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.

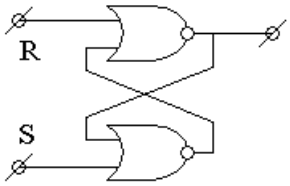


Схема состоит из двух логических схем 2ИЛИ-НЕ. Схема содержит два входа и один выход.

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.

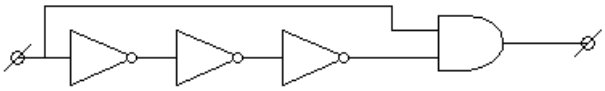
Рассматриваемый *RS*-триггер является элементарной ячейкой памяти, в которую можно записать одно из двух состояний 0 или 1. Записанное состояние остается в этой ячейке памяти и после того, как пропадает импульс записи.

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.



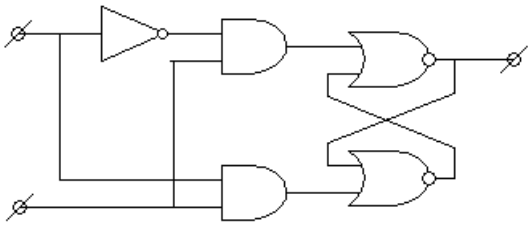
С учетом задержки распространения на выходе появляется короткий импульс при каждом перепаде вверх напряжения на входе схемы. Эту схему можно назвать одновибратором.

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.**

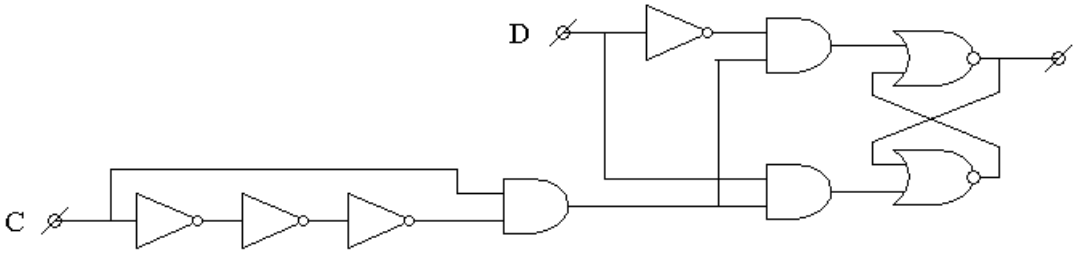
В схеме, приведенной ниже, справа *RS*-триггер, выполненный на двух схемах ИЛИ-НЕ. Напряжение с верхнего входа схемы проходит на выход схемы и может установить или сбросить *RS*-триггер, но только в том случае, когда на нижнем входе разрешающее высокое напряжение.



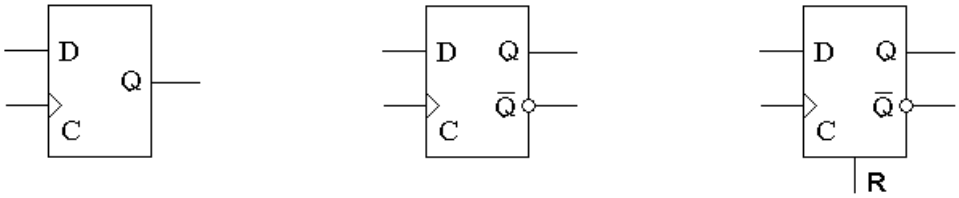
Если на нижний вход подать короткий импульс, то *RS*-триггер установится или сбросится в зависимости от состояния верхнего входа именно в момент короткого синхроимпульса.

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.

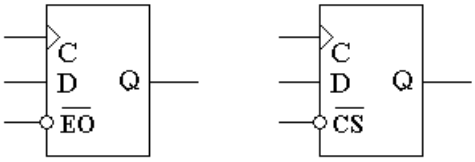


Вся эта схема *D*-триггера обозначается, как



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

Выходы с тремя состояниями.



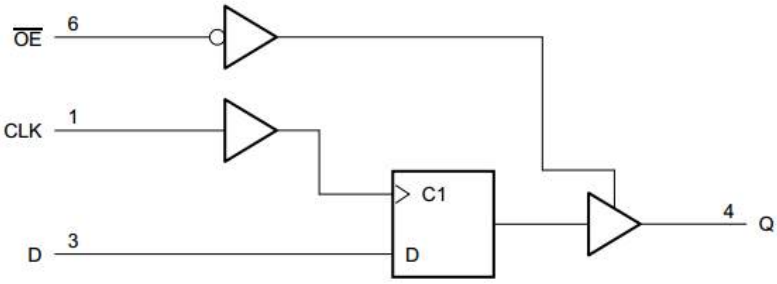
Выходы с тремя состояниями. Здесь $\overline{EO}$ (enable output) — разрешение выхода низким уровнем энергии. Если на входе $\overline{EO}$ высокий уровень энергии, то выход находится в состоянии с высоким выходным сопротивлением. Обозначение $\overline{CS}$ (chip select) действует ровно так же, как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

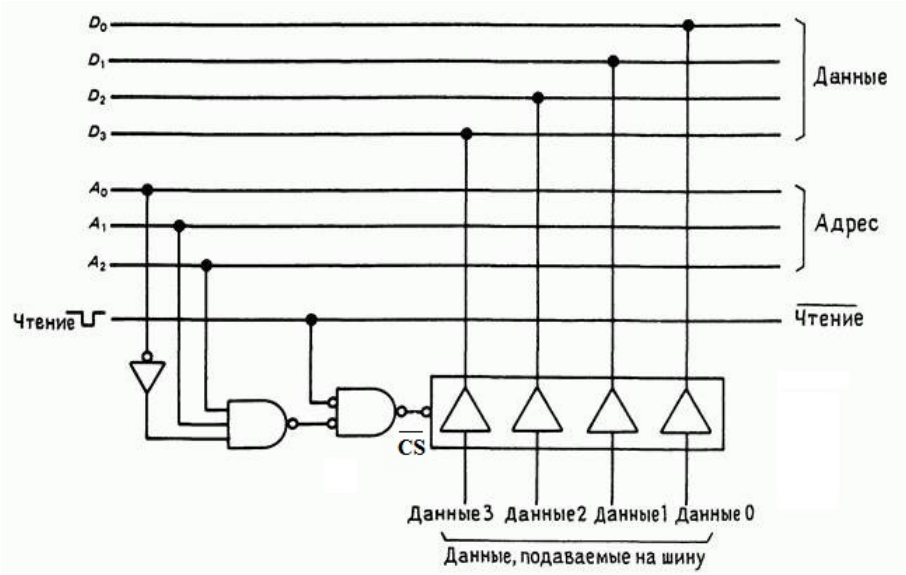
LOGIC DIAGRAM (POSITIVE LOGIC)



<u>Регистр. Шина. Память.</u>

Регистр — несколько параллельных D-триггеров.

Шина данных, адресная шина. Пример регистра, который передает данные на шину, если на адресной шине установлен адрес $6=110_2$:



Микросхемы памяти — множество регистров с разными последовательными адресами.