

**Тема:**  
**Триггеры**

# Триггер

Простейший цифровой автомат, т.е. устройство с памятью. При наличии электропитания способен на длительное время запоминать одно из двух устойчивых состояний (0 или 1) и может переключаться между ними под управлением внешних сигналов.

Таким образом, триггер хранит 1 бит информации. Применяется самостоятельно, а также служит основой для построения других устройств: регистров, счетчиков.

# Классификация триггеров

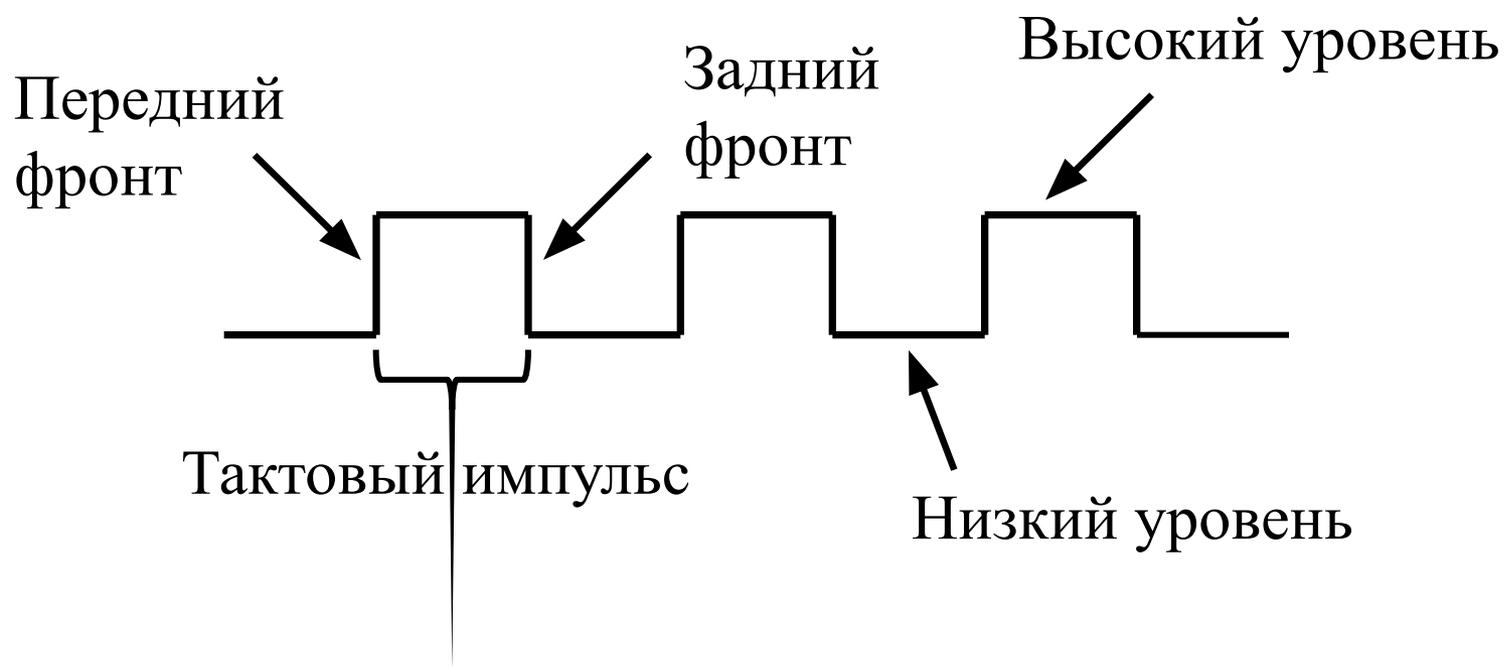
1. По логическому функционированию:  
RS, D, T, JK (основные типы).
2. По способу записи информации:
  - асинхронные;
  - синхронные:
    - со статическим управлением (стробируемые);
    - с динамическим управлением (тактируемые).
3. По количеству ступеней:
  - одноступенчатые;
  - двухступенчатые.

В **асинхронных** триггерах запись информации (переключение состояния) осуществляется в момент подачи сигнала на информационные входы.

В **синхронных** триггерах запись возможна только в присутствии разрешающего сигнала  $C$  (Clock), т.е. сигнала синхронизации.

Для цифровых автоматов синхронизация **очень важна**, поскольку позволяет согласовывать во времени процессы чтения и записи, происходящие в разных частях схемы, реализуя, таким образом, алгоритм работы устройства.

**Сигнал синхронизации** — последовательность дискретных импульсов стабильной частоты.



**Синхронный триггер со статическим управлением** воспринимает информационные сигналы во время действия активного уровня на входе  $C$ , т.е. пока  $C=1$ , происходит постоянная перезапись информации, а когда  $C=0$ , происходит фиксация состояния триггера.

**Синхронный триггер с динамическим управлением** (иначе говоря, фронтной триггер) воспринимает информационные сигналы в моменты переключений синхроимпульса ( $0 \rightarrow 1$  и  $1 \rightarrow 0$ ), т.е. в моменты прихода переднего или заднего фронта сигнала.

Вход  $C$  называется **прямым динамическим**, если переключение триггера происходит в момент прихода переднего фронта, **инверсным динамическим** – если переключение происходит в момент прихода заднего фронта.

**В одноступенчатых триггерах** для запоминания информации используется только одна ступень.

Возникают проблемы при записи и считывании информации в пределах одного такта.

Что считано? Старая информация или новая?

**Двухступенчатые триггеры** состоят из двух одноступенчатых, работающих в противофазе. Работают в 2 раза медленнее, но решают проблему одноступенчатых триггеров: когда вторая ступень еще хранит старую информацию, первая уже может принимать новую.

# Асинхронный RS-триггер на элементах ИЛИ-НЕ

S	R	$Q_i$	$\overline{Q}_i$
0	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
0	1	0	1
1	0	1	0
1	1	*	*

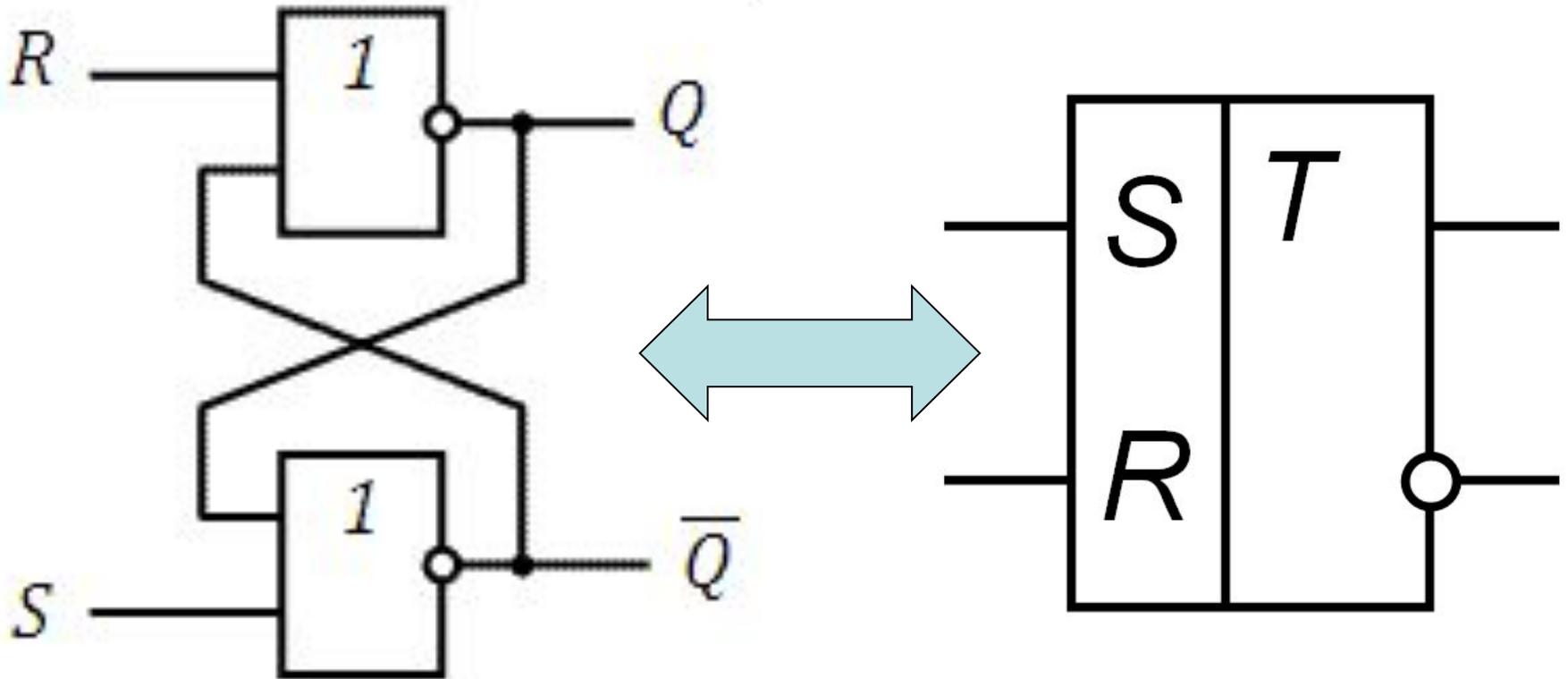
\* – неопределенное состояние.

R – Reset (сброс).

S – Set (установка).

Это простейший триггер, служащий основой для построения более сложных триггеров.

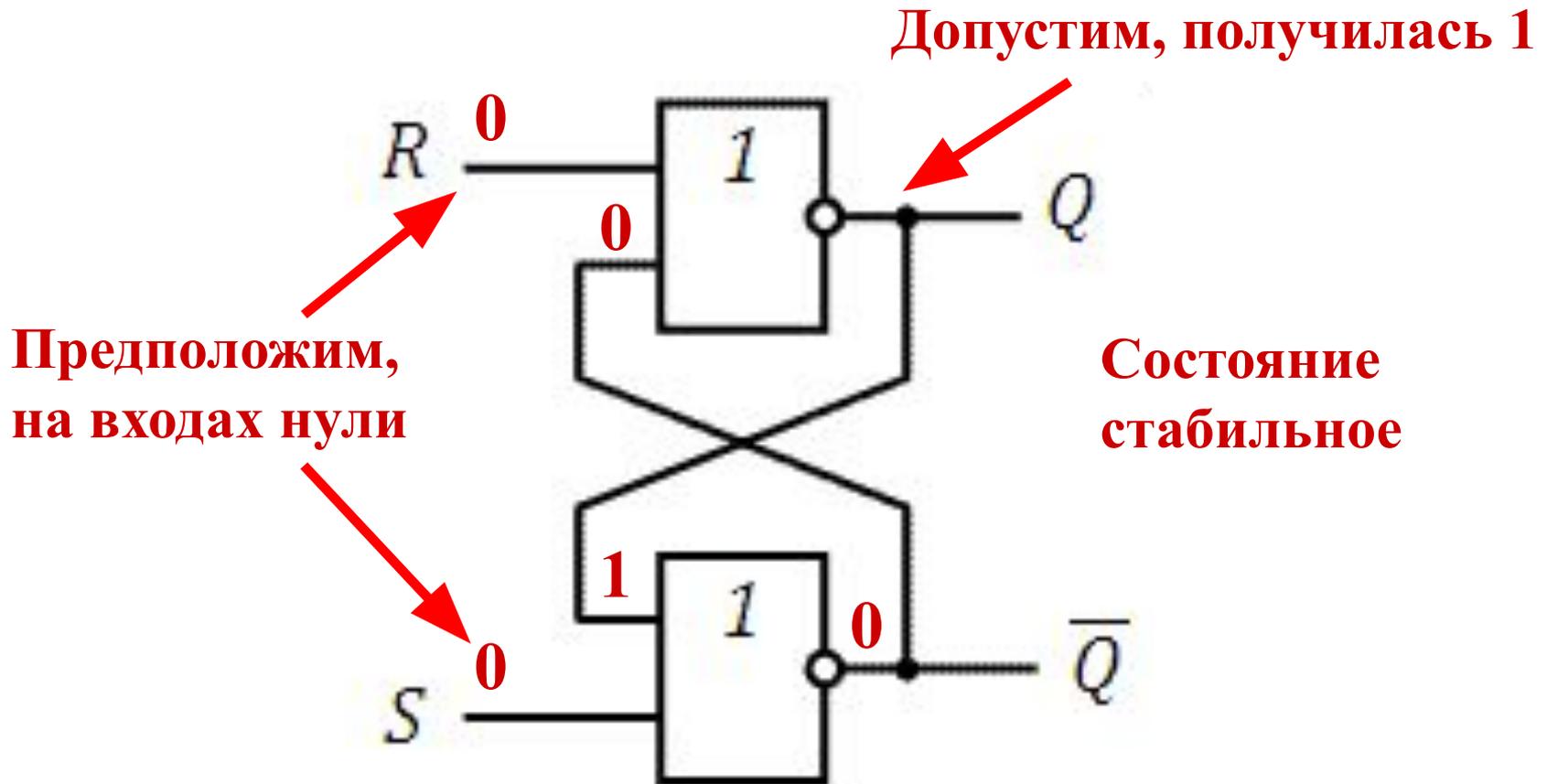
# Асинхронный RS-триггер на элементах ИЛИ-НЕ



