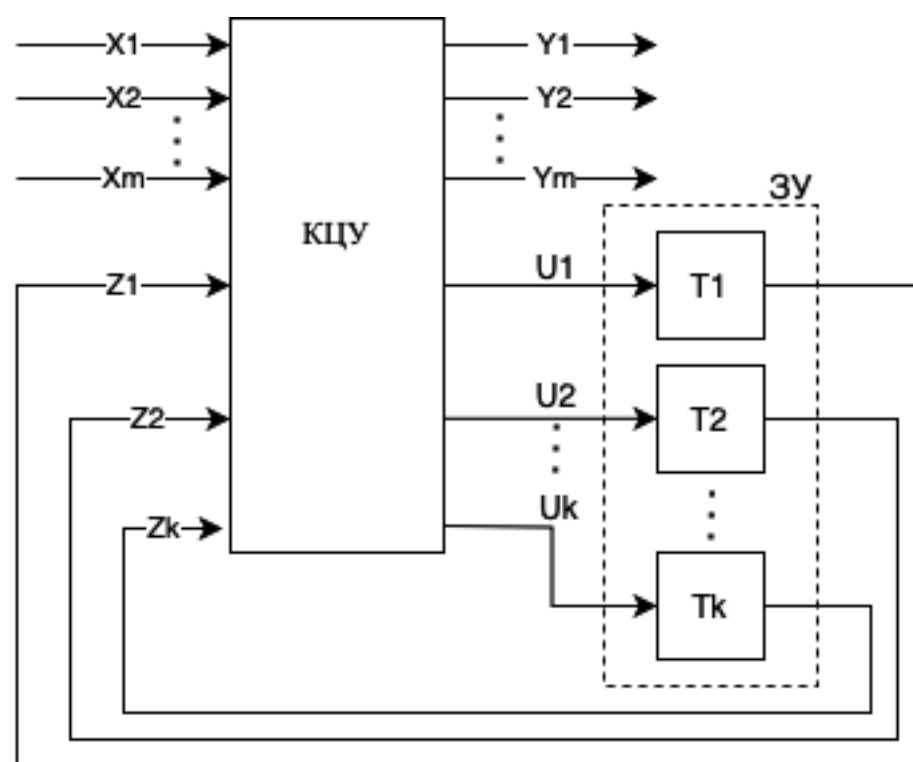
Если обратиться к мультиплексору CD4051, рассмотренному выше, то он умеет выполнять функции демультиплексора, что явно следует из функциональной схемы устройства. То есть присутствует возможность трансляции входящего на 3 вывод сигнала на один из выводов 0...7 данной микросхемы.



Входы и выходы IN/OUT

# ПОСЛЕДОВАТЕЛЬНЫЕ ЦИФРОВЫЕ УСТРОЙСТВА

Последовательные цифровые устройства (ПЦУ) отличаются тем, что их выходные сигналы зависят не только от текущих значений входных сигналов, но также от последовательности значений входных сигналов, поступивших на входы в предыдущие моменты времени. То есть, состояние устройства в текущий момент времени зависит от истории последовательности входных сигналов, которая предшествовала этому моменту. Это свойство делает последовательные цифровые устройства полезными для обработки



Структурная схема ПЦУ

информации, где важен контекст и история входных данных.

КЦУ — комбинационное цифровое устройство, ЗУ — запоминающее устройство, T1, T2 ... TK — ячейки памяти (триггер), т. е. ПЦУ обладают памятью.

## ТРИГГЕР

Триггер представляет собой простейшее последовательное цифровое устройство, предназначенное для записи и хранения одноразрядных двоичных чисел. У триггера есть информационные входы (S, R, J, K, T, D) и управляющие входы (V, C), а также два выхода ( $Q$  и  $\bar{Q}$ ).

Информационные входы позволяют устанавливать состояние триггера. Например, S и R используются для установки триггера в состояния "1" и "0", соответственно. J и K, используемые в универсальных триггерах, также позволяют устанавливать состояния "1" и "0". Вход D также используется для установки триггера в состояние "1" или "0".

