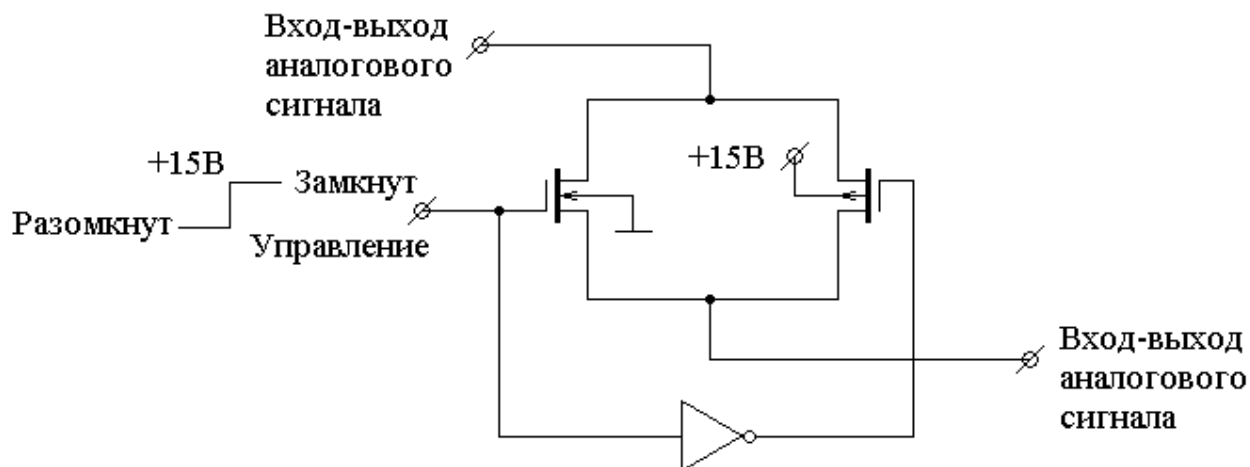


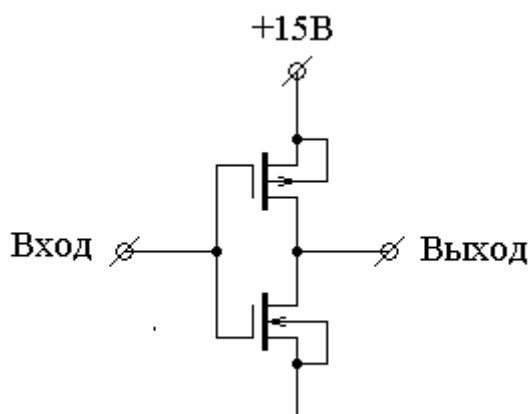
### Аналоговые ключи на полевом транзисторе.

Пример аналогового ключа на обогащенных МОП-транзисторах (**MOSFET**), который способен пропускать без искажений положительные аналоговые сигналы от 0 В до +15 В.



### Логический КМОП-инвертор.

Логический инвертор на обогащенных КМОП-транзисторах (комплементарные металл-окисел-полупроводник транзисторы).



### Логические микросхемы.

Обычно логические микросхемы имеют однополярное питание +5 Вольт и имеют ногу, соединенную с общим проводом схемы.

На входах и выходах логических схем различают только два напряжения: около нуля (логический ноль) и около +5 Вольт (логическая единица).

Работа простейших логических схем характеризуется таблицей истинности.

Рассмотрим для примера логическую схему 2И. Схема имеет два входа, поэтому 2И, и один выход. Таблица истинности схемы 2И имеет вид:

0	0		0
0	1		0
1	0		0
1	1		1

Здесь первый столбец — это возможные варианты логических уровней на первом входе схемы, второй столбец — уровни на втором входе схемы, третий столбец — уровни на выходе.

Таблица истинности для схемы 2И показывает, что напряжение логической единицы на выходе схемы присутствует в единственном случае, когда единица одновременно присутствует и на первом и на втором входе, поэтому — 2И.