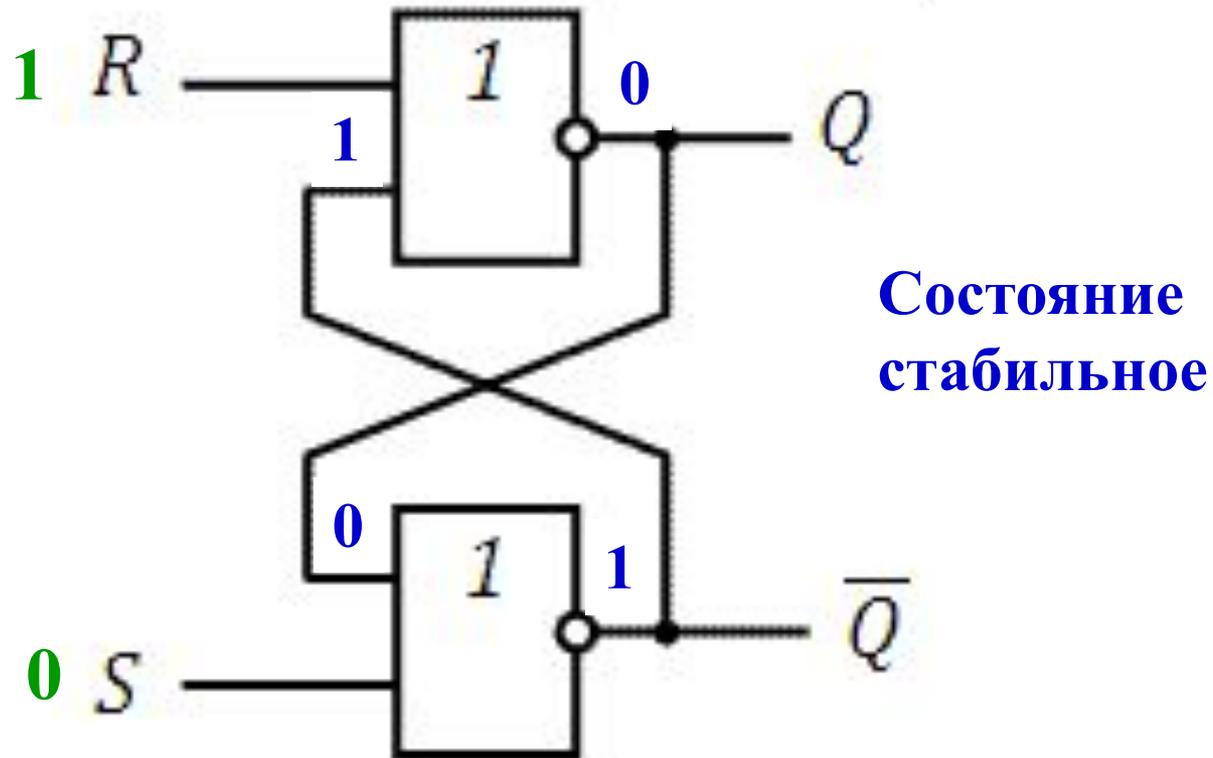
Поскольку реальные времена срабатывания логических элементов зависят от незначительных отклонений в процессе их изготовления, то при включении питания триггер непредсказуемо принимает одно из двух состояний.

Это приводит к необходимости выполнять первоначальную установку триггера в требуемое исходное состояние.

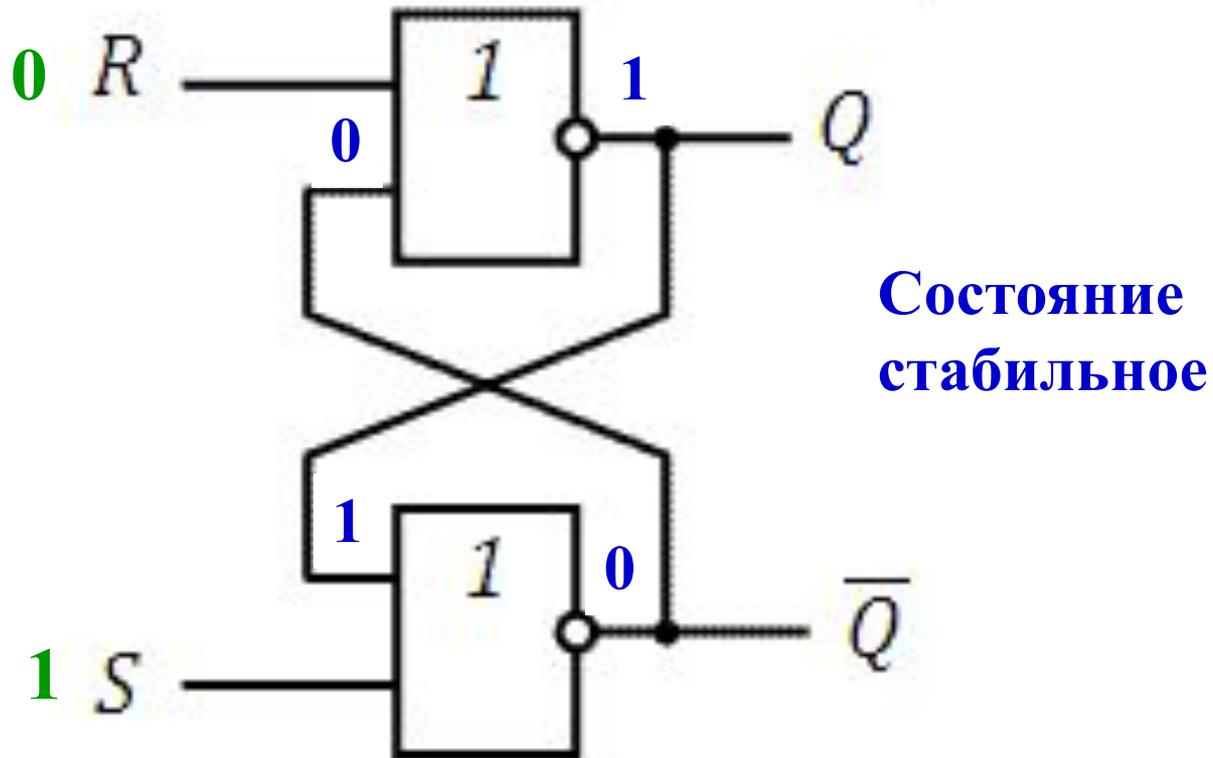
# Асинхронный RS-триггер на элементах ИЛИ-НЕ (хранение)



# Асинхронный RS-триггер на элементах ИЛИ-НЕ (сброс)

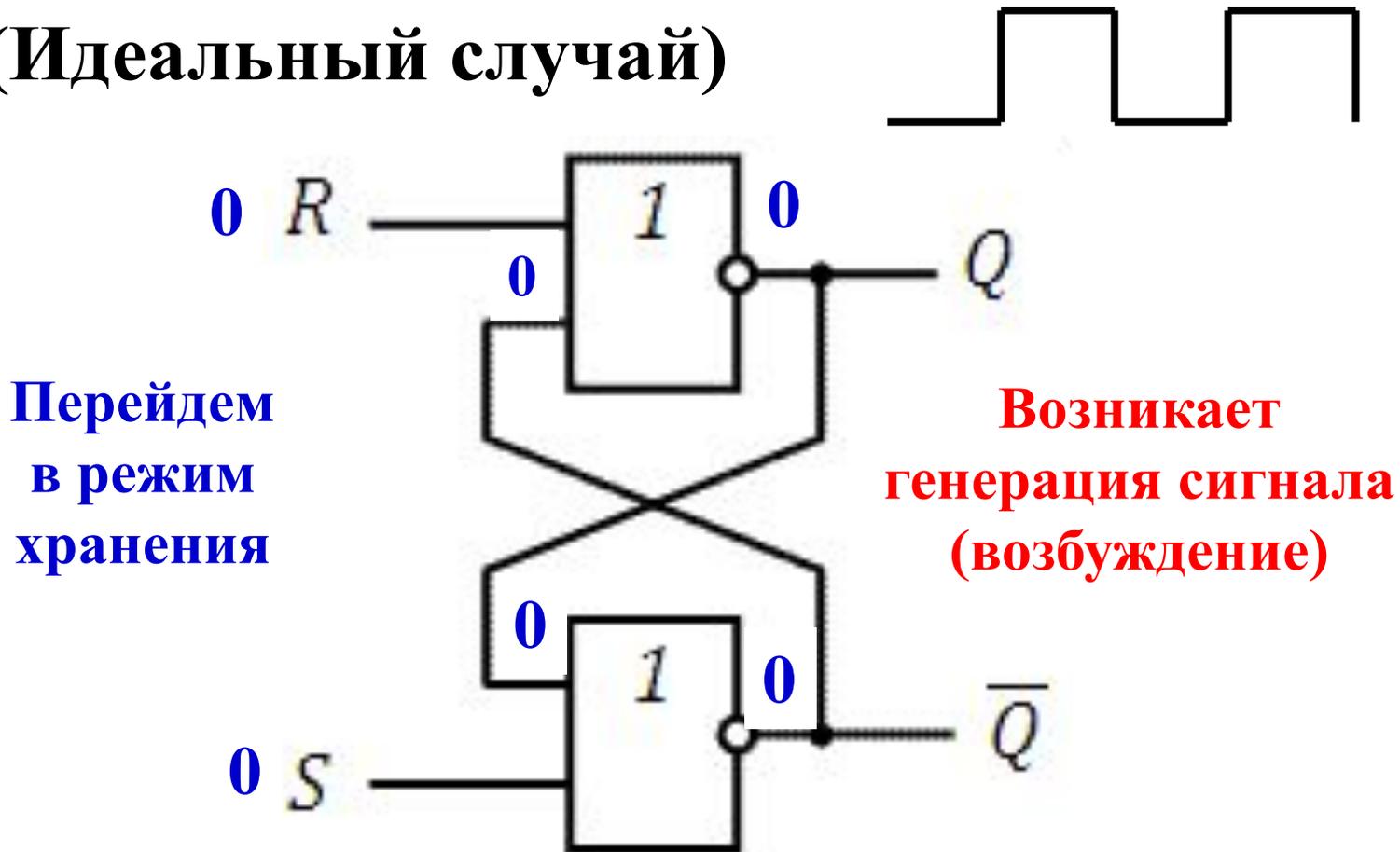


# Асинхронный RS-триггер на элементах ИЛИ-НЕ (установка)

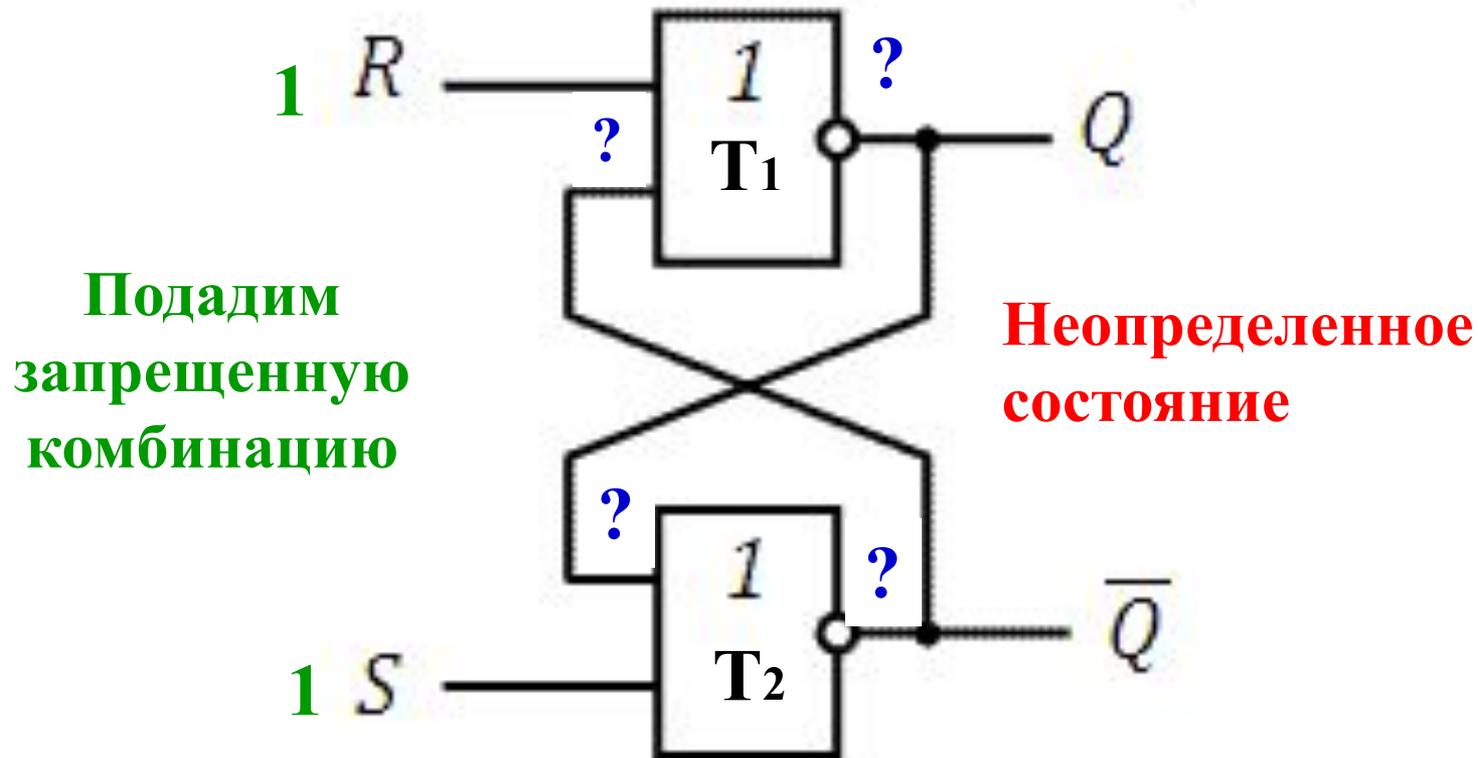


# Асинхронный RS-триггер на элементах ИЛИ-НЕ (запрещенная комбинация)

(Идеальный случай)