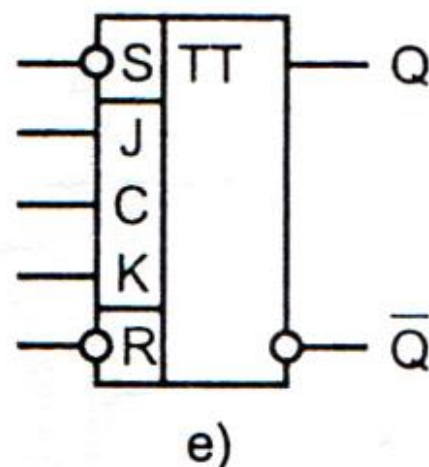
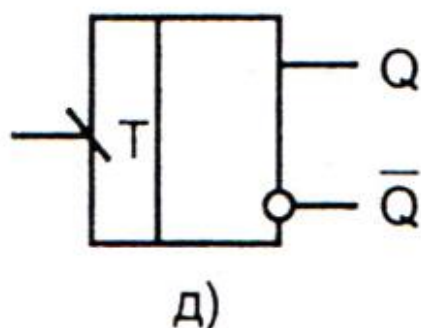
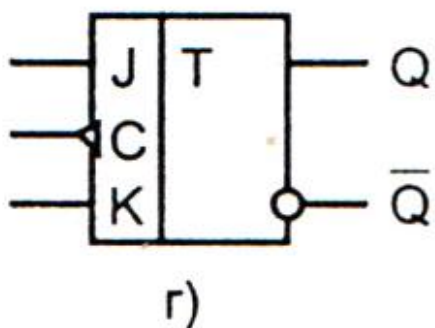
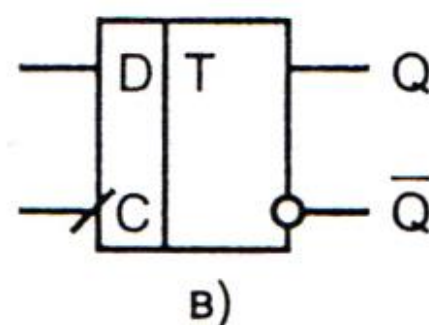
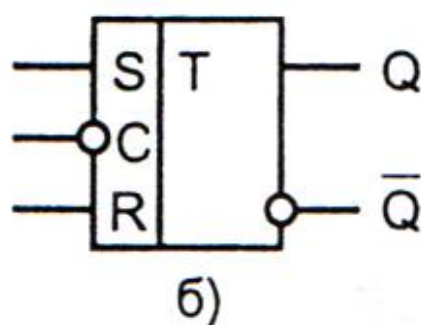
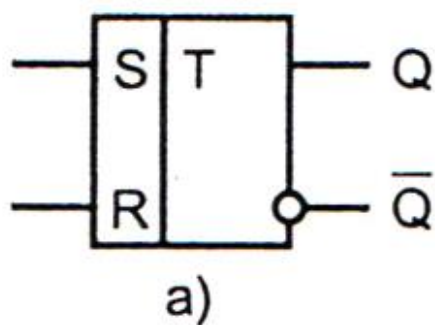
Управляющие входы V и С определяют, когда триггер будет реагировать на информацию. Управляющий вход С (иногда обозначаемый как Е) используется для разрешения приема информации. Вход Т (счетный) и управляющий вход С синхронизируют работу триггера.

Триггеры делятся на асинхронные и синхронные по способу приема информации. Асинхронные триггеры реагируют на информацию немедленно при ее появлении на входах, в то время как синхронные триггеры реагируют только при наличии сигнала на управляющем входе С (входе синхронизации).

По принципу построения синхронные триггеры могут быть одноступенчатыми или двухступенчатыми. Двухступенчатые триггеры имеют две ступени запоминания информации, а одноступенчатые - только одну.

Триггеры также классифицируются по функциональным возможностям: RS-триггеры с отдельной установкой состояния "0" и "1", универсальные JK-триггеры, D-триггеры с приемом информации по одному входу D и Т-триггеры со счетным входом Т.

Вне зависимости от вида, состояние триггера определяется значениями выходов  $Q$  и  $\bar{Q}$ , где  $Q = 1$  и  $\bar{Q} = 0$  соответствуют единичному состоянию, а  $Q = 0$  и  $\bar{Q} = 1$  - нулевому состоянию.



Выше приведены графические обозначения (УГО) триггеров, принятые в системе ЕСКД:

А) асинхронный RS — синхронный триггер;

Б) синхронизируемый RS — триггер. Синхронизация производится логическим 0;

В) D — триггер, срабатывающий на передний фронт 0,1;

Г) JK — триггер, срабатывающий по фронту 1, 0;

Д) T — триггер, срабатывающий по фронту 1, 0;

Е) двухступенчатый JK — триггер со входами разделённой установки в нулевое (R) и единичное (S) состояние.

