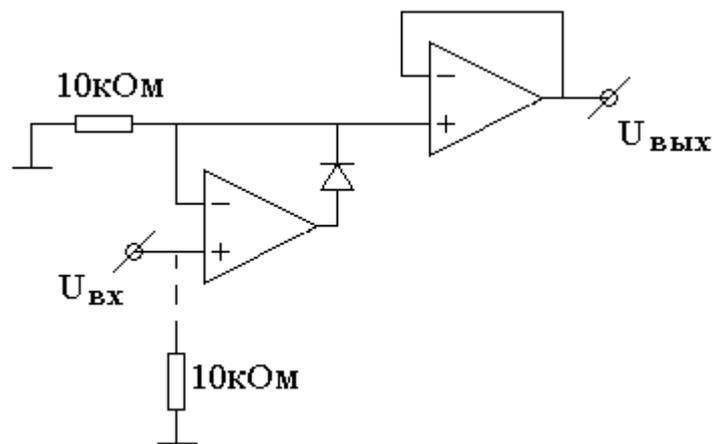
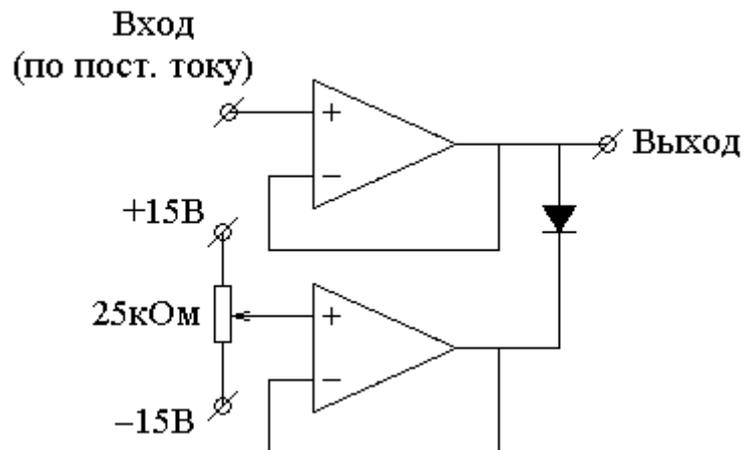
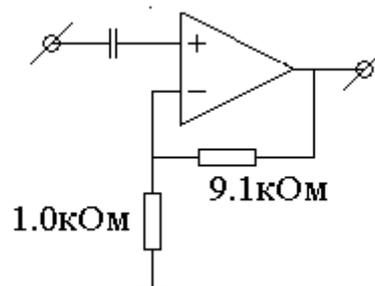


## Негодные схемы (проверка).

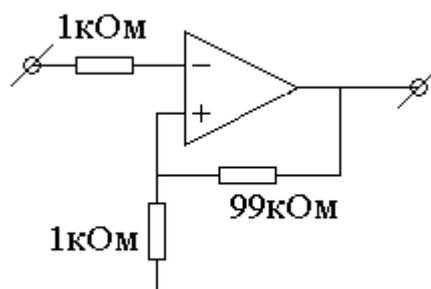
Регулируемый ограничитель:



Десятикратный усилитель переменного тока:



100-кратный усилитель постоянного напряжения:



### 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.

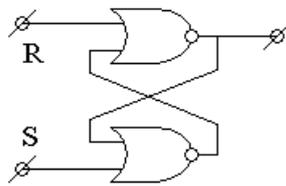


Схема состоит из двух логических схем 2ИЛИ-НЕ. Схема содержит два входа и один выход.

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.

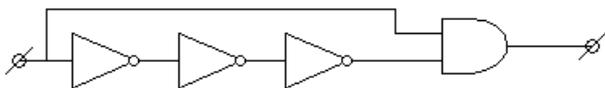
Рассматриваемый  $RS$ -триггер является элементарной ячейкой памяти, в которую можно записать одно из двух состояний 0 или 1. Записанное состояние остается в этой ячейке памяти и после того, как пропадает импульс записи, остается до выключения питания.

Этот принцип сохранения состояния используется и в более сложных микросхемах памяти.