# Асинхронный RS-триггер на элементах ИЛИ-НЕ (запрещенная комбинация) (В реальности)



# Асинхронный RS-триггер на элементах И-НЕ

S	R	$Q_i$	$\overline{Q}_i$
1	1	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	0	0	1
0	1	1	0
0	0	*	*

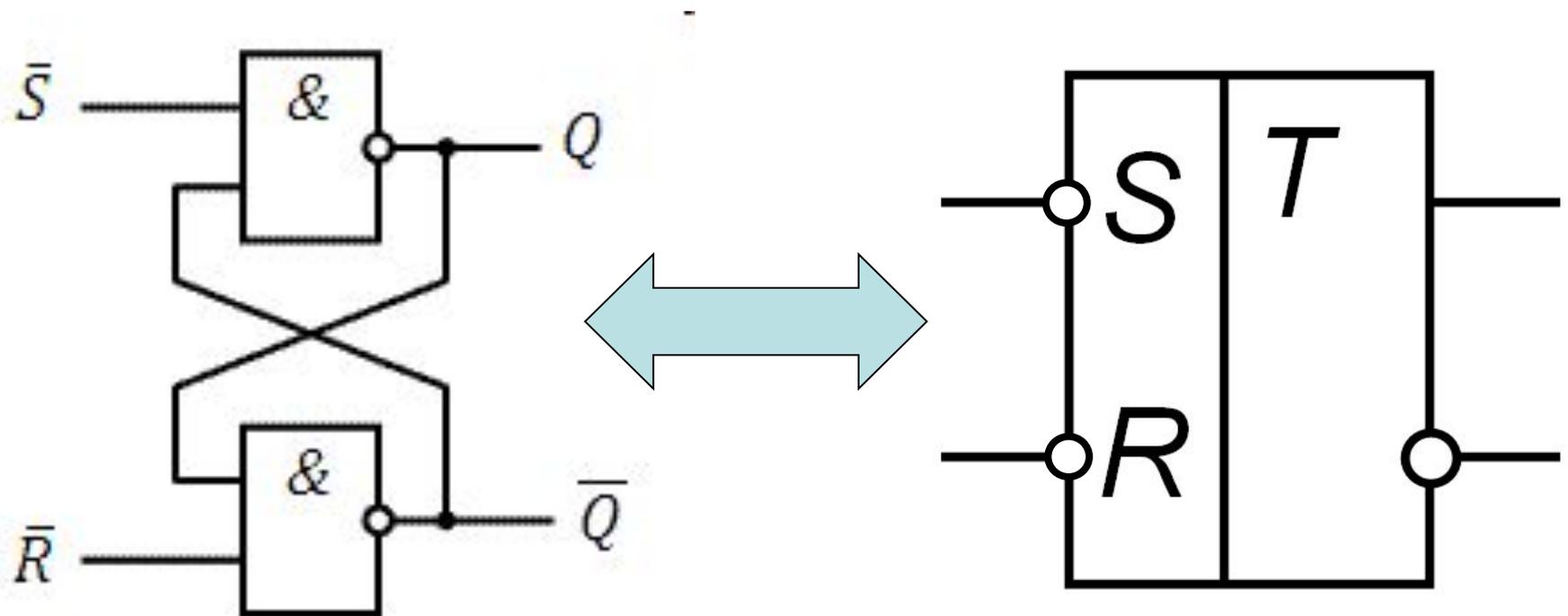
\* – неопределенное состояние.

R – Reset (сброс).

S – Set (установка).

Это простейший триггер, служащий основой для построения более сложных триггеров.

# Асинхронный RS-триггер на элементах И-НЕ



**Возьмем его за основу...**

# Синхронный RS-триггер со статическим управлением

C	S	R	$Q_i$	$\overline{Q}_i$
0	X	X	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	0	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	0	1	0	1
1	1	0	1	0
1	1	1	*	*

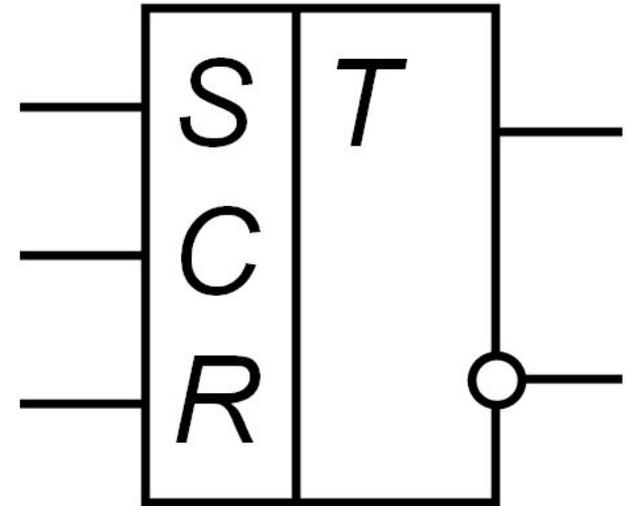
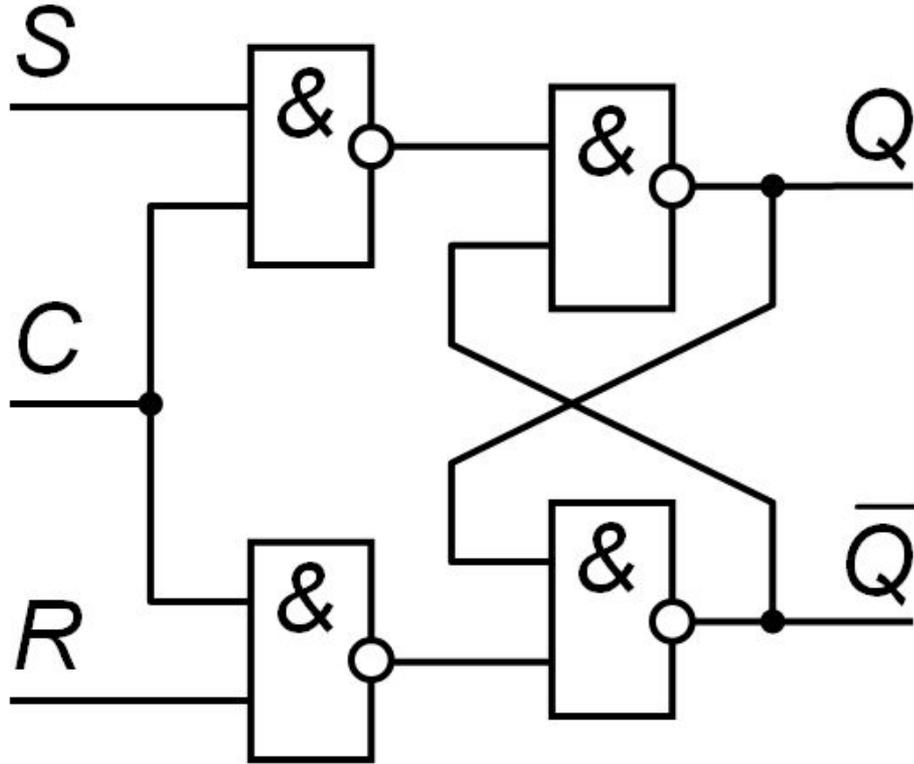
\* – неопределенное состояние.

X – любое значение.

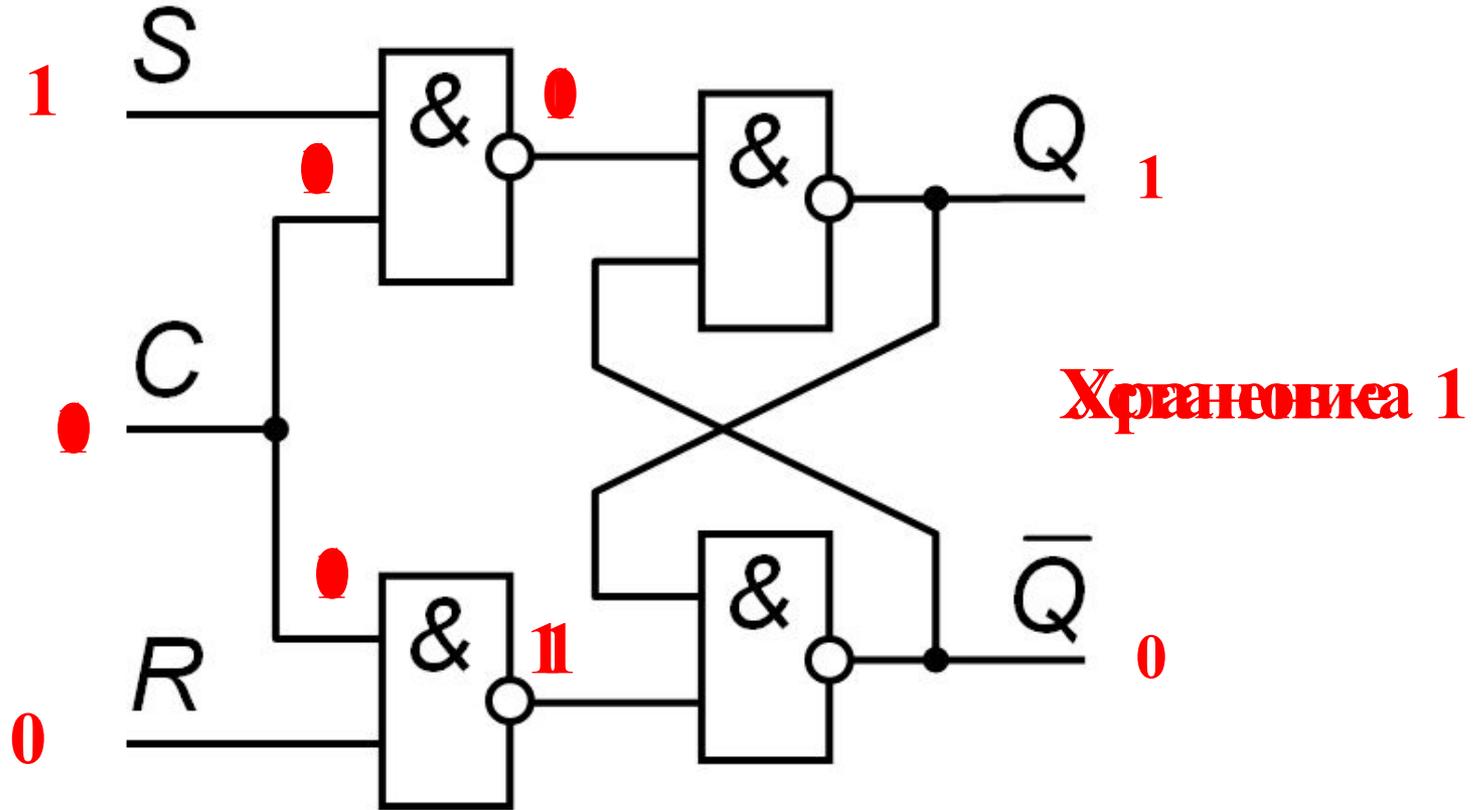
R – Reset (сброс).

S – Set (установка).