Функционирование триггеров описывается таблицами переходов (истинности):

RS-триггеры

| Входы |   | Выходы |                |
|-------|---|--------|----------------|
| R     | S | Q      | $\overline{Q}$ |
| 0     | 0 | исх.   | исх.           |
| 0     | 1 | 1      | 0              |
| 1     | 0 | 0      | 1              |
| 1     | 1 | неопр. | неопр.         |

T-триггеры

| Входы | Выходы               |
|-------|----------------------|
| T     | Q                    |
| 0     | исх.                 |
| 1     | $\overline{Q}_{n-1}$ |

jk-триггеры

| Входы |   | Выходы               |
|-------|---|----------------------|
| j     | k | Q                    |
| 0     | 0 | исх.                 |
| 0     | 1 | 0                    |
| 1     | 0 | 1                    |
| 1     | 1 | $\overline{Q}_{n-1}$ |

D-триггеры

| Входы |   | Выходы |
|-------|---|--------|
| D     | C | Q      |
| 1     | 1 | 1      |
| 0     | 1 | 0      |

Таблица истинности триггеров

- «ИСХ» — означает режим хранения (исходное состояние или без изменения).
- «НЕОПР» — означает, что состояние выходов неопределённое, т. е. комбинация сигналов  $R = S = 1$  является запрещенной.
- В T-триггере при  $T = 1$  триггер переходит в инверсное состояние  $\overline{Q}_{n-1}$  (счетный режим).



- В JK-триггере при  $j = k = 1$  триггер переходит в инверсное состояние  $\overline{Q}n - 1$  (счетный режим).
- $\overline{Q}n - 1$  — это состояние выхода, предшествующее появлению информационных сигналов.

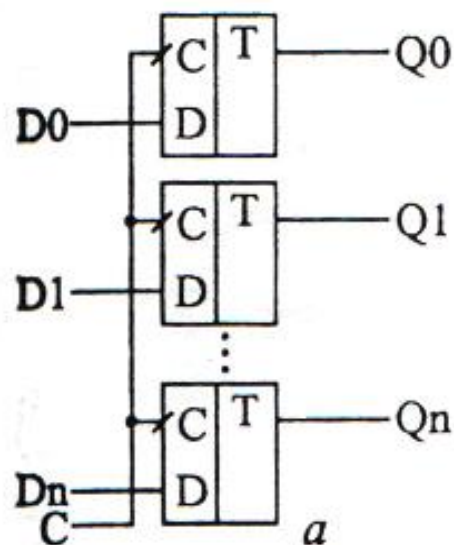
## РЕГИСТР

Регистр — это последовательное логическое устройство, используемое для хранения  $n$ -разрядных двоичных чисел и выполнения различных операций над ними. Регистр представляет собой упорядоченную последовательность триггеров, число которых соответствует числу разрядов в слове (обычно от 4 до 16). На схемах регистры обозначаются буквами RG. Регистры обеспечивают выполнение следующих типичных операций:

- Прием слова в регистр;
- Передача слова из регистра;
- Поразрядные логические операции;
- Сдвиг слова влево или вправо на заданное число разрядов;
- Преобразование последовательного кода слова в параллельный и обратно;
- Установка регистра в начальное состояние (сброс).

Регистры классифицируются по разным видам:

- Накопительные регистры, также называемые регистрами памяти или хранения.
- Сдвигающие регистры, которые могут быть параллельными, последовательными или комбинированными.