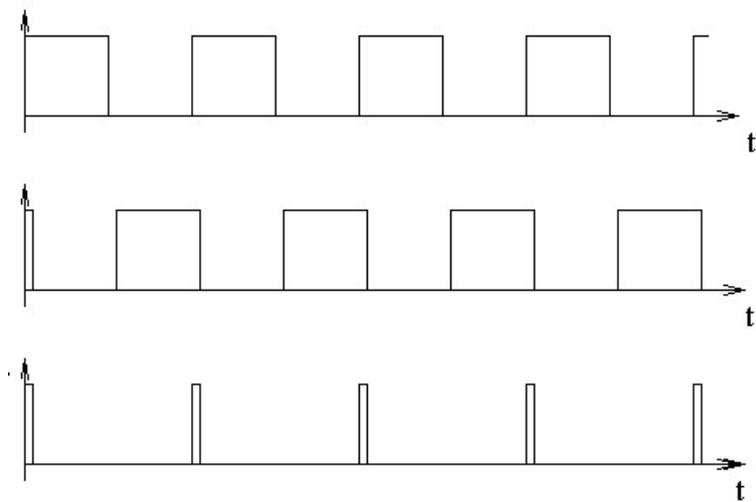
### **Логические иголки. Одновибратор.**

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.

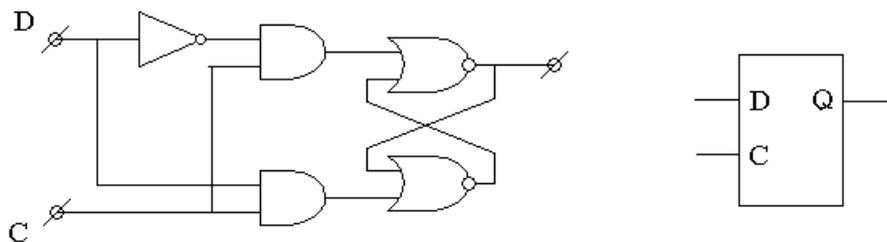


В сложных схемах короткие логические иголки могут представлять серьезную проблему. Для ее устранения была придумана синхронная последовательная логика.

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

### **D-триггер защелка (latch).**

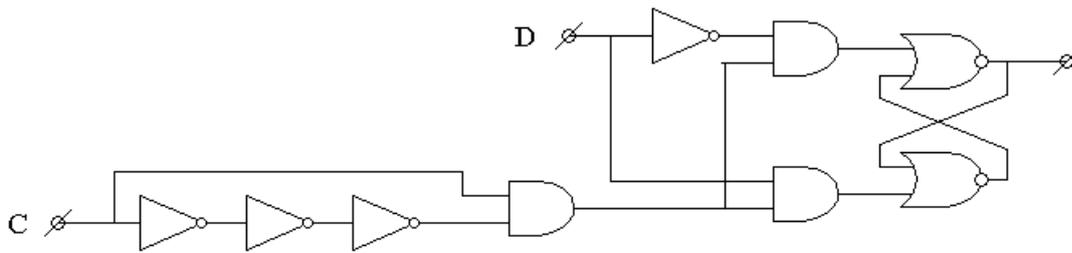
В схеме, приведенной ниже, справа *RS*-триггер, выполненный на двух схемах ИЛИ-НЕ. Напряжение с верхнего *D*-входа схемы (вход данных Data) проходит на верхний или нижний вход *RS*-триггера и может сбросить или установить *RS*-триггер, но только в том случае, когда на нижнем *C*-входе (тактовом входе Clock) разрешающее высокое напряжение.



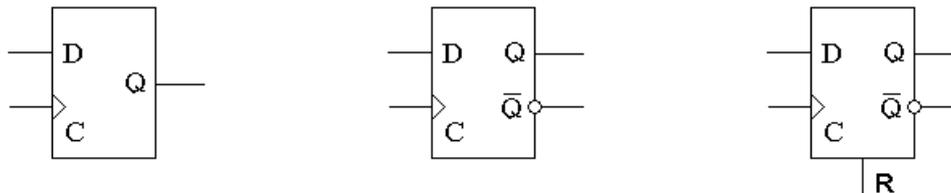
Если на нижний вход подать короткий импульс, то *RS*-триггер установится или сбросится в зависимости от состояния верхнего входа именно в момент короткого синхроимпульса.

