

R-S-триггер

Объяснение работы триггера начинаем с подачи на входы R и S логического нуля (0-зеленый цвет), а на другие входы ячеек за счет обратной положительной связи соответственно Q и не Q. Через ячейки ИЛИ они пройдут без изменения с последующей инверсией (см. рис 1). Следовательно информация на выходах не изменилась. Такая операция обеспечивает хранение информации в триггере.

Если на два входа R и S подаем логические единицы (1), то на выходах получаем нули независимо от того, что подадим на вторые входы ячеек ИЛИ.

Данная ситуация не имеет смысла и является запрещенной.

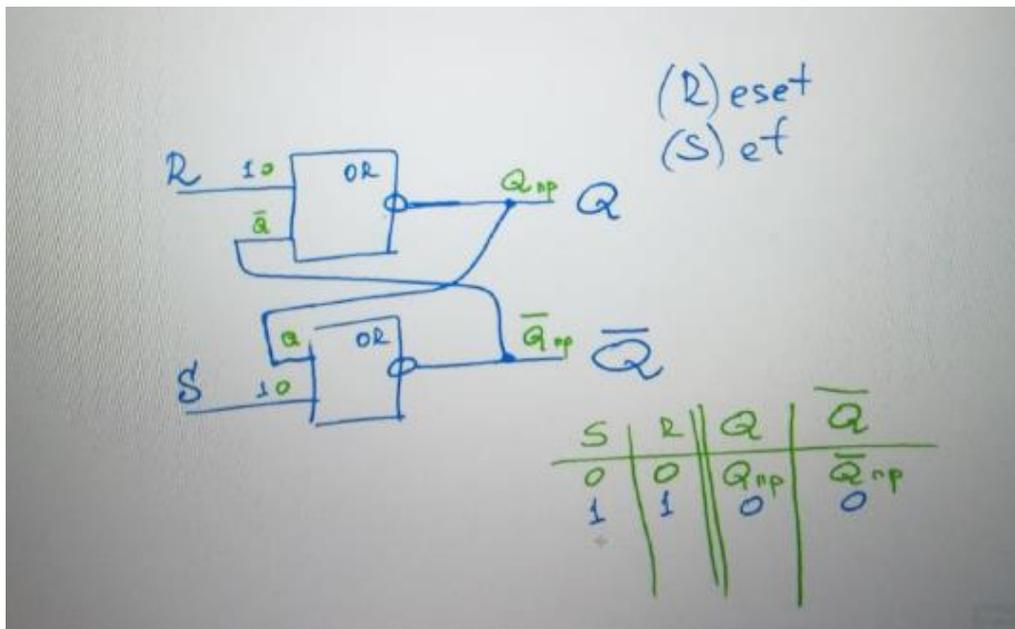


Рис. 1

Подаем на вход S = «1» (рис 2), на выходе этой ячейки получаем не Q равное «0», а на входе первой ячейки R = «0» и не Q = «0», то на прямом выходе произошла запись бита «1».