Рассмотрим схему 2И-НЕ. Логические уровни на ее выходе отличаются от уровней схемы 2И тем, что они инвертированы:

0 0 | 1  
 0 1 | 1  
 1 0 | 1  
 1 1 | 0

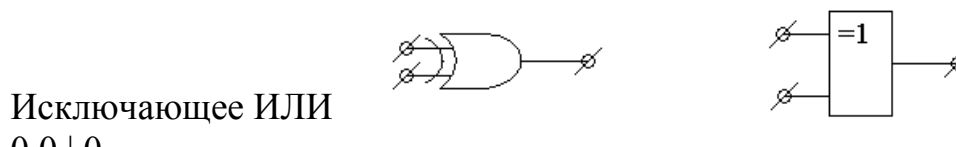
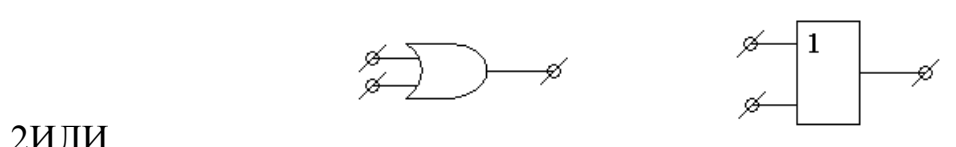
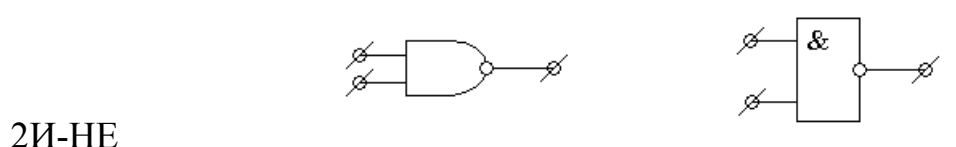
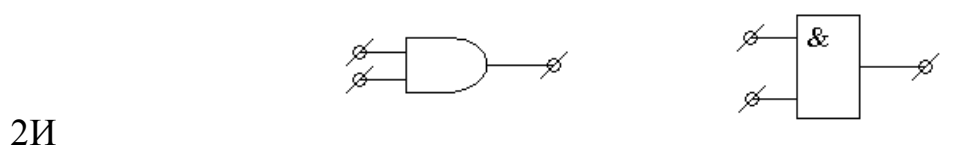
Для сравнения приведем таблицу истинности для схемы 3ИЛИ:

0 0 0 | 0  
 0 0 1 | 1  
 0 1 0 | 1  
 0 1 1 | 1  
 1 0 0 | 1  
 1 0 1 | 1  
 1 1 0 | 1  
 1 1 1 | 1

Заметим, что в одном корпусе микросхемы обычно присутствуют несколько независимых логических схем с общим питанием и одним общим проводом. Например, 4-2И-НЕ — четыре схемы 2И-НЕ в одном корпусе.

Условные обозначения логических микросхем в западной литературе и в советской литературе различаются.

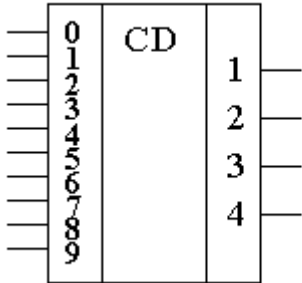
Микросхема.      Западная литература.      Советская литература.



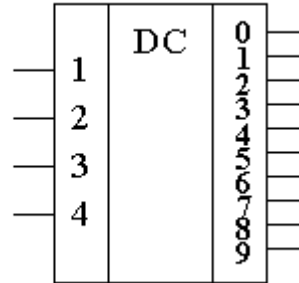
0 0 | 0  
 0 1 | 1

1 0 | 1  
 1 1 | 0

Шифратор формирует номер канала, по которому пришла логическая единица. Дешифратор по номеру канала направляет логическую единицу.