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

Это возможный вариант схемы *D*-триггера.



Вся эта схема  $D$ -триггера обозначается, как



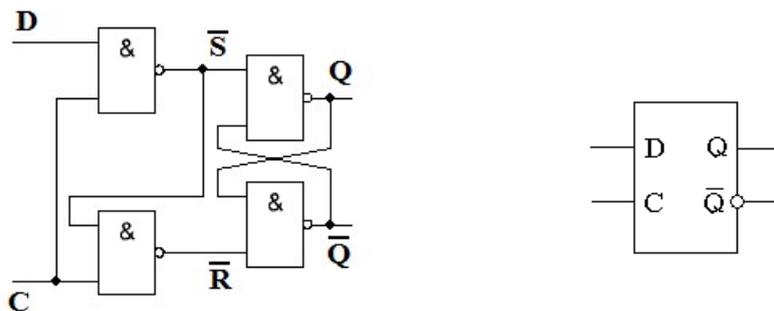
Здесь  $D$  — это вход данных,  $Q$  — выход, треугольником обозначен тактовый вход  $C$ , если срабатывание схемы ( $D$ -триггера) происходит по фронту, а не по уровню, тактового импульса.

### Flip-flop $D$ trigger. (шлёп-шлёп триггер)

$D$ -триггеры бывают двух типов.

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

На рисунке ниже приведен еще один вариант  $D$ -триггера защелки.

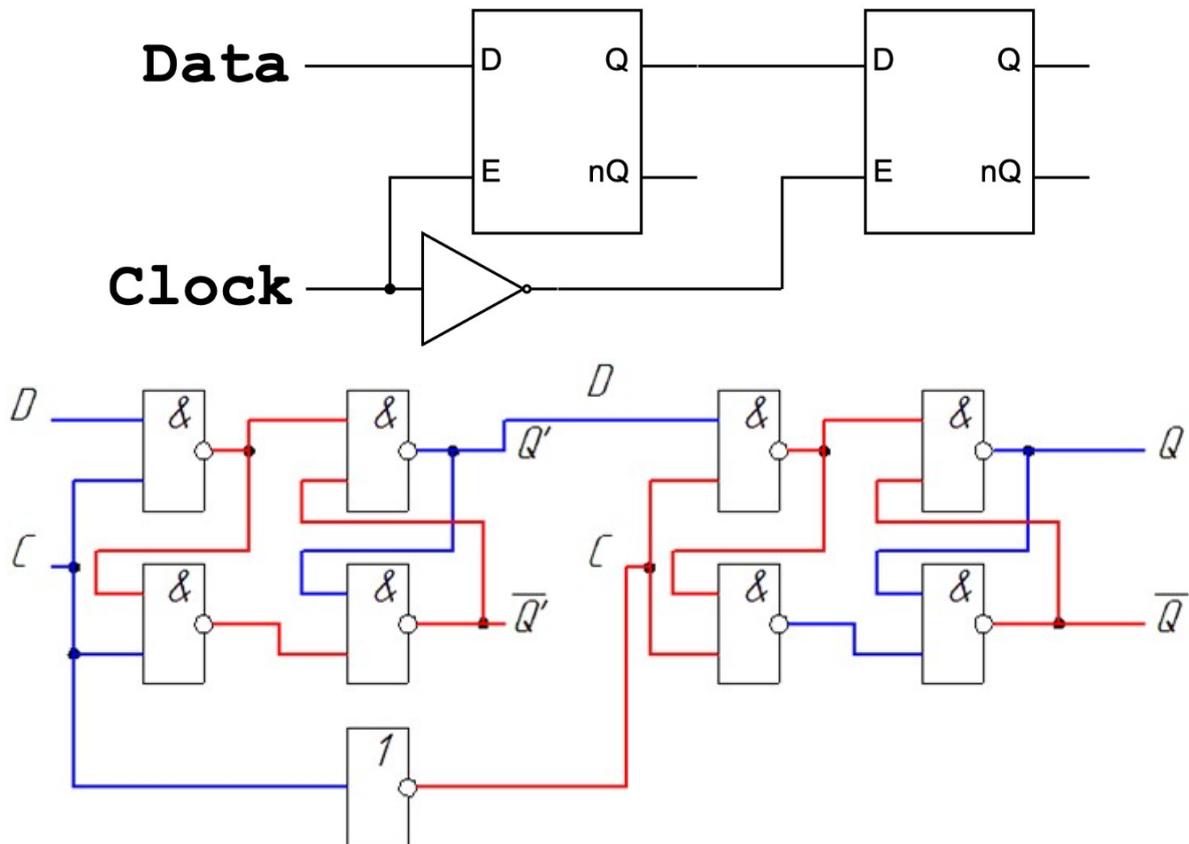
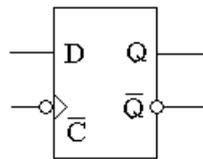