# Синхронный RS-триггер со статическим управлением



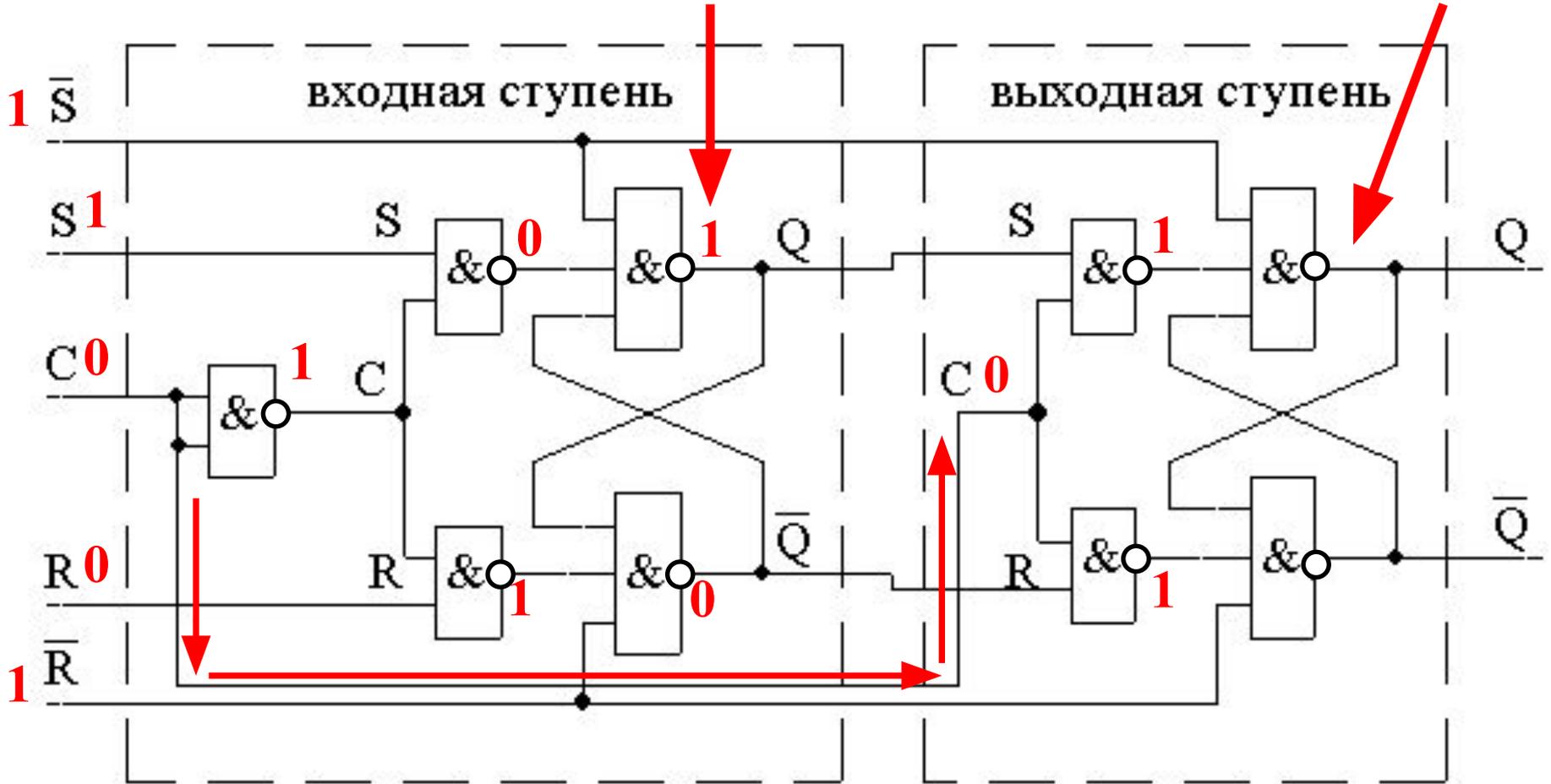
# Синхронный RS-триггер со статическим управлением



# Синхронный двухступенчатый RS-триггер с асинхронными входами и статическим управлением

**Записано новое значение**

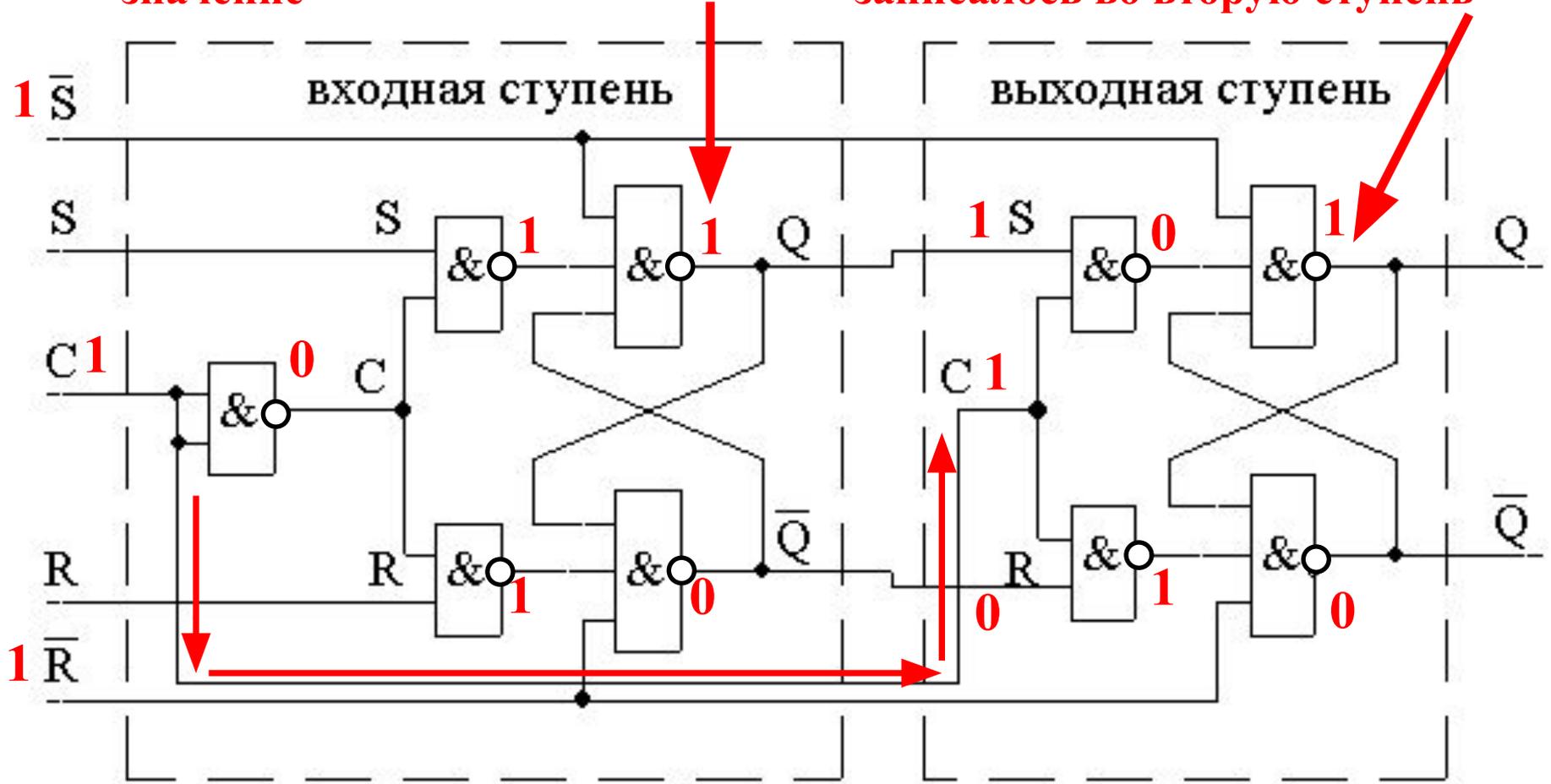
**Хранится старое значение**



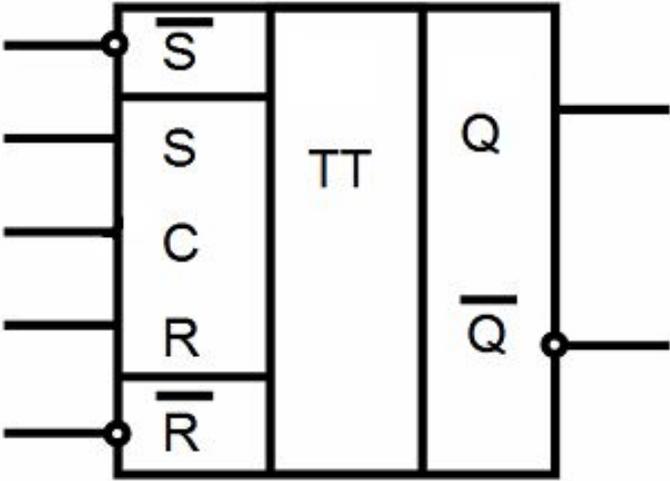
# Синхронный двухступенчатый RS-триггер с асинхронными входами и статическим управлением

Хранится поступившее ранее значение

Поступившее значение записалось во вторую ступень



# Синхронный двухступенчатый RS-триггер с асинхронными входами и статическим управлением



## Синхронный RS-триггер с динамическим управлением

C	S	R	$Q_i$	$\overline{Q}_i$
0	x	x	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	x	x	$Q_{i-1}$	$\overline{Q}_{i-1}$
1->0	x	x	$Q_{i-1}$	$\overline{Q}_{i-1}$
0->1	0	0	*	*
0->1	1	0	0	1
0->1	0	1	1	0
0->1	1	1	$Q_{i-1}$	$Q_{i-1}$

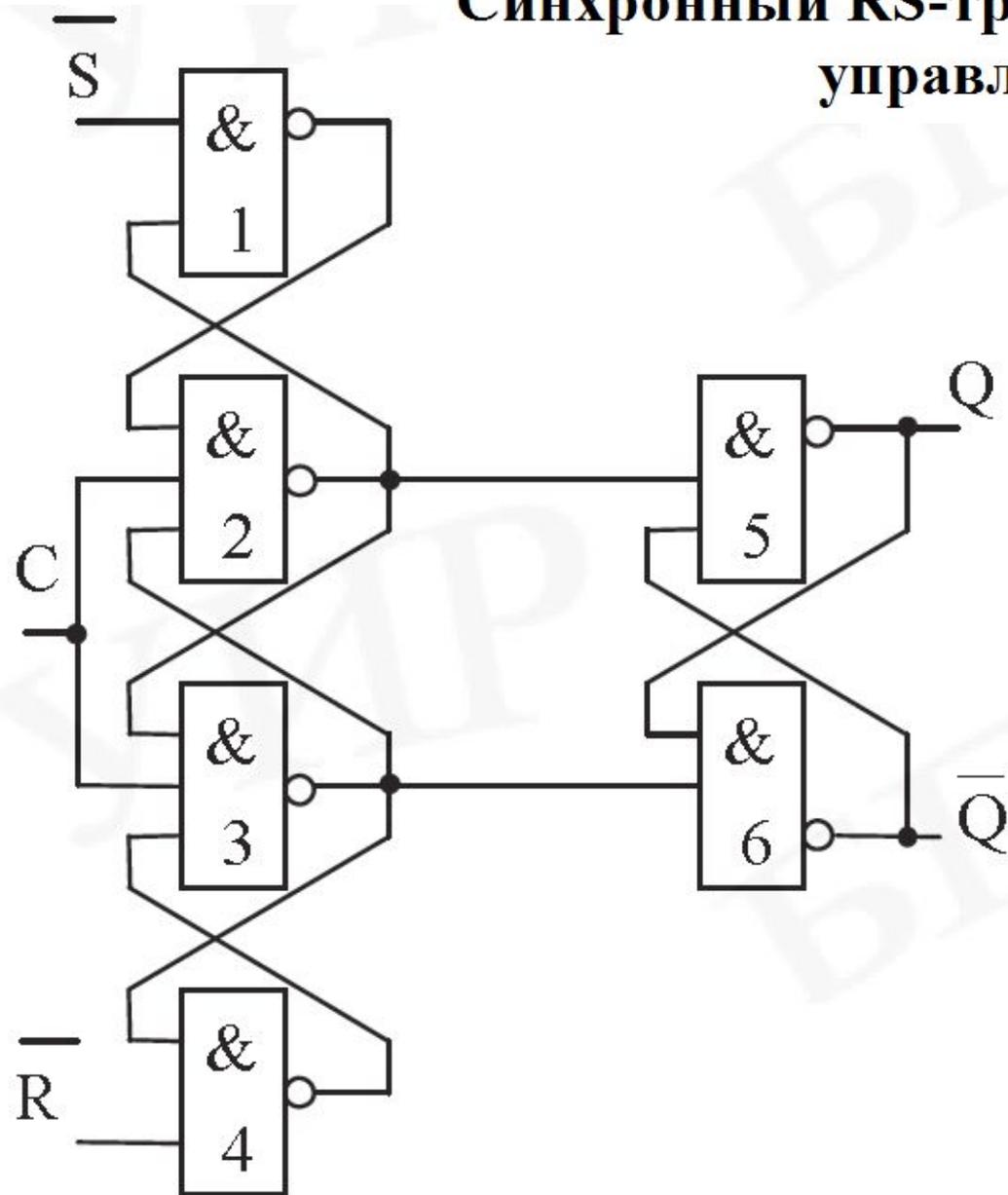
\* – неопределенное состояние.

X – любое значение.

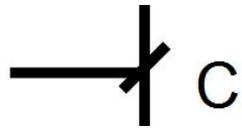
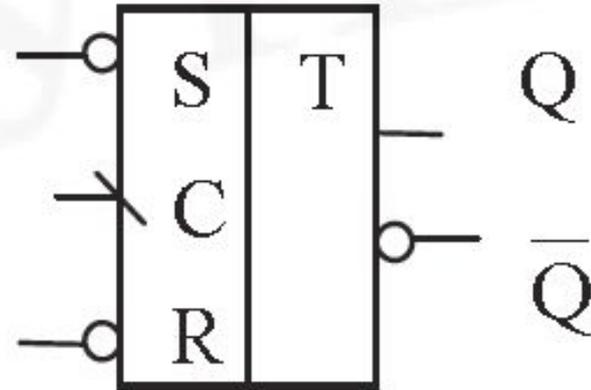
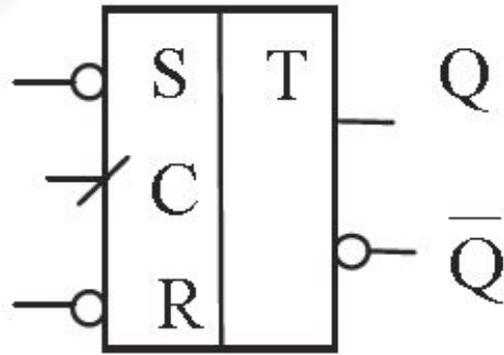
R – Reset (сброс).