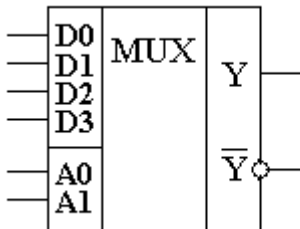


Шифратор (кодер).

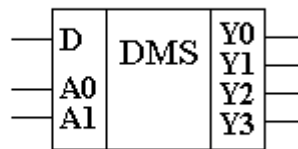


Дешифратор.

Мультиплексор подключает один из входов к выходу.



Мультиплексор.



Демультимплексор.

### Последовательная логика. RS-триггер.

Рассмотренные до этого момента логические схемы — это схемы комбинационной логики. В схемах комбинационной логики состояние выходов однозначно определяется состоянием входов в текущий момент времени. В схемах последовательной логики состояние выходов сейчас зависит от состояния входов не только в текущий момент времени, но и от состояния входов в предшествующие моменты времени.

Простейшим примером последовательной логики является RS-триггер. RS-триггер — это элементарная ячейка памяти, в которую можно записать одно из двух состояний 0 или 1.

Рассмотрим работу логической схемы RS-триггера.

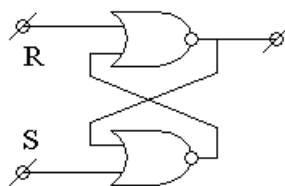


Схема состоит из двух логических схем 2ИЛИ-НЕ. Схема содержит два входа и один выход.

Вход  $R$  (reset) — сброс. Вход  $S$  (set) — установка.

В нормальном состоянии оба входа  $R$  и  $S$  находятся в состоянии логического нуля. При этом на выходе схемы может быть любое из двух состояний 0 или 1 в зависимости от предыстории состояний входов.

И действительно.

Предположим, что выход схемы находится в состоянии 1. Это напряжение поступает на верхний вход нижней схемы 2ИЛИ-НЕ. Тогда на входах нижней схемы 2ИЛИ-НЕ будут логические уровни 1 и 0. Логическая функция 2ИЛИ нижней схемы будет равна 1. Тогда выход нижней схемы 2ИЛИ-НЕ будет в состоянии 0. Это напряжение поступает на нижний вход верхней схемы 2ИЛИ-НЕ. На входах этой схемы будут уровни 0 и 0. Логическая функция 2ИЛИ верхней схемы будет равна 0. Тогда выход верхней схемы 2ИЛИ-НЕ будет в состоянии 1, что соответствует начальному предположению о состоянии 1 выхода всей схемы.

Предположим теперь, что выход всей схемы находится в состоянии 0. Это напряжение поступает на верхний вход нижней схемы 2ИЛИ-НЕ. Тогда на входах нижней схемы 2ИЛИ-НЕ будут логические уровни 0 и 0. Логическая функция 2ИЛИ нижней схемы будет равна 0. Тогда выход нижней схемы 2ИЛИ-НЕ будет в состоянии 1. Это напряжение поступает на нижний вход верхней схемы 2ИЛИ-НЕ. На входах этой схемы будут уровни 1 и 0. Логическая функция 2ИЛИ верхней схемы будет равна 1. Тогда выход верхней схемы 2ИЛИ-НЕ будет в состоянии 0, что соответствует начальному предположению о состоянии 0 выхода всей схемы.

Получается, что при нулевых значениях на входах  $R$  и  $S$  любое из двух возможных состояний 0 или 1 выхода схемы является устойчивым состоянием.