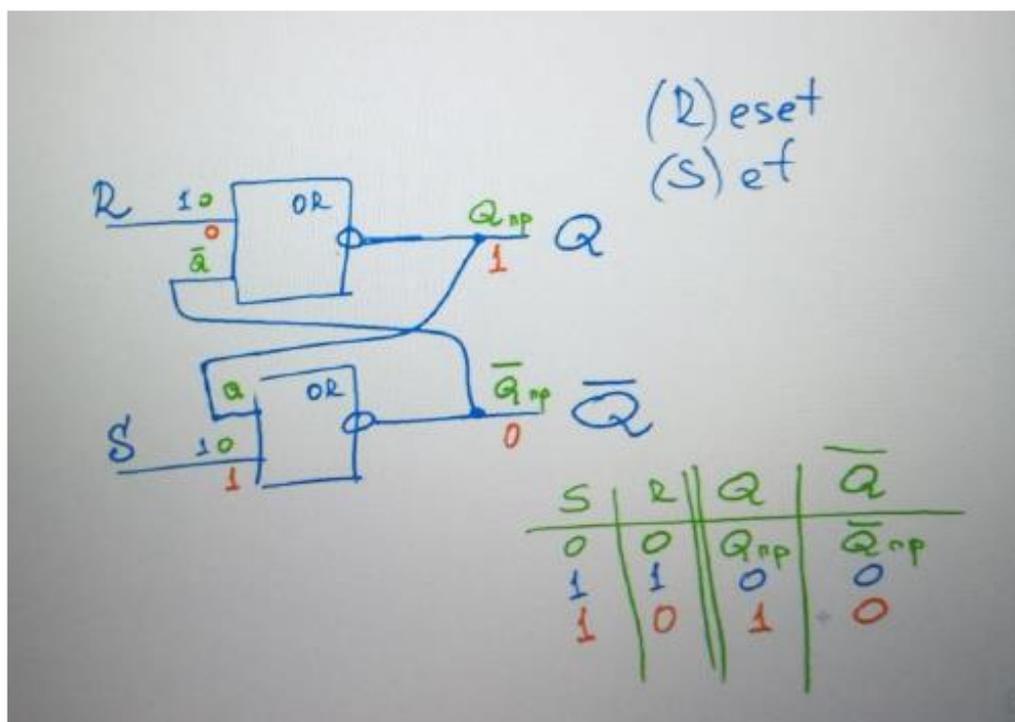


Рис. 2

Все рассуждения надо начинать с подачи на входы «1», так как подача «0» не изменяет ситуацию.

Подаем на вход R = «1» (рис 3), на втором входе этой ячейки не Q равное «0», а на выходе первой ячейки получаем «0» (все черного цвета); S = «0» и Q = «0», то на выходе не Q произошла запись бита «1».