S – Set (установка).

## Синхронный RS-триггер с динамическим управлением



# Синхронный RS-триггер с динамическим управлением



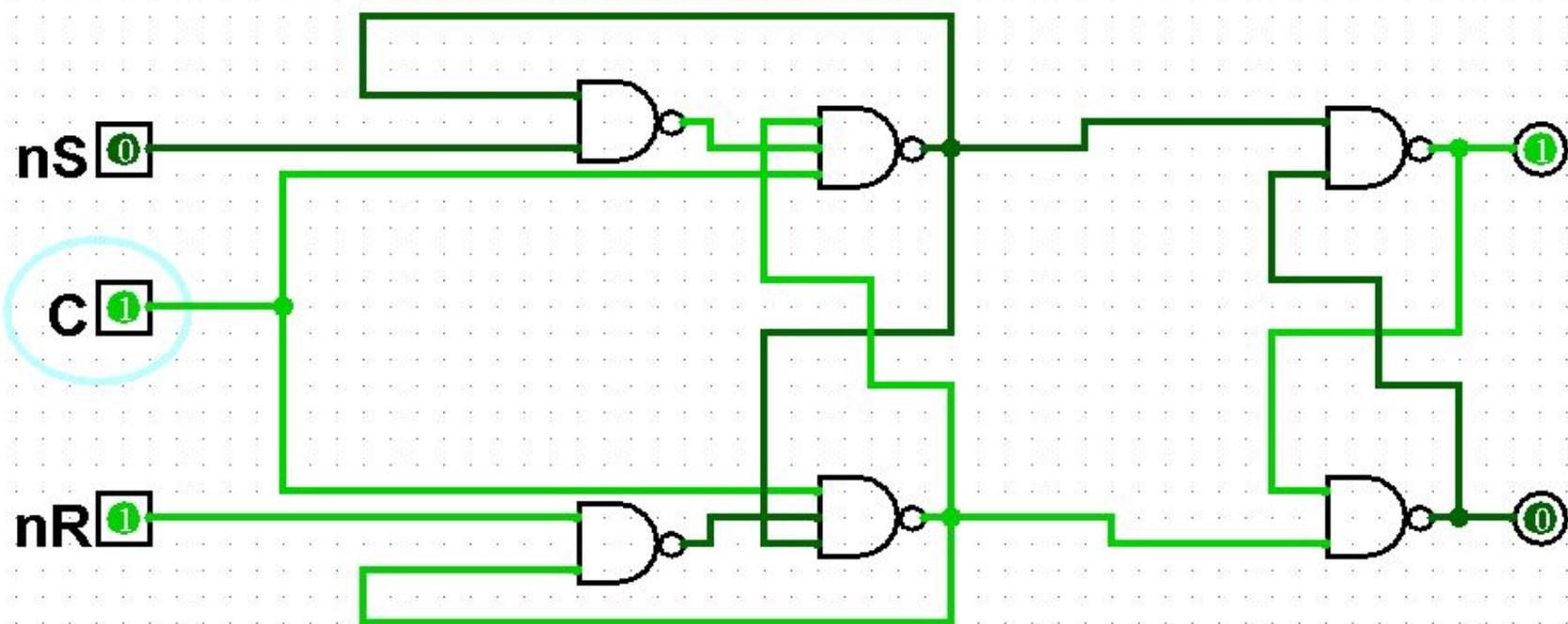
– работает по переднему фронту



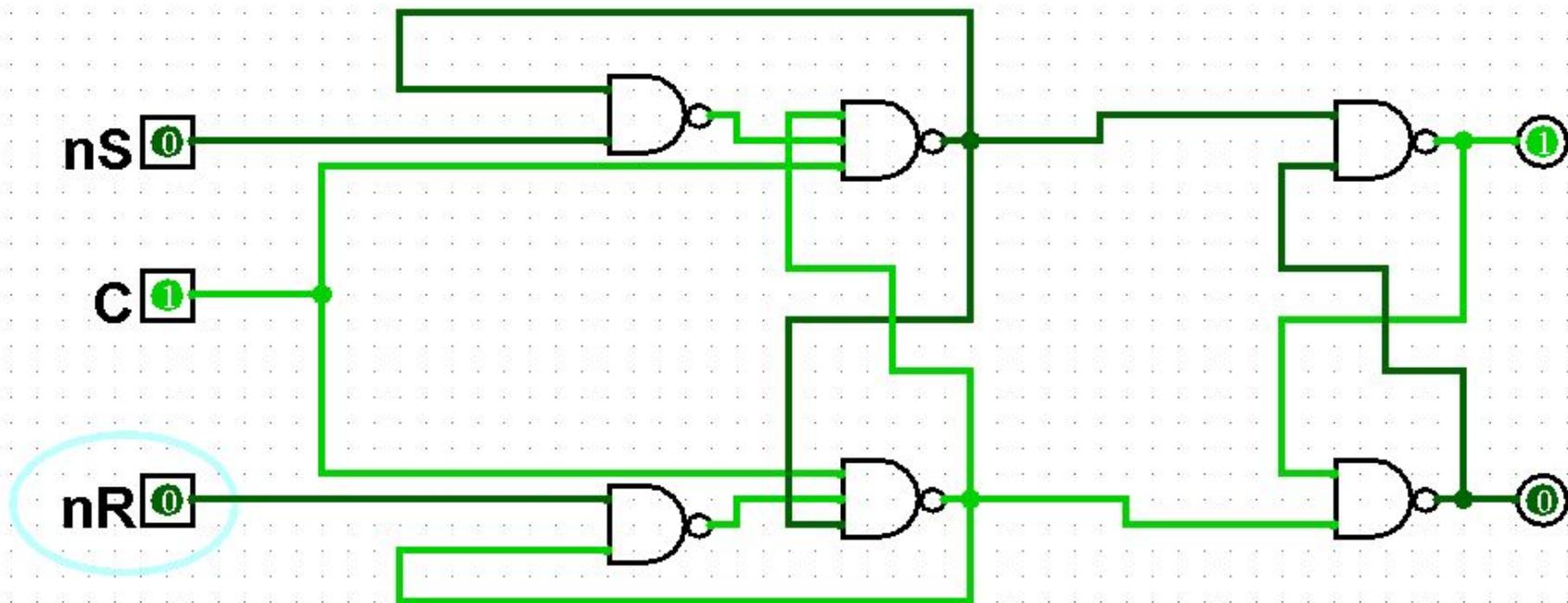
– работает по заднему фронту



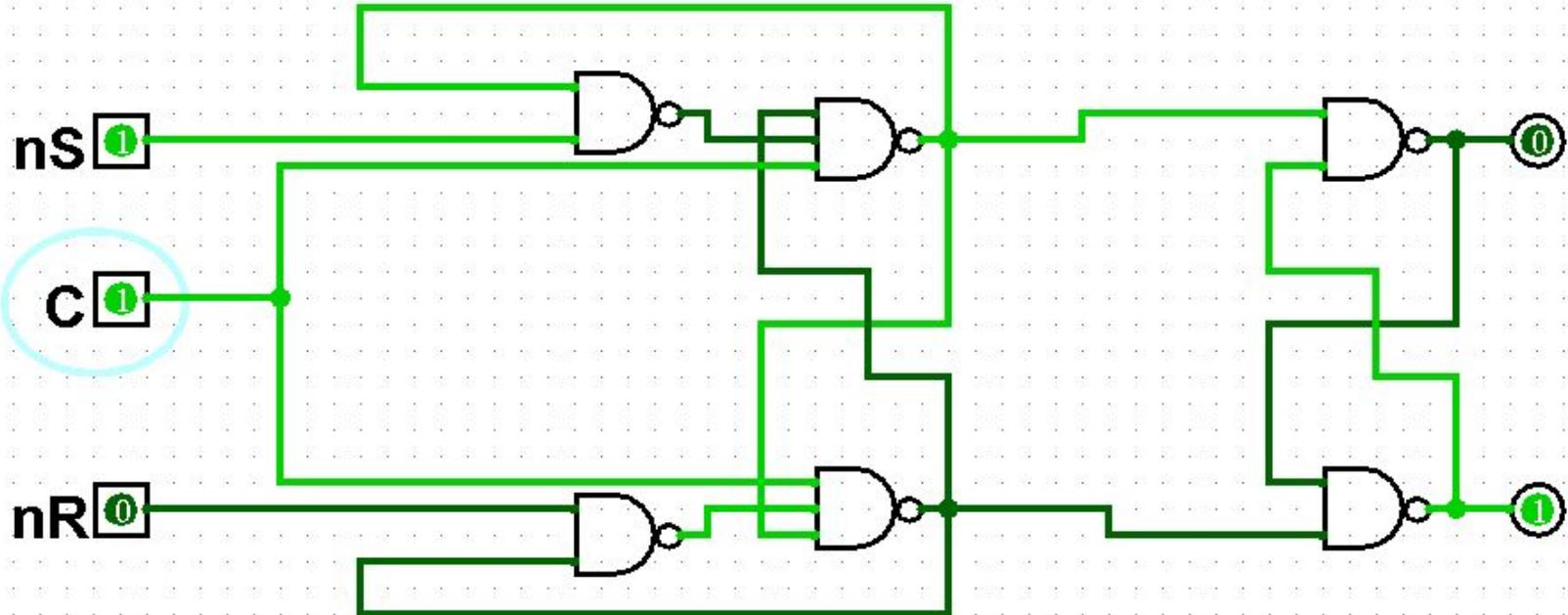
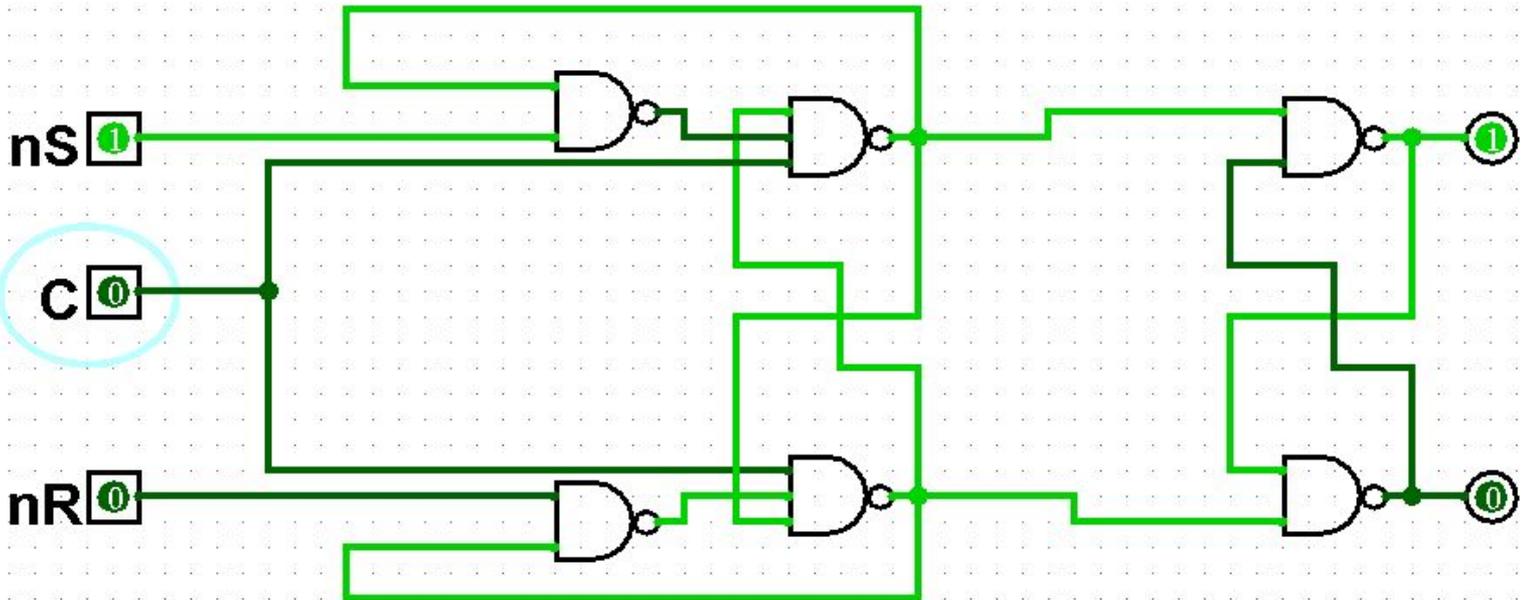
# Синхронный RS-триггер с динамическим управлением



# Синхронный RS-триггер с динамическим управлением







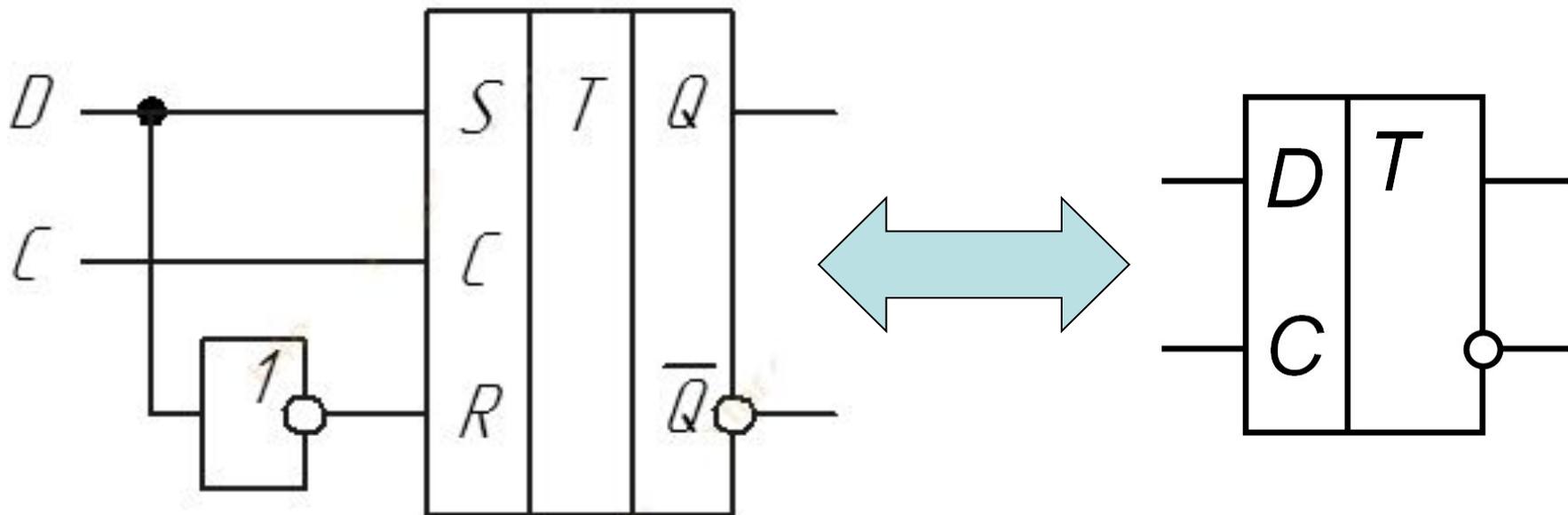
# D-триггер

Триггер-задержка – хранит предыдущее состояние до прихода очередного синхроимпульса.