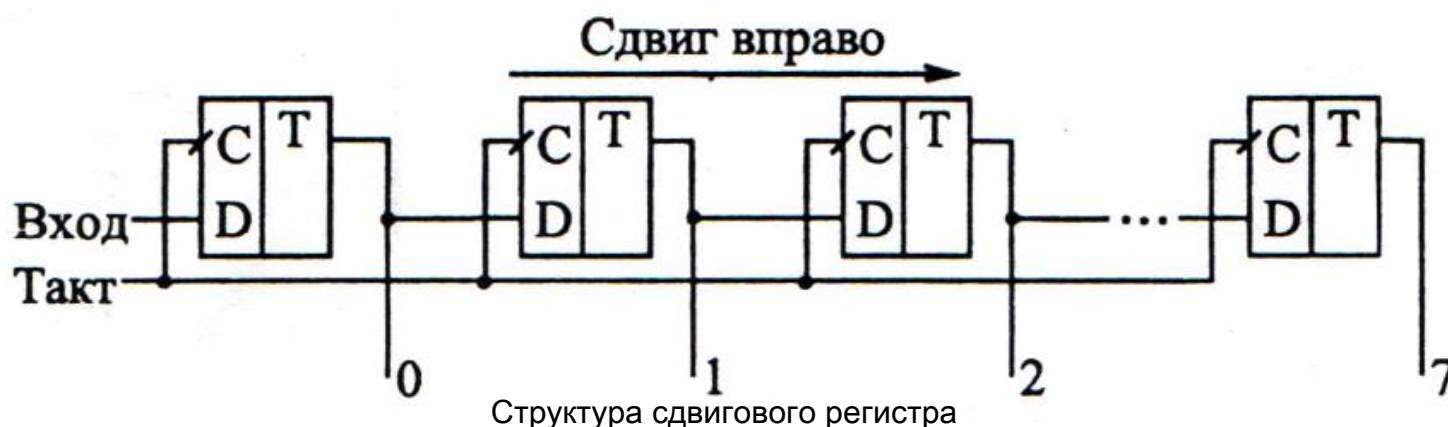


Структура параллельного регистра

- Сдвигающие регистры также могут быть однонаправленными или реверсивными в зависимости от направления передачи информации.

Параллельный регистр строится на основе D-триггеров. Каждый из этих триггеров имеет свой независимый информационный вход (D) и соответствующий независимый информационный выход. Тактовые входы всех D-триггеров соединены между собой. В результате параллельный регистр представляет собой многоразрядное устройство с множеством информационных входов.

D-триггеры реплицируют значения сигналов, находящихся на их информационных входах ( $D_0 - D_n$ ), при действии управляющего сигнала С. Это означает, что входная информация может быть параллельно записана в регистр с помощью управляющего сигнала С. Таким образом, параллельный регистр позволяет одновременно записывать информацию во все его разряды, что делает его очень эффективным для операций, где требуется быстрое параллельное чтение и запись данных.



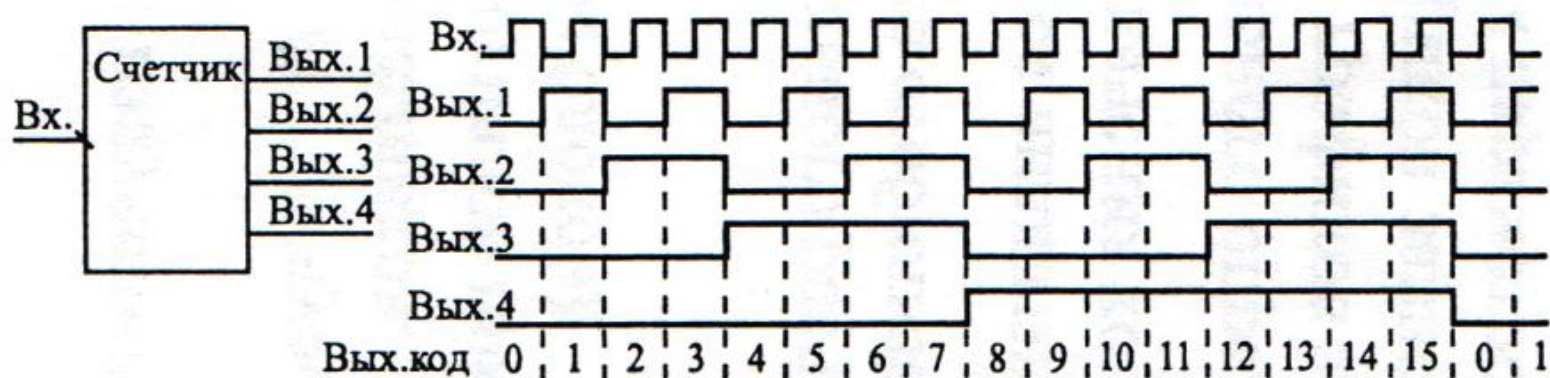
Сдвиговые регистры представляют собой последовательную цепочку D-триггеров, где выход каждого предыдущего триггера соединен с входом D следующего триггера. Тактовые входы всех триггеров объединены между собой. В результате такой регистр может рассматриваться как линия задержки, где входной сигнал последовательно перезаписывается из триггера в триггер по фронту тактового сигнала С.