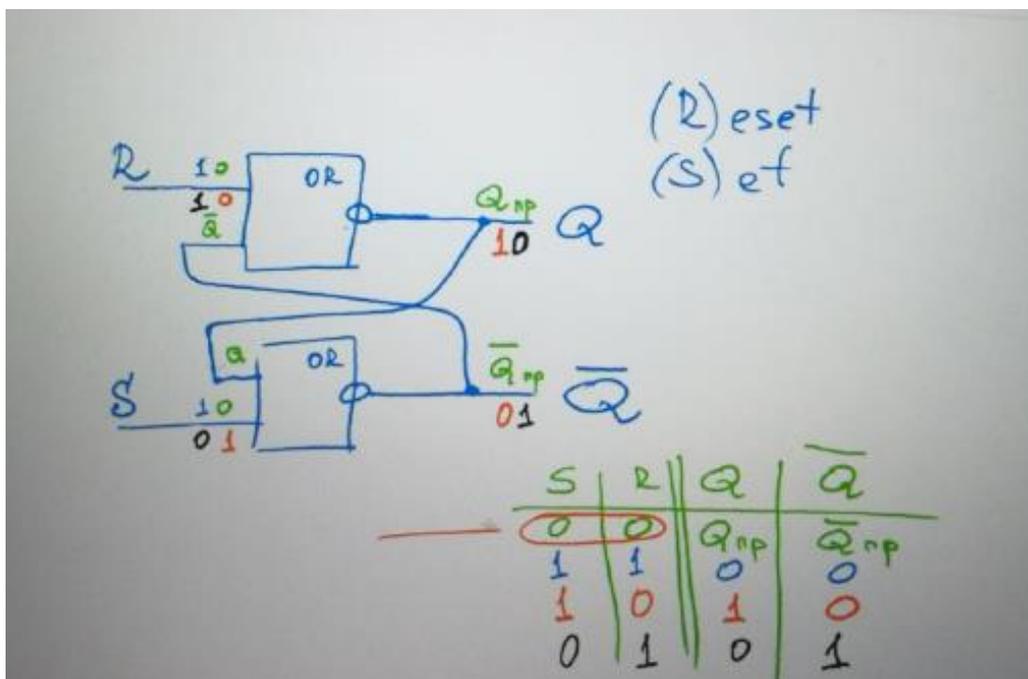


Рис. 3

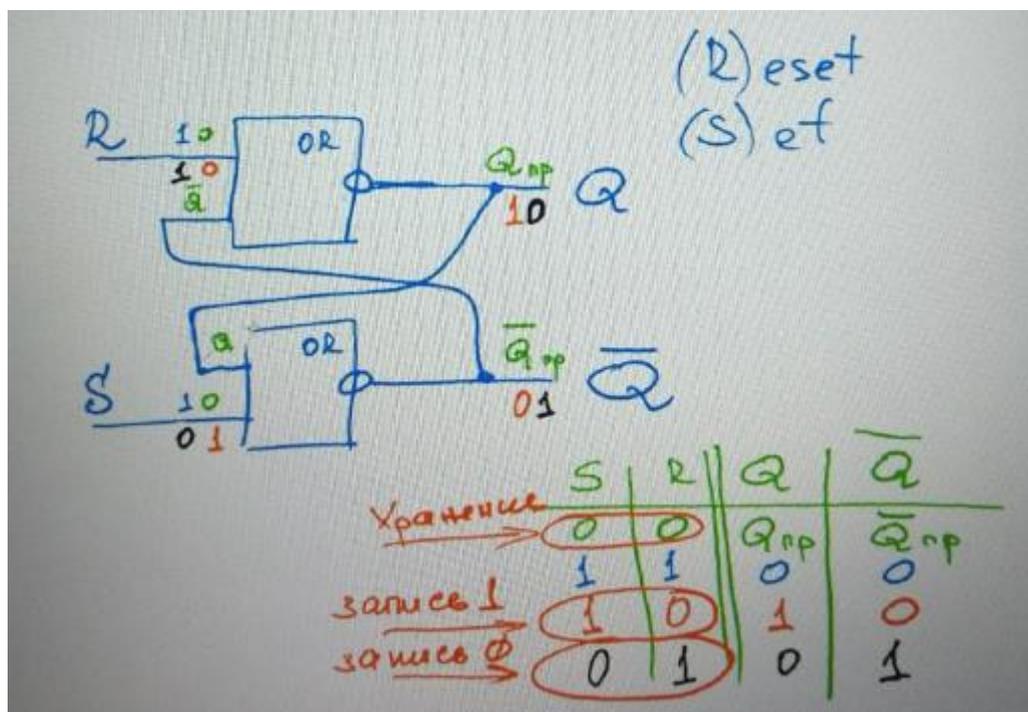


Рис 4

В итоге на рисунке 4 подведен итог записи «1» и «0» в R-S-триггере.

Выше мы рассмотрели работу R-S триггера и можем предсказать его состояние (таблица на рисунке 4). Но что будет на выходе и когда – зависит от R и S. Они взаимосвязаны, и мы не можем определить с начала что будет храниться в триггере, а потом определить когда. Для этого модифицируем схему триггера, чтобы разделить что и когда будет храниться в

триггере. Надо обеспечить управление временем, для этого добавим два элемента И с инвертором (разделяем время и данные, рис 5). Если $CLK = 0$, то элементы И не будут пропускать сигнал независимо от того, что будет на других входах этих элементов (ситуация установок отмечена черным). Если $CLK = 1$ и $D = 1$, то во втором элементе И – 1 и 1, а это ситуация Set. При $D = 0$ – ситуация – Reset. Таким образом мы получили систему что будет записываться в триггер и когда.