Предположим теперь, что на вход  $R$  поступает короткий импульс логической единицы 1. Логическая функция 2ИЛИ верхней схемы будет во время этого импульса равна 1 независимо от значения напряжения на втором входе верхней схемы 2ИЛИ-НЕ. Тогда на выходе верхней схемы 2ИЛИ-НЕ будет логический 0. Логический 0 на выходе всей схемы останется в устойчивом состоянии и после окончания импульса логической 1 на входе  $R$ . Следовательно, импульс на входе  $R$  сбрасывает выход всей схемы в состояние 0, то есть выполняет функцию reset.

Предположим теперь, что на вход  $S$  поступает короткий импульс логической единицы 1. Логическая функция 2ИЛИ нижней схемы будет во время этого импульса равна 1 независимо от значения напряжения на втором входе нижней схемы 2ИЛИ-НЕ. Тогда на выходе нижней схемы 2ИЛИ-НЕ будет логический 0. Это напряжение поступает на один из входов верхней логической схемы 2ИЛИ-НЕ. На обоих входах верхней логической схемы 2ИЛИ-НЕ при этом логические 0. Логическая функция 2ИЛИ верхней схемы будет равна 0. Тогда на выходе верхней схемы 2ИЛИ-НЕ будет логическая 1. Логическая 1 на выходе всей схемы останется в устойчивом состоянии и после окончания импульса логической 1 на входе  $S$ . Следовательно, логическая 1 на входе  $S$  устанавливает выход всей схемы в состояние 1, то есть выполняет функцию set.

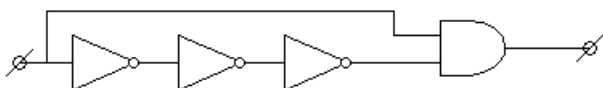
Рассматриваемый *RS*-триггер является элементарной ячейкой памяти, в которую можно записать одно из двух состояний 0 или 1. Записанное состояние остается в этой ячейке памяти и после того, как пропадает импульс записи.

Этот принцип сохранения состояния используется во всех более сложных микросхемах памяти.

### Логические иголки.

В сложных схемах комбинационной логики возможны непредвиденные короткие импульсы на выходах, связанные с разным временем распространения логических сигналов по разным путям. Эти короткие импульсы называют логическими иголками. Логические иголки могут приводить к неожиданным срабатываниям логических схем.

В следующей схеме, если не учитывать конечное время распространения логического сигнала через каждый из трех инверторов, сигнал на выходе схемы 2И всегда ноль.



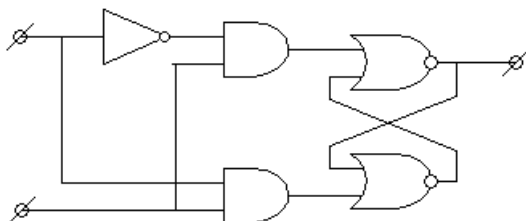
С учетом задержки распространения на выходе появляется короткий импульс при каждом перепаде вверх напряжения на входе схемы. Эту схему можно назвать одновибратором.

В сложных схемах короткие логические иголки могут представлять серьезную проблему. Для ее устранения была придумана синхронная последовательная логика.

Суть работы схем синхронной последовательной логики состоит в том, что они анализируют входные сигналы только на фронте так называемого тактового сигнала.

### D-триггер.

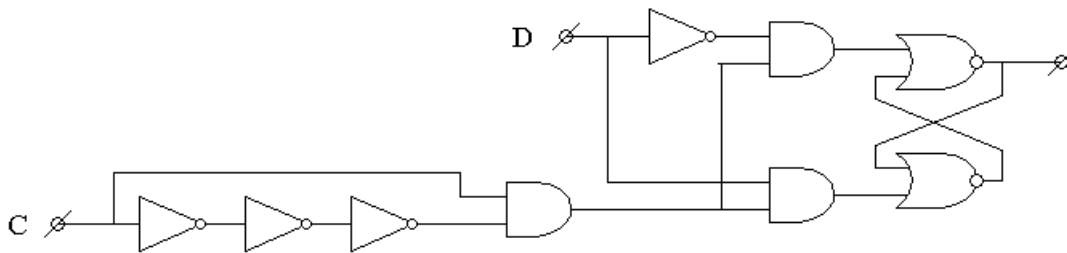
В схеме, приведенной ниже, справа *RS*-триггер, выполненный на двух схемах ИЛИ-НЕ. Напряжение с верхнего входа схемы проходит на выход схемы и может установить или сбросить *RS*-триггер, но только в том случае, когда на нижнем входе разрешающее высокое напряжение.