Здесь вторая пара схем 2И-НЕ — это RS-триггер, запись в который происходит низким логическим уровнем. Вход  $C$  является входом разрешения для двух левых схем 2И-НЕ. Если на входе разрешения высокий логический уровень, то выходы двух левых схем 2И-НЕ имеют противоположные логические значения. При этом 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  $C$  уровень входа  $D$  изменяется, то одновременно изменяется выход  $Q$ .

Если изменения на входе D происходят несколько раз за время высокого уровня на входе C, то несколько раз изменяется напряжение на выходе Q. Это может быть неудобно.

Это неудобство исключено во втором типе D-триггера, который срабатывает по фронту. Вместо рассмотренного нами ранее варианта D-триггера с одновибратором в микросхемах используется другой вариант D-триггера, который срабатывает по фронту тактового импульса, это так называемый flip-flop trigger.

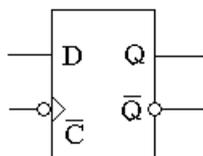


При переходе входа C на высокий уровень во вторую слева пару схем 2И-НЕ происходит передача состояния входа D, как в рассмотренную ранее D-триггер защелку. Тактовый вход C правой четверки схем 2И-НЕ при этом остается низким, запрещающим запись в правый D-триггер защелку. То есть при изменениях входа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

При переходе входа C левой защелки на низкий уровень левая защелка запоминает состояние входа D левой защелки и оставляет его на выходе левой

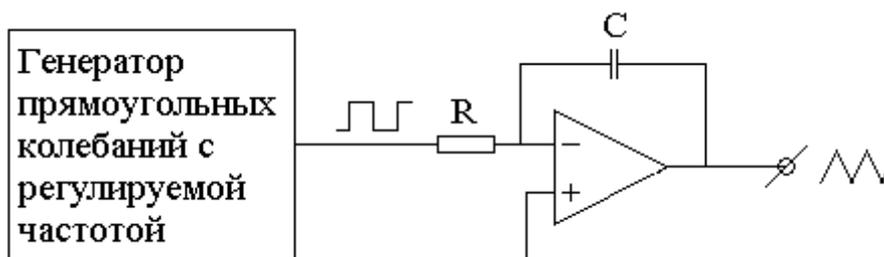
защелки. Одновременно появляется высокий уровень на входе С правой защелки. Это позволяет запомнить в правую защелку данные, которые в этот момент были сохранены в левой защелке.