C	D	$Q_i$	$\overline{Q_i}$
0	0	$Q_{i-1}$	$\overline{Q_{i-1}}$
0	1	$Q_{i-1}$	$\overline{Q_{i-1}}$
1	0	0	1
1	1	1	0

Можно построить на RS-триггере, если подавать сигнал D на линию S напрямую, а на R – через инверсию.

# D-триггер



# T-триггер

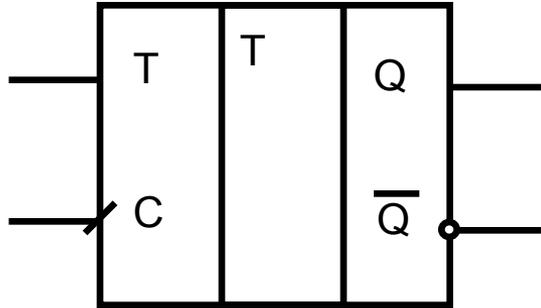
Триггер-счетчик – с приходом очередного счетного импульса меняет свое состояние на противоположенное.

Таблица истинности для синхронного T-триггера

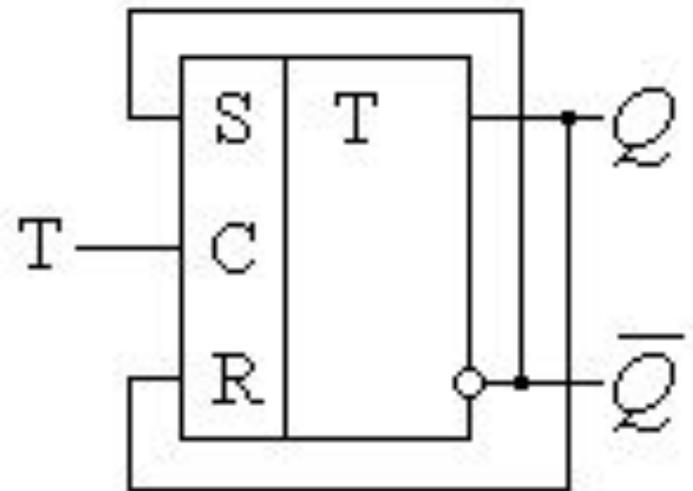
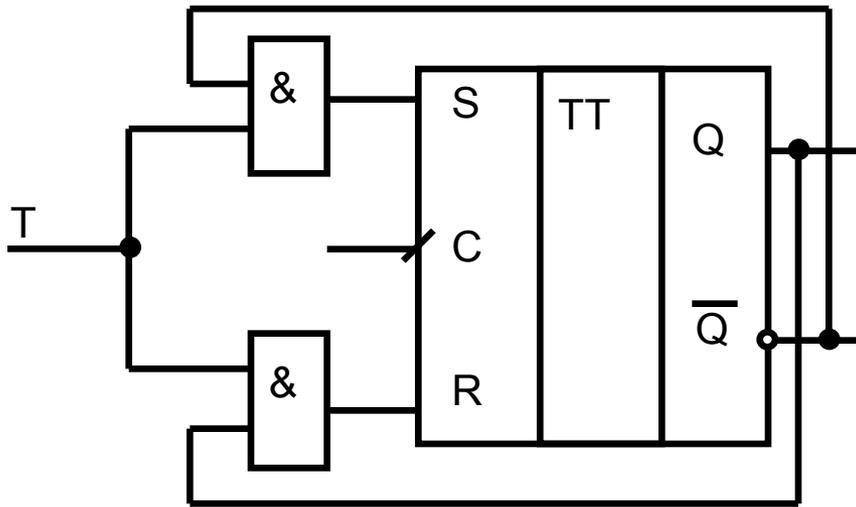
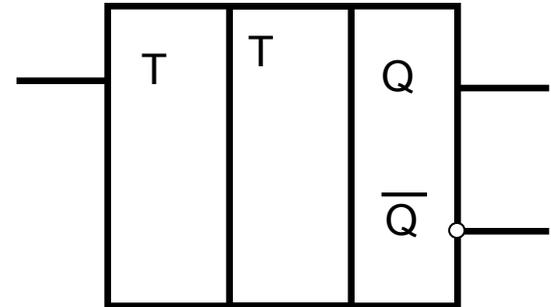
C	T	$Q_i$	$\overline{Q}_i$
0	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
0	1	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	0	$Q_{i-1}$	$\overline{Q}_{i-1}$
1	1	$\overline{Q}_{i-1}$	$Q_{i-1}$

# T-триггер

Синхронный



Асинхронный



# JK-триггер

Универсальный триггер – отличается от RS-триггера тем, что не имеет запрещенных состояний.

J – аналогичен сигналу S (переводит триггер в единицу).

K – аналогичен сигналу R (переводит триггер в ноль).

При J и K одновременно равных единице JK-триггер работает как T-триггер, меняя свое состояние на противоположное.

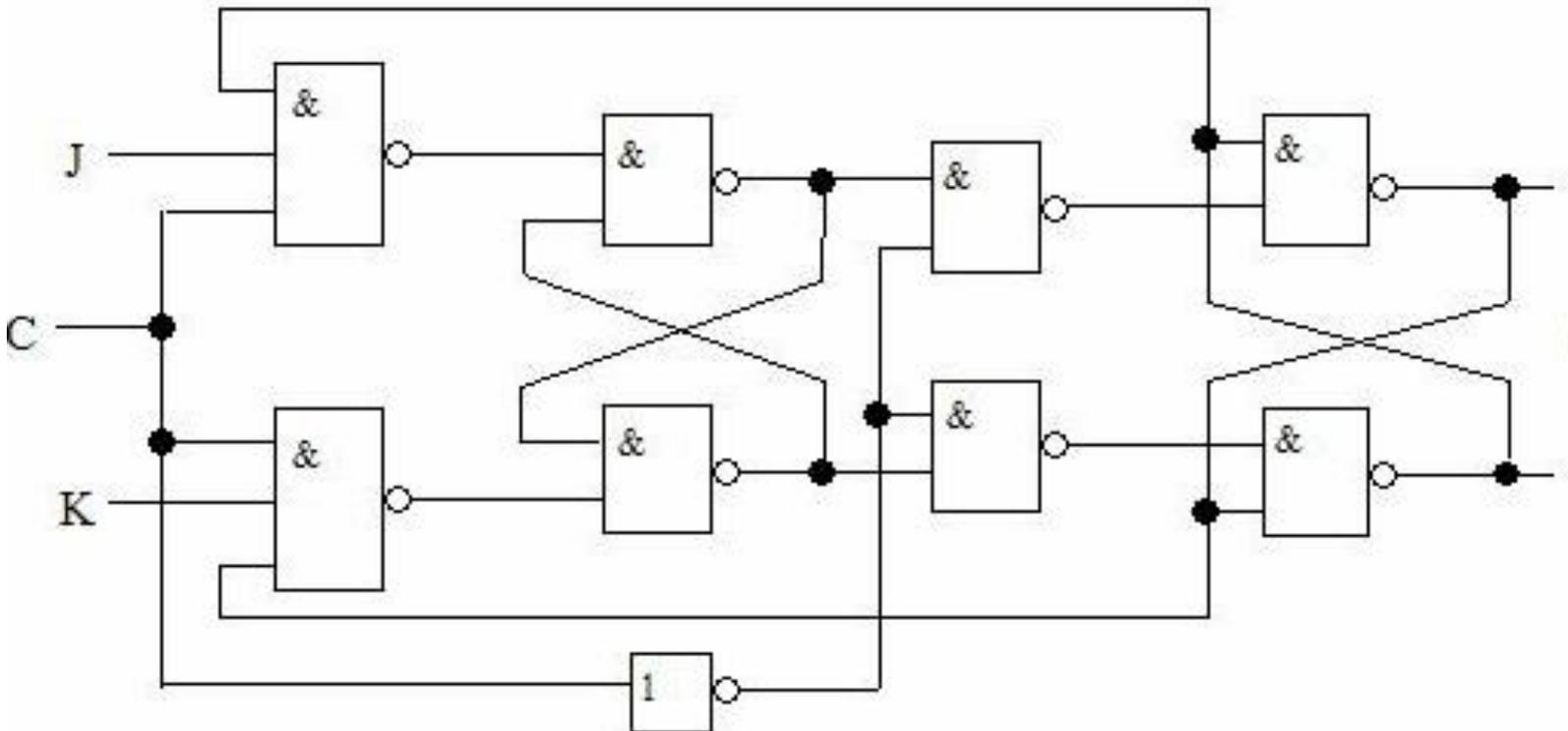
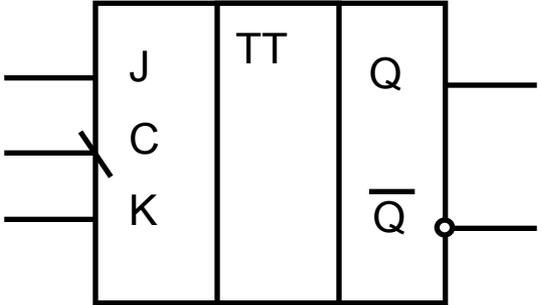
# Синхронный двухступенчатый JK-триггер

C	J	K	$\overline{Q}_i$	$Q_i$
1	0	0	$\overline{Q}_{i-1}$	$Q_{i-1}$
1	0	1	0	1
1	1	0	1	0
1	1	1	$Q_{i-1}$	$\overline{Q}_{i-1}$
0	x	x	$\overline{Q}_{i-1}$	$Q_{i-1}$

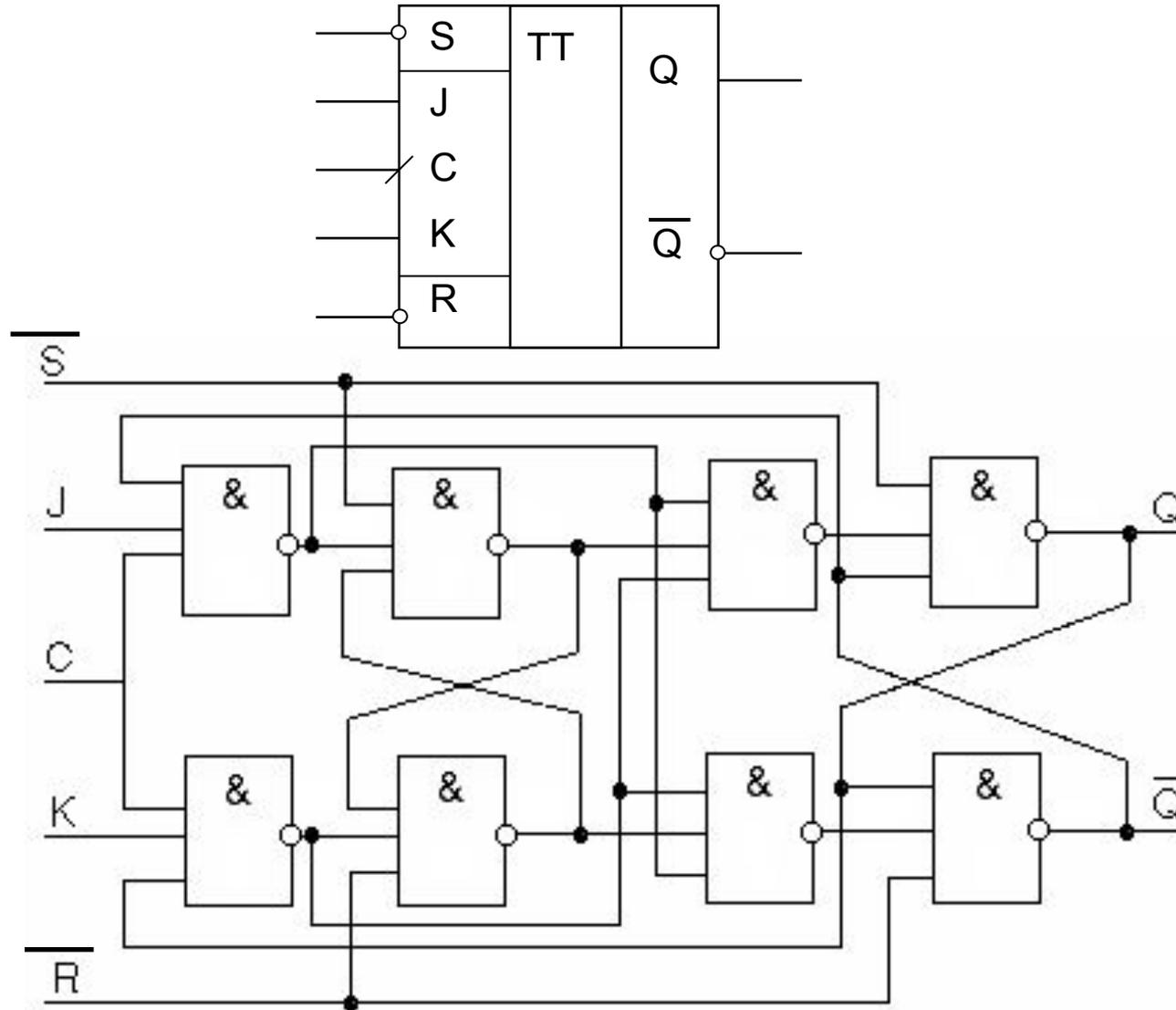
K – сброс.