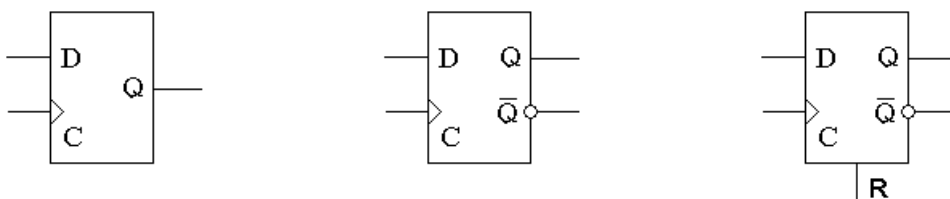
Если на нижний вход подать короткий импульс, то *RS*-триггер установится или сбросится в зависимости от состояния верхнего входа именно в момент короткого синхроимпульса.

Если на нижний вход подавать сигнал через одновибратор, то изменение состояния выхода схемы будет возможно только по переднему фронту тактового импульса на входе одновибратора.

Эта схема  $D$ -триггера.



Вся эта схема  $D$ -триггера обозначается, как

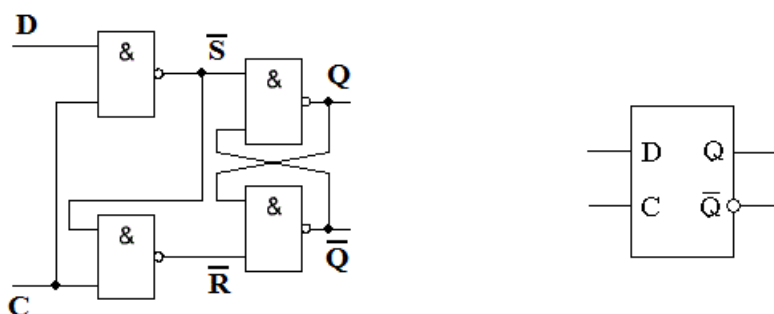


Здесь  $D$  — это вход данных,  $Q$  — выход, треугольником обозначен тактовый вход  $C$ .

### Еще раз о $D$ -триггере.

$D$ -триггеры бывают двух типов.

Первый тип —  $D$ -триггер защелка (latch), который срабатывает по уровню входа  $C$ . Этот тип  $D$ -триггера мы уже обсуждали на прошлой лекции, как некоторую промежуточную схему перед обсуждением  $D$ -триггера срабатывающего по фронту тактового импульса.