В сдвиговом регистре цифровой код представляется как последовательность логических уровней, соответствующих значениям разрядов. Этот код называется последовательным, и каждый бит этого кода передается по одной и той же линии.

## СЧЕТЧИК ИМПУЛЬСОВ

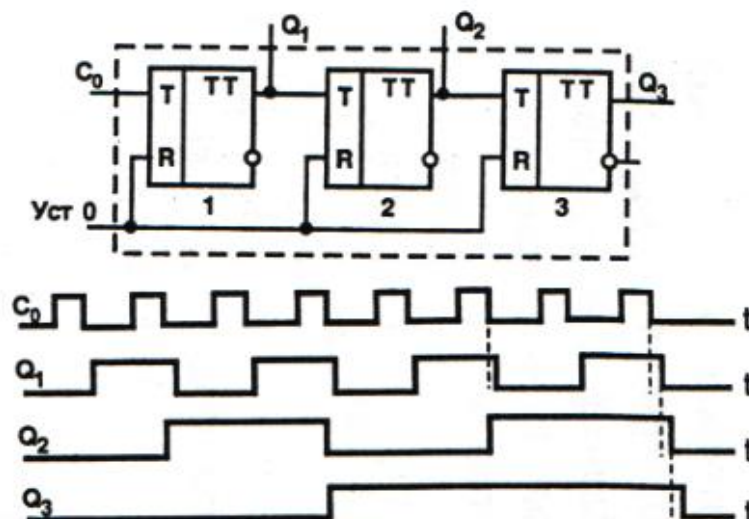
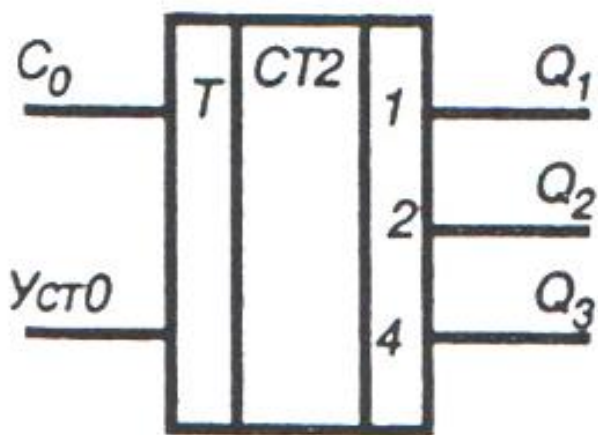
Счетчики представляют собой более сложные цифровые устройства, чем регистры. Несмотря на то, что они также состоят из триггеров, счетчики имеют более сложные связи между триггерами, что позволяет им выполнять более сложные функции. Внутренняя память счетчиков является оперативной, что означает, что она сохраняет данные только при наличии питания.

Счетчики импульсов представляют собой последовательные цифровые устройства, которые могут хранить информацию и выполнять операции счета. Основным параметром счетчика - модуль счета, который определяет максимальное количество импульсов, которое счетчик может подсчитать. Счетчики могут быть классифицированы по различным параметрам, таким как модуль счета, направление счета (суммирующие, вычитающие, реверсивные) и другие характеристики.



Работа четырехразрядного двоичного счетчика

Счетчики обычно реализуются с использованием Т-триггеров, но также могут быть построены с использованием Д-триггеров и JK-триггеров. Выходы счетчика, обозначаемые как выходы 1 до 4, представляют собой выходы четырех триггеров и представляют разряды двоичного кода. В случае четырехразрядного счетчика в режиме прямого счета он будет считать от 0 (0000) до 15 (1111). После достижения кода 1111, при следующем входном импульсе счетчик снова переключится в 0 (0000) и продолжит счет. Это означает, что счетчик работает в цикле, считая от 0 до максимального значения и затем снова с нуля.