J – установка.

# Синхронный двухступенчатый JK-триггер

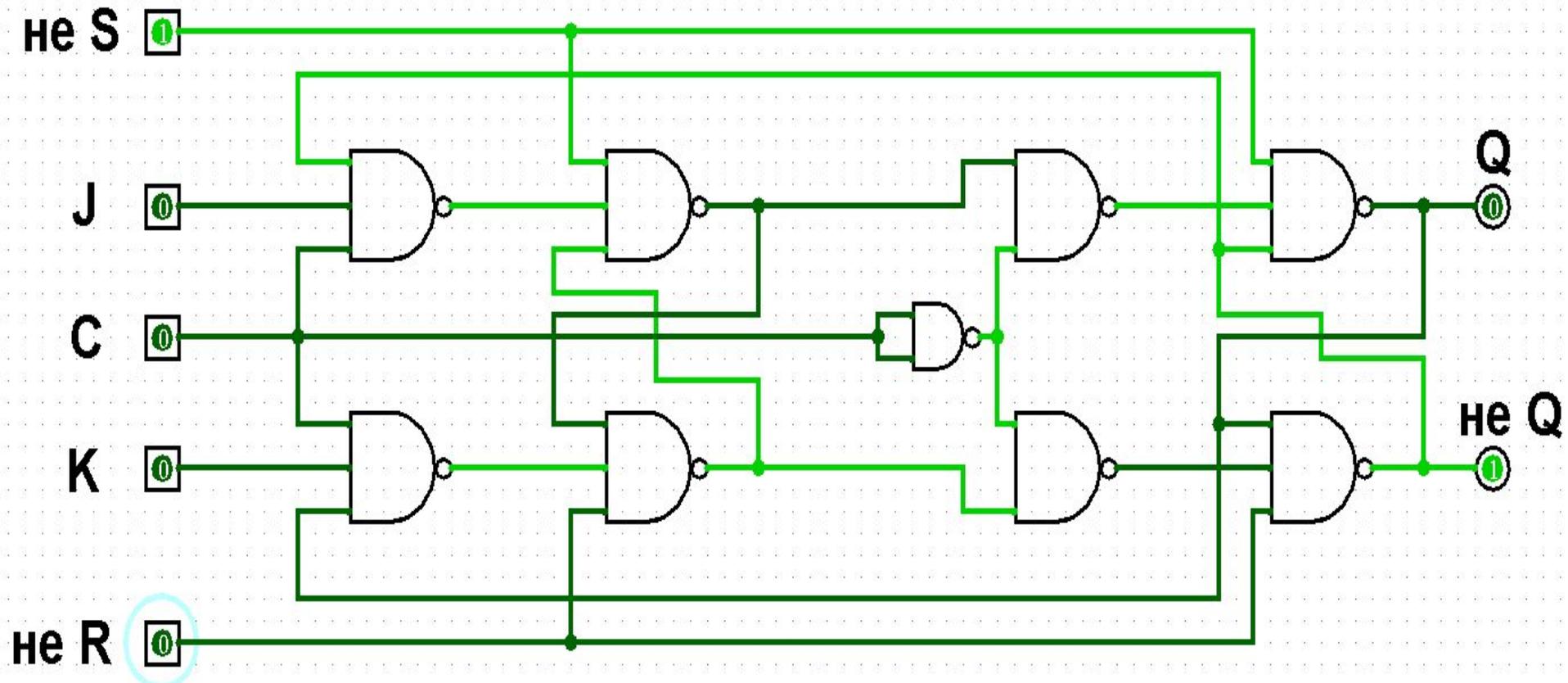


# Синхронный двухступенчатый JK-триггер с асинхронными входами

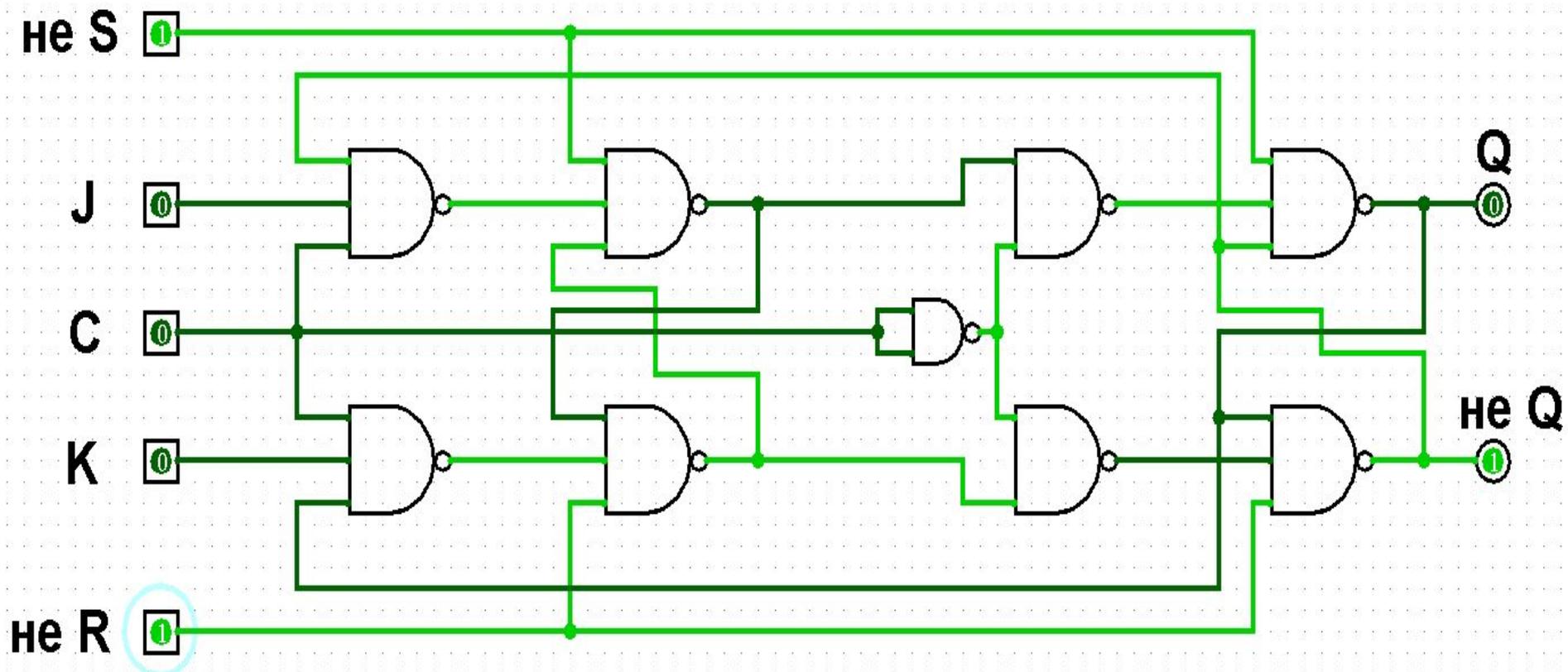




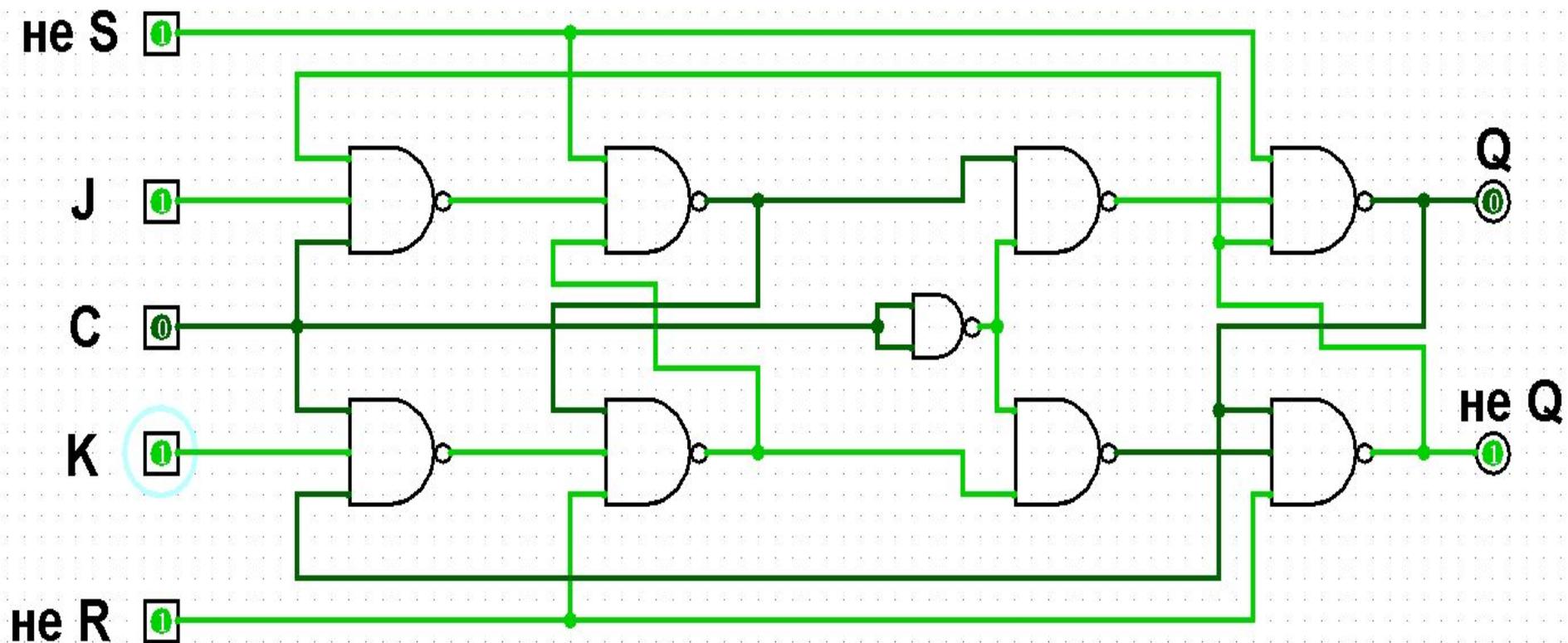
# Сбросим триггер в 0



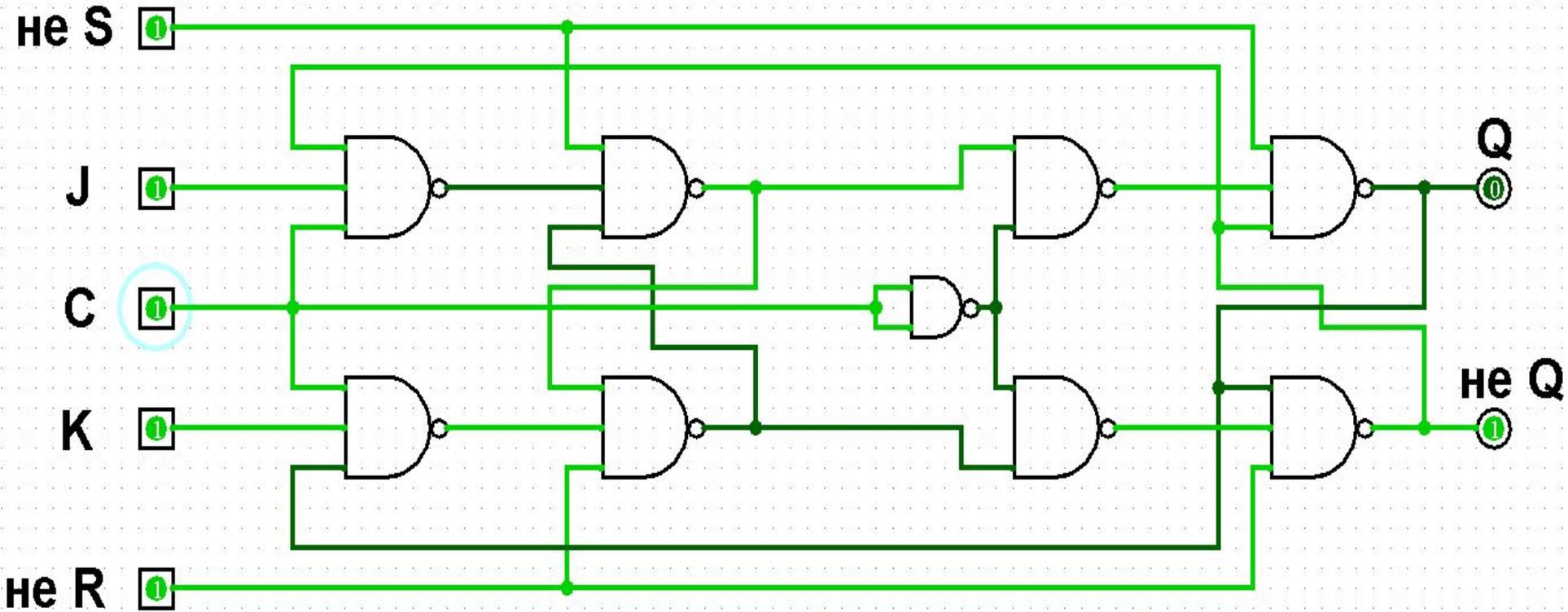
# Восстановим НЕ R = 1



# Подадим $J = 1$ и $K = 1$



# Подадим синхросигнал $C = 1$



**ПереклЮчилась  
первая ступень**



**Тема:**  
**Счетчики**

**Счетчиком** называется цифровой автомат, предназначенный для счета входных импульсов хранения их количества в виде двоичного числа.