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

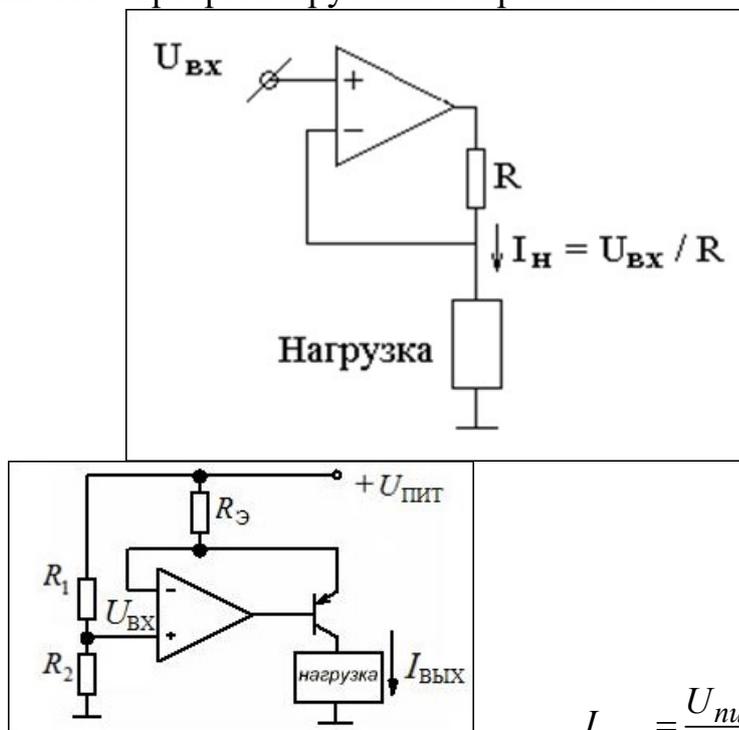


### Негодные схемы (на дом).

Генератор треугольных колебаний с регулируемой частотой:

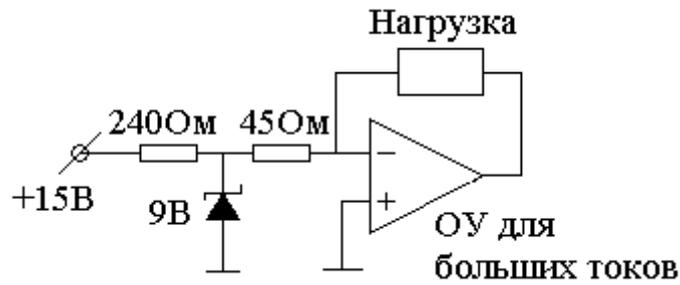


Источник тока программируемый напряжением:

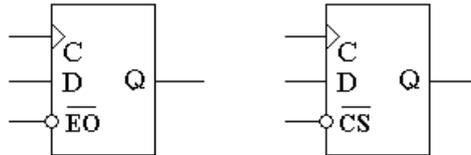


$$I_{вых} = \frac{U_{пит} - U_{вх}}{R_3}$$

Источник тока 200 мА:



### Выходы с тремя состояниями.



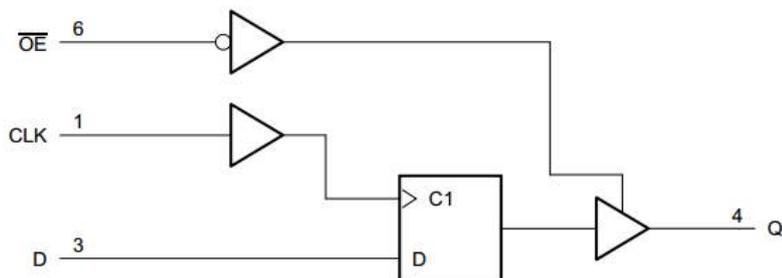
Выходы с тремя состояниями. Здесь  $\overline{EO}$  (enable output) — разрешение выхода низким уровнем напряжения. Если на входе  $\overline{EO}$  высокий уровень энергии, то выход находится в состоянии с высоким выходным сопротивлением. Обозначение  $\overline{CS}$  (chip select) действует ровно так 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)