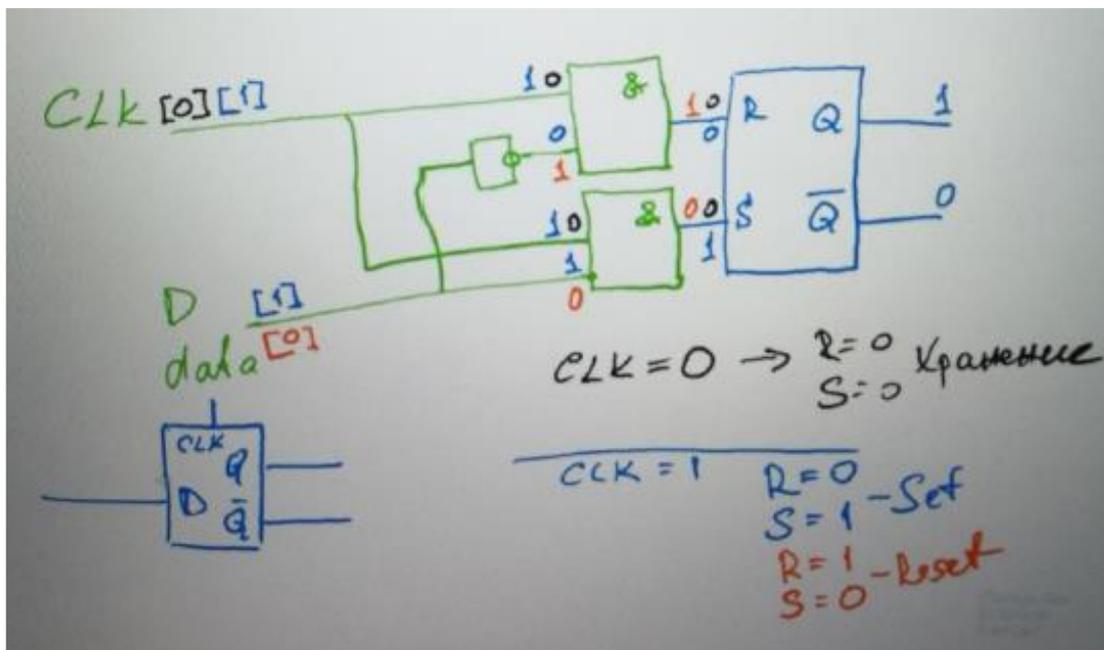


Рис. 5

Выше рассмотренные триггеры являются однокаскадными («защелки»), и они зависят только от своих входов. Разберем двухкаскадный триггер, в котором хранится одновременно сразу два значения, чтобы избежать различного рода помех. В таких триггерах устанавливается сначала значение, которое мы должны хранить, а потом это значение при воздействии синхронизирующего сигнала копируется в следующий каскад (рисунки 6,7,8).

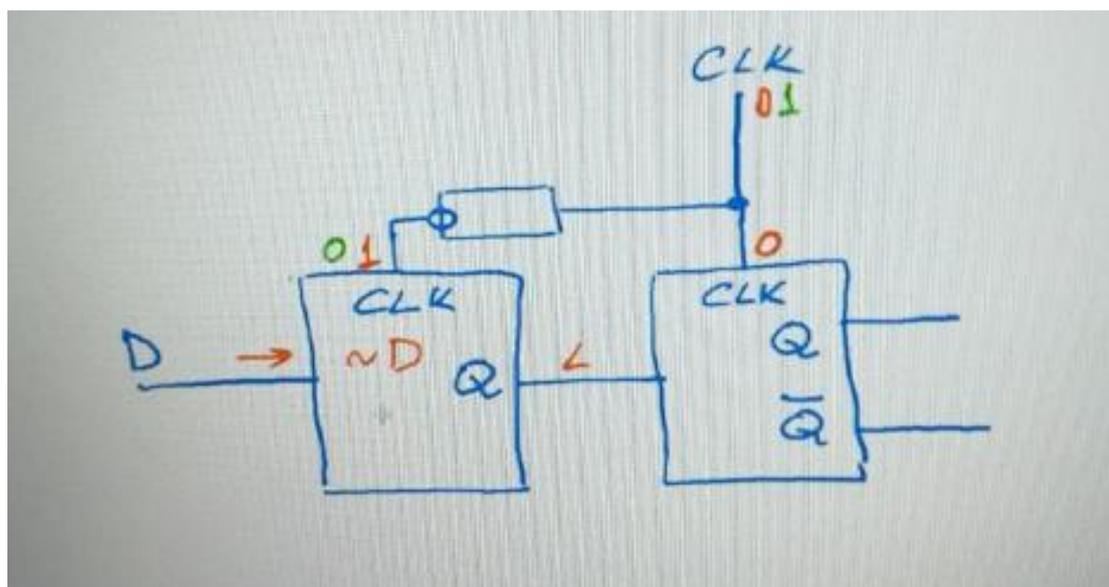


Рис. 6

Допустим $CLK = 0$, второй триггер не работает, а в первый триггер записывается все, что приходит на вход D и будет в нем храниться. Если $CLK = 1$, то с этого момента левый

триггер никак не будет реагировать на значения D. Зато значение левого триггера скопируется в правый триггер. Значение копируется по фронту нарастания.