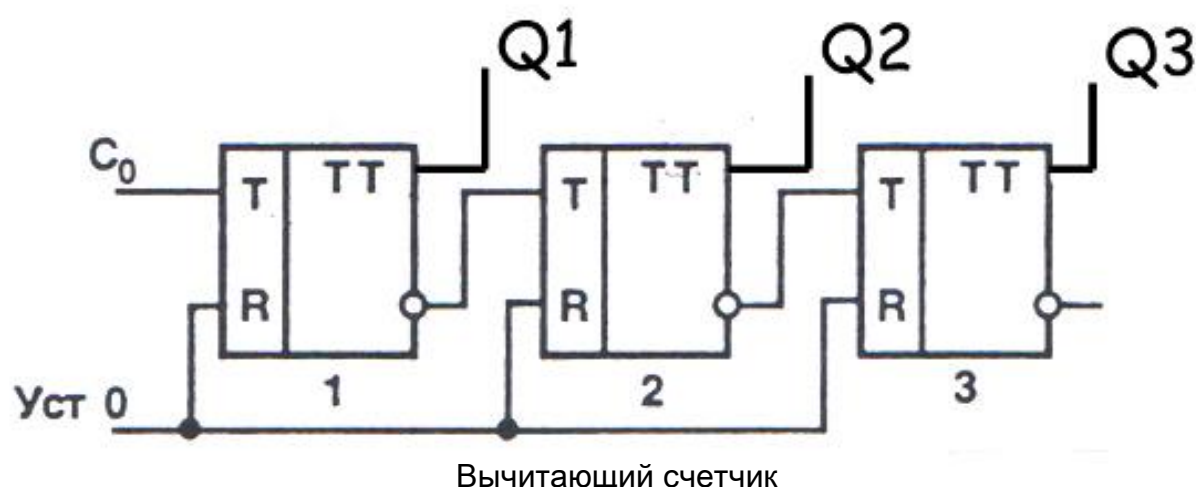
Графическое обозначение, структурная схема и временные диаграммы трехразрядного суммирующего счётчика

В данной схеме исходное состояние счетчика устанавливается путем подачи сигнала на вход Уст. Триггеры Т изменяют свое состояние по спаду входного сигнала, то есть при переходе сигнала от уровня 1 к уровню 0. Входной сигнал, подаваемый на вход C0, используется как счетный сигнал для первого триггера.

Работу счетчика можно описать с помощью временной диаграммы, которая показывает состояние каждого триггера в разные моменты времени. Общее количество возможных состояний счетчика ( $K_{\text{сост}}$ ) определяется числом триггеров  $n$ , где  $K_{\text{сост}} = 2^n$ . В данном случае  $n = 3$ , поэтому  $K_{\text{сост}} = 8$ .

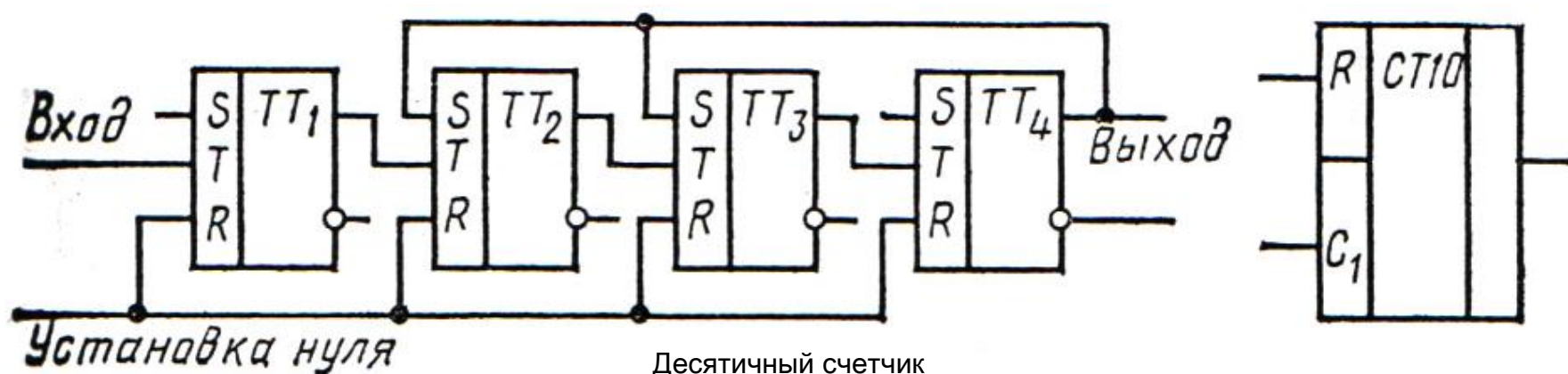
Обозначение "СТ 2" указывает на то, что это двоичный счетчик, а выходы 1, 2, и 4 ( $Q_1$ ,  $Q_2$ ,  $Q_3$ ) обозначают двоичные разряды, где  $2^0 = 1$ ,  $2^1 = 2$ , и  $2^2 = 4$ . Это представляет двоичное значение, которое счетчик считает в разных состояниях.