### Регистр. Шина. Память.

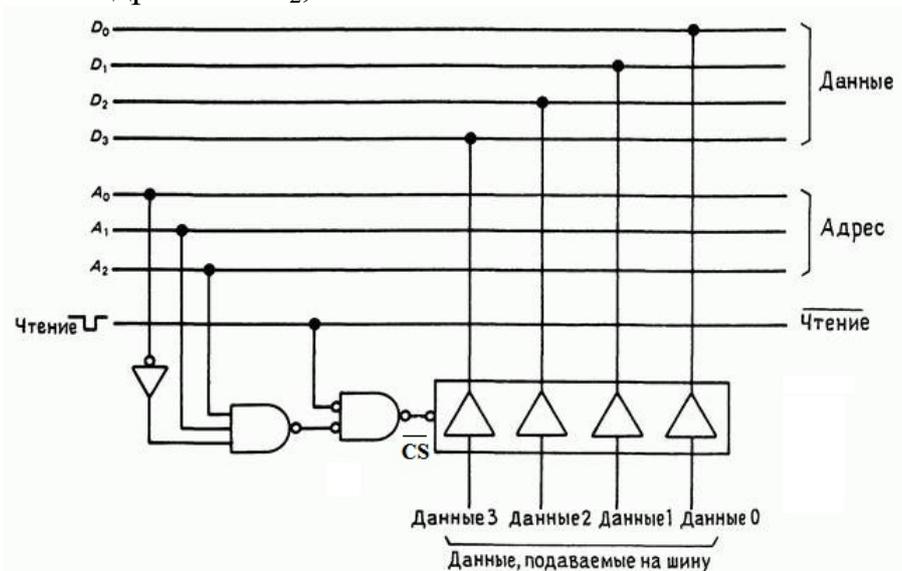
Регистр — несколько параллельных D-триггеров с выходами с тремя состояниями. Много регистров может быть в одной микросхеме — это микросхема памяти.

Допустим, что в каждом регистре 8 D-триггеров — 8 бит или 1 байт. Допустим, что в микросхеме 1 гигабайт памяти (1GB или 1ГБ). Если у каждого регистра микросхемы будут свои выходы (ножки) данных, то микросхема должна иметь 8 миллиардов (точнее  $2 \cdot 10^{33}$ ) ножек выходов регистров. Ни в каждую комнату поместится такая микросхема.

Вместо такого большого числа выходов можно оставить 8 выходов данных (или 4 или 16 или 32 или 64) — так называемая шина данных. Каждый из регистров микросхемы имеет выходы с тремя состояниями. В каждый момент времени к шине данных подключены выходы не более чем одного регистра. Иначе один регистр будет стремиться удержать на какой-то линии данных высокое напряжение логической единицы, а другой регистр в тот же момент времени на той же линии будет удерживать низкое напряжение логического нуля.

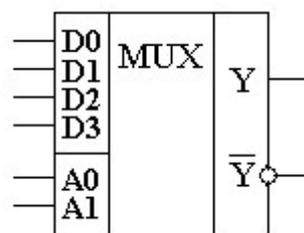
Чтобы задать микросхеме, какой именно регистр должен в данный момент подключить свои выходы к шине данных, нужна адресная шина.

Пример регистра, который передает данные на шину, если на адресной шине установлен адрес  $6 = 110_2$ ,



три логических схемы слева выполняют функцию, которую в микросхеме памяти выполняет встроенный в микросхему мультиплексор.

Микросхемы памяти — множество регистров с разными последовательными адресами и мультиплексор для каждого бита данных.



**JK-триггер.**

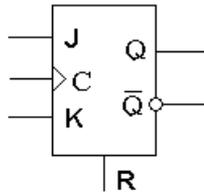
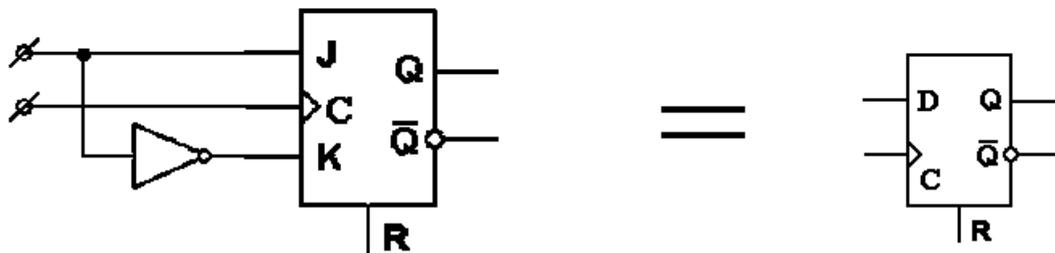


Таблица истинности JK-триггера:

J	K	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\neg Q_n$

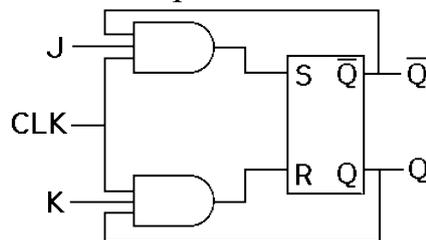
JK-триггер работает так же как RS-триггер, с одним лишь исключением: при подаче логической единицы на оба входа J и K состояние выхода триггера изменяется на противоположное, то есть выполняется операция инверсии (чем он отличается от RS-триггеров с доопределённым состоянием, которые строго переходят в логический ноль или единицу, независимо от предыдущего состояния). Вход J аналогичен входу S у RS-триггера. Вход K аналогичен входу R у RS-триггера.