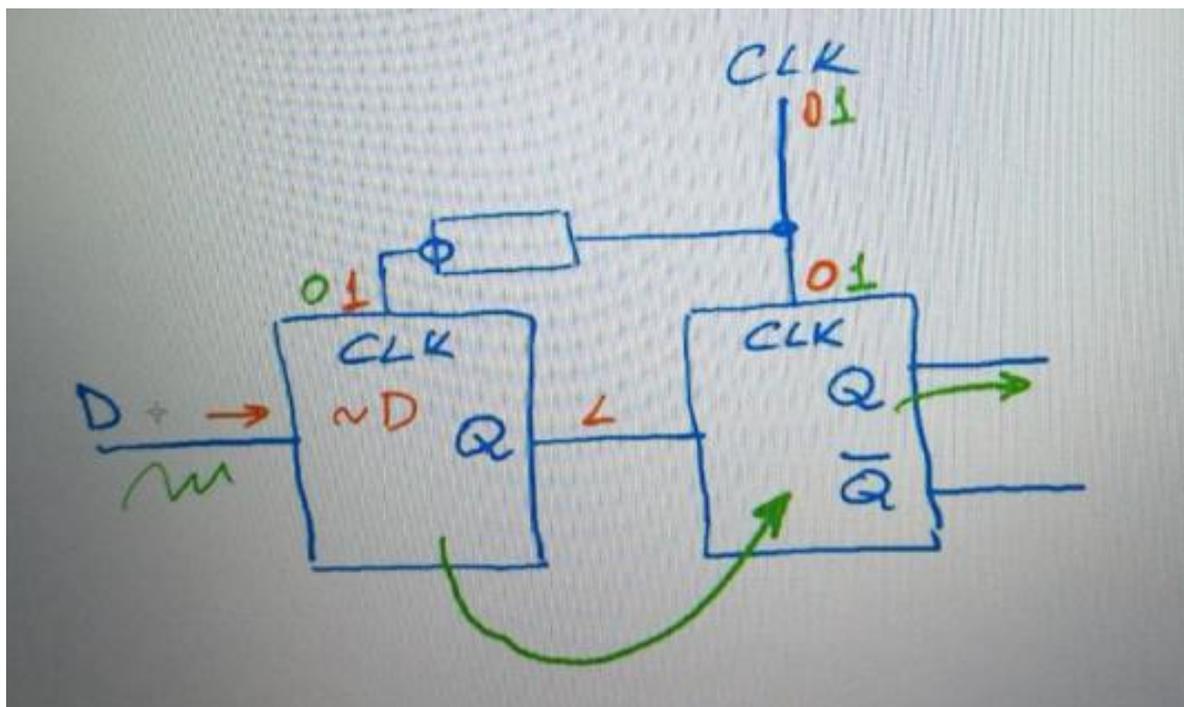


Рис. 7

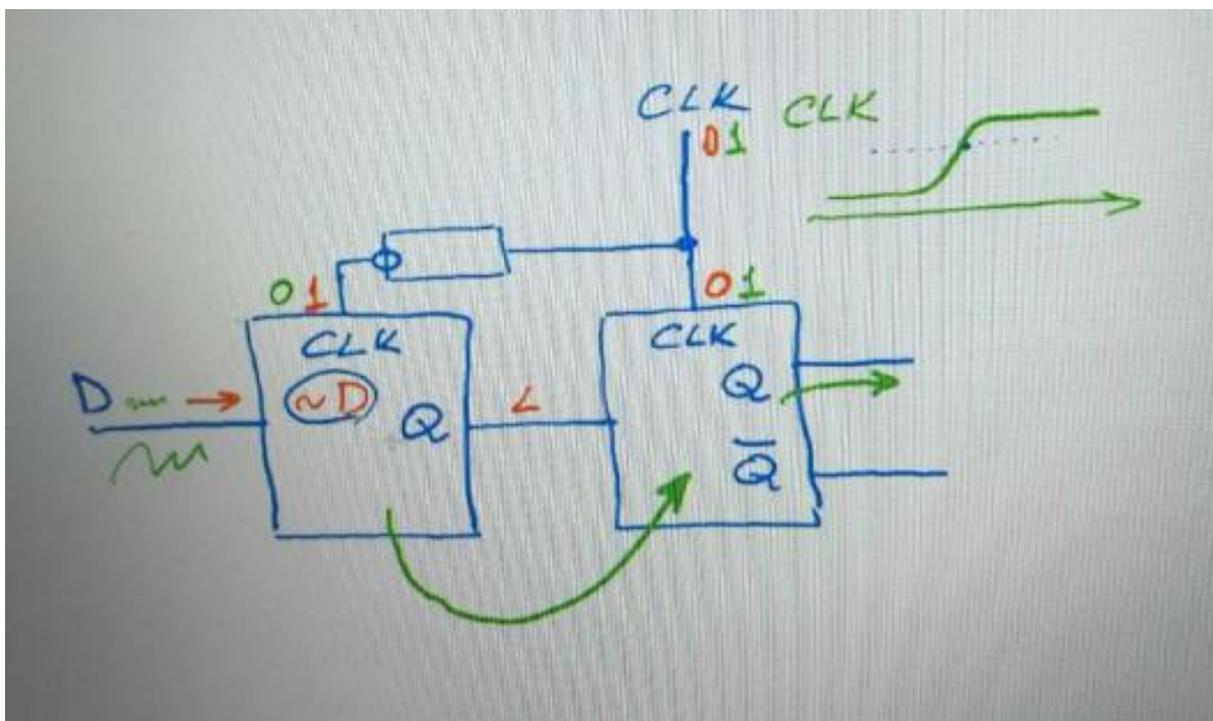


Рис. 8