Для создания вычитающего счетчика, каждый последующий триггер должен быть связан с инверсным выходом предыдущего триггера. В этом случае, вычитающий счетчик будет выполнять микрооперацию уменьшения числа в счетчике на "1" при приходе каждого следующего счетного сигнала. То есть, он будет считать в обратном направлении, начиная с кода 111 и заканчивая кодом 000.



Реверсивный счетчик представляет собой счетчик, в котором счетные сигналы проходят через логические элементы, которые управляют направлением счета, то есть определяют, выполняется ли суммирование или вычитание. Это позволяет счетчику менять направление счета, в зависимости от внешних сигналов или условий.

Для создания счетчика, работающего в другой системе счисления, такой как десятичная, используют обратные связи. Например, если мы хотим создать десятичный счетчик импульсов на базе 4 триггеров, можем использовать обратные связи. Для этого сигналы обратной связи от выхода последнего триггера ( $T_4$ ) подаются на входы более младших триггеров ( $T_2$  и  $T_3$ ).





Когда поступает восьмой импульс, на выходе T4 появляется сигнал лог. 1, который переключает T3 и T2 в состояние лог. 1. Девятый импульс переводит T1 в состояние лог. 1, и все триггеры находятся в состоянии лог. 1. Десятый импульс возвращает все триггеры в состояние лог. 0, и счет начинается снова.

| Номер входного импульса | Состояние триггеров |                |                |                |
|-------------------------|---------------------|----------------|----------------|----------------|
|                         | T <sub>4</sub>      | T <sub>3</sub> | T <sub>2</sub> | T <sub>1</sub> |
| 0                       | 0                   | 0              | 0              | 0              |
| 1                       | 0                   | 0              | 0              | 1              |
| 2                       | 0                   | 0              | 1              | 0              |
| 3                       | 0                   | 0              | 1              | 1              |
| 4                       | 0                   | 1              | 0              | 0              |
| 5                       | 0                   | 1              | 0              | 1              |
| 6                       | 0                   | 1              | 1              | 0              |
| 7                       | 0                   | 1              | 1              | 1              |
| 8                       | 1                   | 0 (1)          | 0 (1)          | 0              |
| 9                       | ↓                   | ↑              | ↑              | ↑              |
| 10                      | 1                   | 1              | 1              | 1              |
|                         | 0                   | 0              | 0              | 0              |

Таблица переходов

Используя этот метод обратных связей, можно создать счетчики, работающие в системе счисления с любым основанием, с произвольным постоянным модулем счета, а также с переменным модулем счета и кольцевые счетчики.

### ЦИФРОВЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА (ЗУ)

Устройства для цифровой памяти предназначены для записи, хранения и считывания цифровых данных. Эти устройства более сложны, чем устройства, о которых ранее говорилось. Каждый набор данных хранится в отдельной ячейке, называемой ячейкой памяти.

Основные параметры устройств цифровой памяти включают в себя информационную емкость, энергопотребление, время удержания данных и скорость доступа. Информационная емкость определяется количеством кодов (m), которые могут храниться, и разрядностью этих кодов (n), и вычисляется по формуле:  $N = n * m$ .

Для измерения количества ячеек памяти используются следующие специальные единицы:

- 1К (кило) - 1024 (или  $2^{10}$ ), приближенно к 1000;
- 1М (мега) - 1048576 (или  $2^{20}$ ), приближенно к миллиону;
- 1Г (гига) - 1073741824 (или  $2^{30}$ ), приближенно к миллиарду.

Организация памяти записывается в формате "Количество ячеек x Разрядность кода". Например, «64К x 8» означает, что память имеет 64 тысячи (или 65536) ячеек, и каждая из них имеет восемь бит. Таким образом, информационная емкость составляет 524288 бит.

Энергопотребление — это мощность, которую устройство потребляет в режиме постоянной работы.

Время удержания данных — это интервал времени, в течение которого устройство сохраняет информацию в установленном режиме.

Скорость доступа — это время, необходимое для записи или считывания данных.

Устройства памяти можно разделить на три основных типа, в зависимости от того, как в них записывается и хранится информация:

- Постоянная память (ПЗУ) или память только для чтения (ROM), в которую информация записывается один раз на стадии производства и не теряется при отключении питания.