Если с J входа через инвертор подать сигнал на K вход, то получится D-триггер.

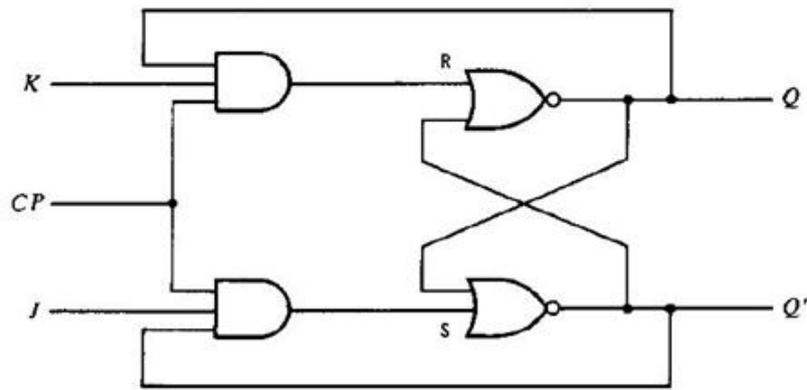


Если задать постоянные значения  $J=1$  и  $K=1$ , то получится T-триггер.

Пример упрощенной неработоспособной модели JK-триггера, который перебрасывается только от очень короткого тактового импульса:



Ниже эта же схема, в которой RS-триггер представлен более подробно:



(a) Logic diagram

$Q$	$J$	$K$	$Q(t+1)$
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

(b) Characteristic table

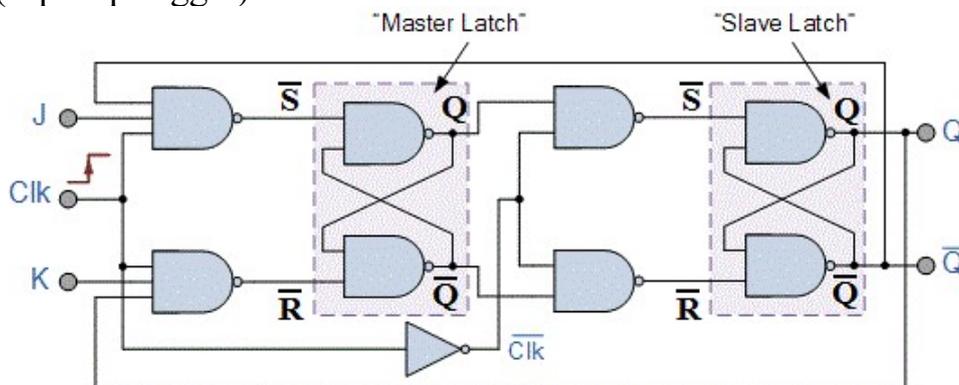
$Q$	$JK$			
	00	01	11	10
0			1	1
1	1			1

$Q(t+1) = JQ' + K'Q$

(c) Characteristic equation

Недостатком этой схемы является то, что при одновременном высоком уровне на входах J и K и постоянно высоком уровне напряжения на входе CP выходы схемы непрерывно переключаются с максимально высокой частотой, которая определяется быстродействием логических схем И и ИЛИ-НЕ.

Этот недостаток устранен в так называемой двухступенчатой схеме триггера (flip flop trigger):

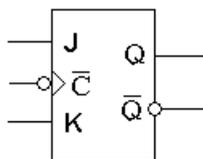


Здесь при высоком уровне на тактовом входе Clk входы J и K работают как входы Set и Reset для первого RS-триггера, за исключением одного состояния входов J=1 и K=1. Вход правой половины схемы при этом закрыт для изменений низким уровнем входа разрешения (инвертированным уровнем Clk).