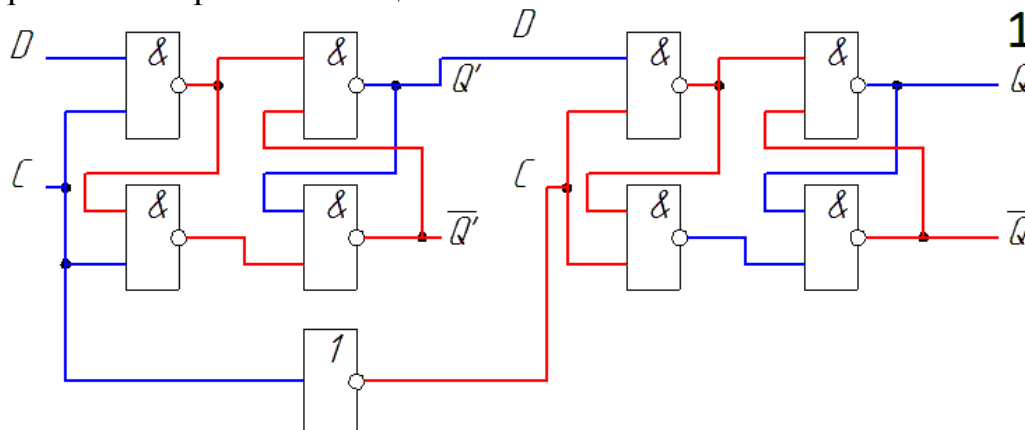


Здесь вторая пара схем 2И-НЕ — это RS-триггер, запись в который происходит низким логическим уровнем. Вход  $C$  является входом разрешения для двух левых схем 2И-НЕ. Если на входе разрешения высокий логический уровень, то выходы двух левых схем 2И-НЕ имеют противоположные логические значения. При этом они переводят правую пару 2И-НЕ либо в состояние Set, либо в состояние Reset.

Заметим, что состояние триггера защелкивается в момент перехода на низкий уровень входа  $C$ , а пока вход  $C$  остается на высоком логическом уровне сигнал с линии данных пропускается на выход  $Q$ . Если при высоком уровне на

входе С уровень входа D изменяется, то одновременно изменяется выход Q. Если изменения на входе D происходят несколько раз за время высокого уровня на входе С, то несколько раз изменяется напряжение на выходе Q. Это может быть неудобно.

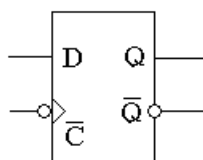
Это неудобство исключено во втором типе D-триггера, который срабатывает по фронту тактового импульса (flip-flop trigger), но в два этапа в отличие от того гипотетического варианта D-триггера, который мы рассматривали на прошлой лекции.



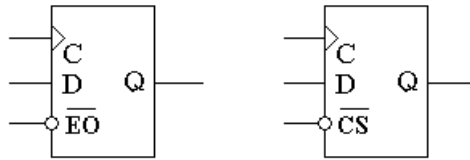
При переходе входа С на высокий уровень во вторую слева пару схем 2И-НЕ происходит передача состояния входа D, как в рассмотренную ранее D-триггер защелку. Тактовый вход С правой четверки схем 2И-НЕ при этом остается низким, запрещающим запись в правый D-триггер защелку. То есть при изменениях входа D левой защелки при высоком уровне входа С левой защелки напряжение на выходе правой защелки не изменяется и вообще остается в состоянии, которое было до прихода высокого уровня на вход С левой защелки.

При переходе входа С левой защелки на низкий уровень левая защелка запоминает состояние входа D левой защелки и оставляет его на выходе левой защелки. Одновременно появляется высокий уровень на входе С правой защелки. Это позволяет запомнить в правую защелку данные, которые в этот момент были сохранены в левой защелке.

В результате данный flip-flop D trigger срабатывает по перепаду вниз на тактовом входе левой защелки. Соответственно его обозначение будет следующим:



**Выходы с тремя состояниями.**



Выходы с тремя состояниями. Здесь  $\overline{EO}$  (enable output) — разрешение выхода низким уровнем энергии. Если на входе  $\overline{EO}$  высокий уровень энергии, то выход находится в состоянии с высоким выходным сопротивлением. Обозначение  $\overline{CS}$  (chip select) действует ровно так же, как и  $\overline{EO}$ . Состояние на выходе появляется на переднем фронте тактового входа C.

Пример D-триггера с тремя состояниями SN74LVC1G374-Q1 — Single D-Type Flip-Flop with 3-State Output:

**Table 1. FUNCTION TABLE**

INPUTS			OUTPUT Q
OE	CLK	D	
L	↑	L	L
L	↑	H	H
L	H or L	X	Q
H	X	X	Z

**LOGIC DIAGRAM (POSITIVE LOGIC)**