- Программируемая постоянная память (ППЗУ) или программируемая ROM (PROM), в которую информацию можно записать с помощью специальных методов, ограниченное количество раз, и она также не теряется при отключении питания.

- Оперативная память (ОЗУ) или память с произвольным доступом (RAM), в которую информацию можно записывать и считывать многократно, но она теряется при отключении питания.

## **ОПЕРАТИВНЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА**

Оперативная память (RAM - Random Access Memory) может быть разделена на два основных типа: статическую (SRAM) и динамическую (DRAM).

Статическая оперативная память (SRAM) способна сохранять данные при наличии питания в течение неопределенного времени без необходимости постоянно их обновлять. Она использует триггеры в качестве запоминающих ячеек и предоставляет быстрый доступ к данным. В SRAM, данные хранятся как биты, и они могут быть доступны мгновенно. Микросхемы SRAM могут быть подразделены на микромощные варианты с относительно медленным доступом (55-120 нс) и высокоскоростные варианты с более быстрым доступом (7-25 нс), но они потребляют больше энергии.

Динамическая оперативная память (DRAM) использует конденсаторы малой емкости, расположенные на пересечениях строк и столбцов матрицы для хранения данных. Это обеспечивает высокую плотность упаковки и большую удельную информационную ёмкость, но требует периодической регенерации. Это означает, что данные должны периодически считываться и записываться в те же самые ячейки памяти для их сохранения. DRAM чаще используется в компьютерах и другой вычислительной технике.

Существуют и другие разновидности оперативной памяти, такие как "Zero-Power" RAM с встроенной литиевой батареей, которая сохраняет данные даже без постоянного питания, и "Dual-Port" RAM со своей уникальной системой доступа к данным.

## **ПОСТОЯННЫЕ ЗАПОМИНАЮЩИЕ УСТРОЙСТВА**

ПЗУ классифицируют по способу внесения информации и возможности её стирания, если таковая возможность предусмотрена в данном классе ПЗУ. Самым экономичным методом записи является масочное программирование на этапе изготовления кристалла.

Микросхемы ПЗУ с предварительно записанной информацией, так называемые ROM (Read Only Memory), невозможно перепрограммировать и применяются исключительно при массовом производстве, при больших сериях и при жизненной важности точности записываемых данных.

Другим видом ПЗУ являются микросхемы PROM (Programmable Read Only Memory), поставляемые в чистом виде, предоставляя пользователю возможность самостоятельно вносить необходимое содержимое при помощи программатора. Если данный процесс необратим, то такие микросхемы получают название OTP (One Time Programmable) - однократно программируемые.

В случае, если существует возможность стирания содержимого с последующей перезаписью, эти микросхемы называются EPROM (Erasable Programmable Read Only Memory). При этом, в зависимости от метода стирания, они бывают UV-EPROM, где стирание осуществляется с использованием ультрафиолетового излучения, или EEPROM, где стирание выполняется электрическим путем. Впрочем, с течением времени терминология смешивается, и более поздние термины чаще используют аббревиатуру EEPROM, обозначая конкретный вид памяти, который можно рассматривать как энергонезависимое ОЗУ.

ППЗУ с возможностью электрического стирания обычно называют Flash-памятью. Между ним и EEPROM есть значительные различия. EEPROM позволяет достигать до произвольных ячеек памяти при записи, в то время как Flash-память оперирует страницами при стирании и записи. С Flash-памятью невозможно перезаписать только одну конкретную ячейку памяти. Однако при чтении данных между ними нет принципиальной разницы.

Основное применение Flash-памяти связано с хранением программных текстов, таблиц и других данных, которые либо не подлежат изменениям, либо могут меняться, но не очень часто. Программирование Flash-памяти требует выполнения определенных шагов, включая перевод микросхемы в режим программирования и контроль завершения этого процесса.

С другой стороны, EEPROM-память используется для временного хранения данных в процессе работы устройства. Она применяется для хранения переменных, настроек, и других данных, которые нужно сохранить при выключении устройства. К примеру, на EEPROM автоматически сохраняются текущие данные при отключении питания телевизора.

При оценке стоимости хранения одного байта информации Flash-память обычно имеет большую емкость и более низкую цену, что делает ее предпочтительной в ситуациях, где требуется много прочности для долгосрочного хранения данных. С другой стороны, EEPROM является более удобным вариантом для временного хранения данных во время работы устройства.