На перепаде вниз на тактовом входе Clk закрывается для изменений левая половина схемы, и открывается правая половина. При этом состояние левого RS-триггера записывается в правый RS-триггер.

Если же на входах  $J=1$  и  $K=1$ , то непрерывного переключения выходов не происходит, так как одна из двух последовательных защелок всегда закрыта. Состояние выходов схемы в этом случае изменяется на противоположное состояние только по фронту вниз на тактовом входе схемы Clk.

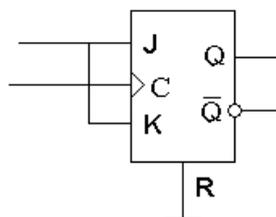
Соответственно обозначение схемы должно быть следующим:



### Т-триггер.

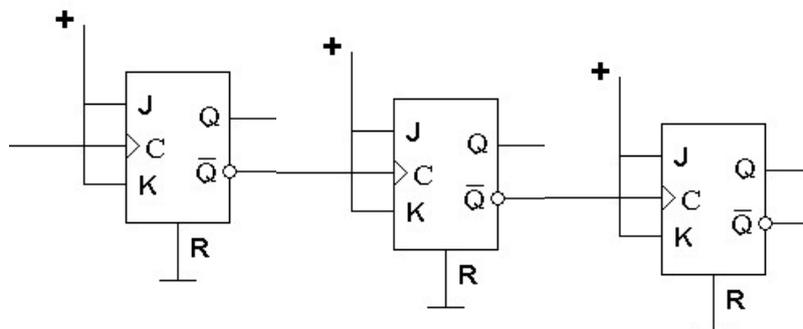
Если соединить входы  $J$  и  $K$  любого JK-триггера, то из JK-триггера получится так называемый Т-триггер, который переключается в противоположное состояние при каждом тактовом импульсе, если на входах высокий логический уровень, и остается в прежнем состоянии, если на входах низкий уровень (первая и четвертая строки таблицы истинности).

J	K	$Q_{n+1}$
0	0	$Q_n$
0	1	0
1	0	1
1	1	$\neg Q_n$



### Асинхронный счетчик.

На основе JK-триггера легко сделать делитель частоты импульсов на 2:



Асинхронный счетчик — несколько делителей частоты на 2 на JK-триггерах. Каждый инвертированный выход предыдущего триггера соединен с тактовым входом следующего триггера. Если тактовые входы инвертированы,

то неинвертированный выход соединяют с тактовым входом следующего триггера.

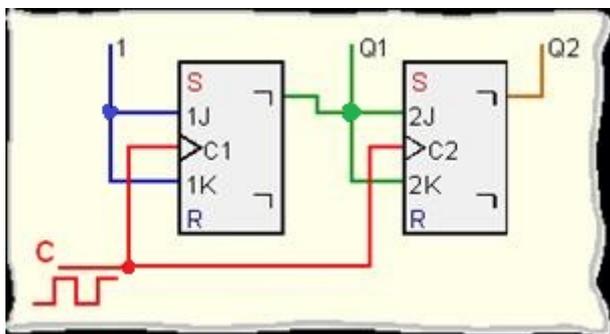
Входы J и K всех триггеров можно соединить и рассматривать, как вход разрешения счета общей схемы счетчика.

### **Синхронный счетчик. Реверсивный счетчик.**

Недостатком асинхронного счетчика является то, что биты счетчика меняются последовательно во времени, а не одновременно. В результате можно прочесть состояние счетчика, когда одна часть битов успела перейти в новое состояние, а другая часть — не успела.

Этот недостаток устранен в синхронном счетчике, в котором выходы всех битов изменяются одновременно.

Два младших бита синхронного счетчика могут работать по следующей схеме.



По фронту вверх тактового импульса состояния выходов изменяются, а по фронту вниз выходы можно прочесть без опасности, что один из выходов переключился, а другой — не успел.

Если битов больше двух, то следующий бит должен переключаться только в том случае, когда каждый из предыдущих битов равен логической единице. Это условие проверяют дополнительные схемы И:

*Синхронный четырёхразрядный счётчик*