Параметры счетчика:

- **Модуль счета ( $M$ )** - максимальное количество единичных импульсов, которое может быть сосчитано счетчиком. Счетчик обнуляется, когда приходит  $M$ -ый импульс.

- **Шаг счета** – приращение значения счетчика при приходе очередного импульса.

- **Направление счета** – в сторону увеличения значений, или в сторону уменьшения значений.

## **По модулю счета:**

- двоичные;
- двоично-десятичные;
- с произвольным постоянным модулем счёта;
- с переменным модулем счёта.

## **Шаг счета:**

- 1;
- 2;
- ...
- $K < M$

## **По направлению счета:**

- суммирующие;
- вычитающие;
- реверсивные.

*По способу организации межразрядных связей* счетчики делятся на:

- *счетчики с последовательным переносом (асинхронные счетчики)*, в которых переключение триггеров разрядных схем осуществляется

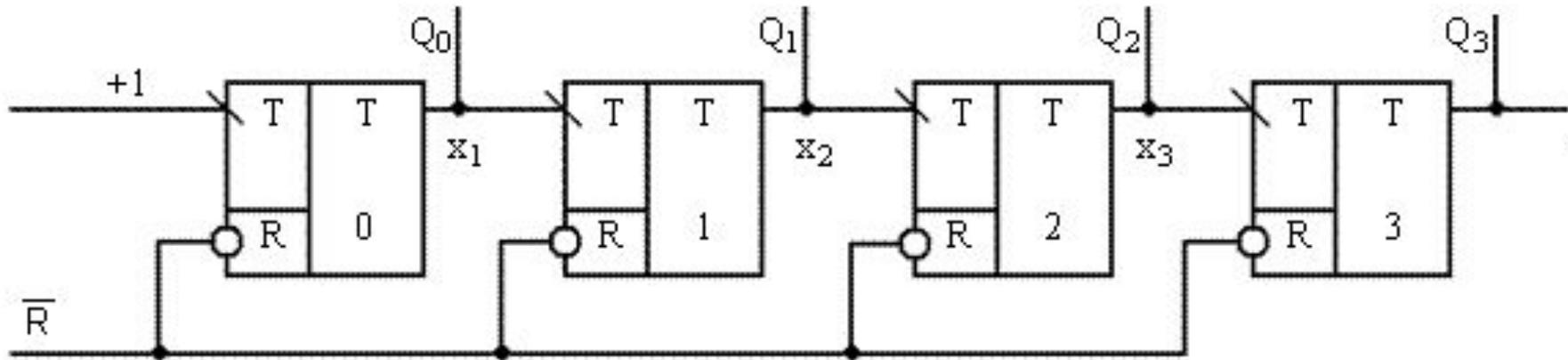
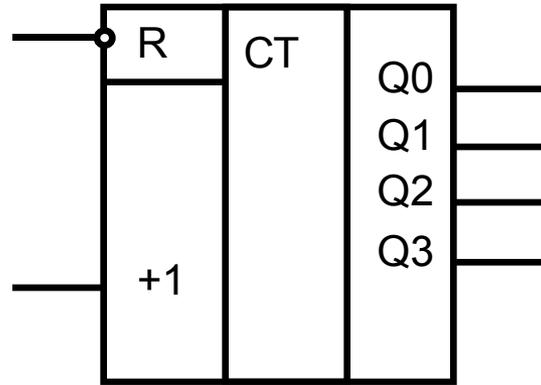
последовательно один за другим;

- *счетчики с параллельным переносом (синхронные счетчики)*, в которых переключение всех триггеров разрядных схем осуществляется

одновременно по сигналу синхронизации;

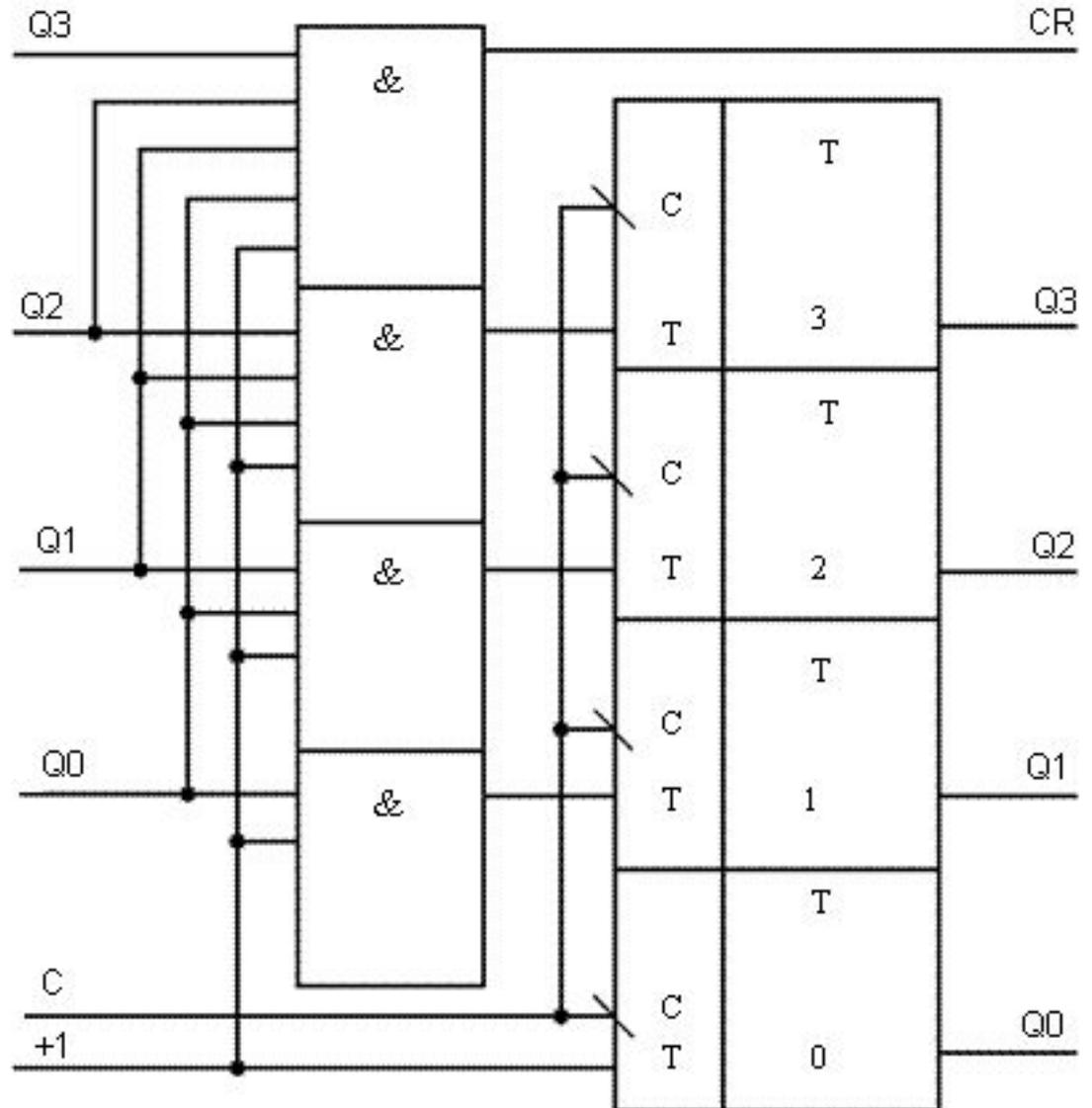
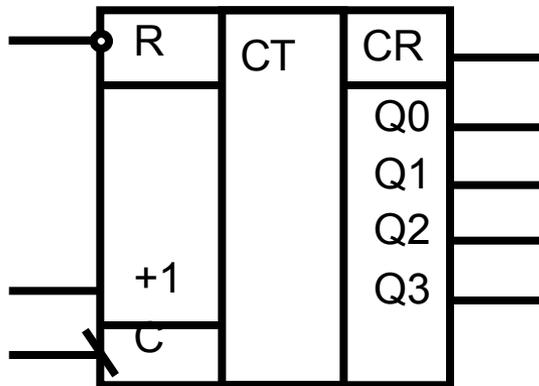
- *счетчики с комбинированным последовательно-параллельным переносом*, при котором используются различные комбинации способов переноса.

# Четырехразрядный суммирующий двоичный счетчик на Т-триггерах с последовательным переносом

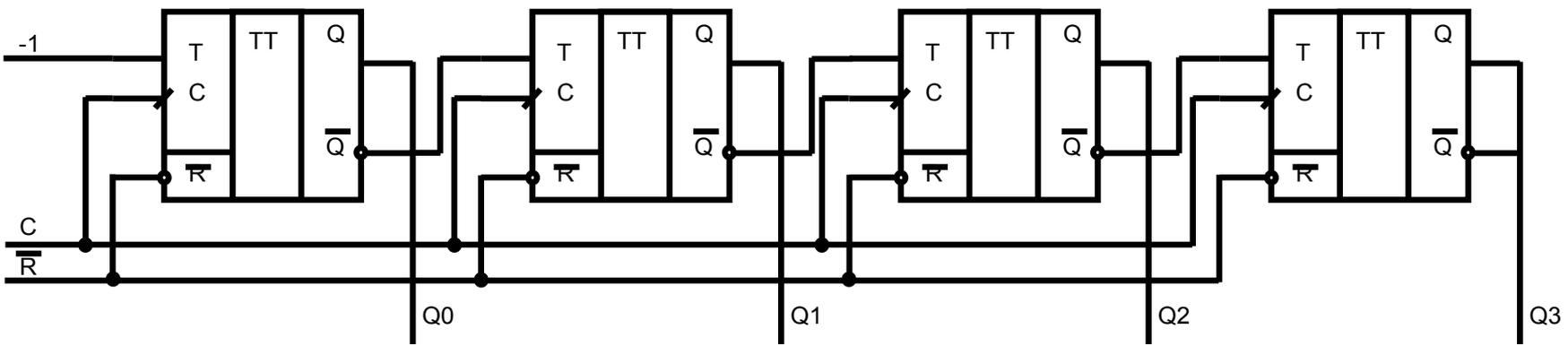
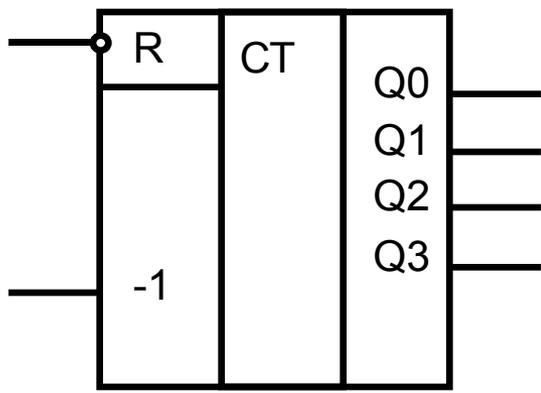




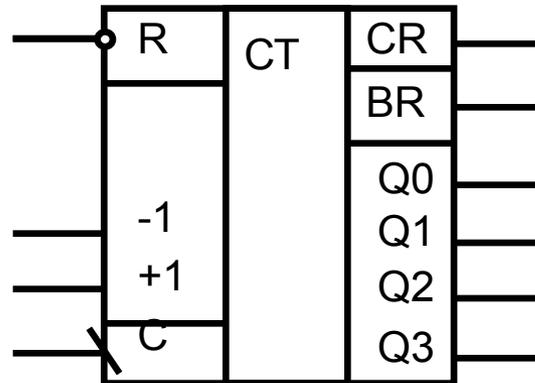
# Счетчик с ускоренным переносом

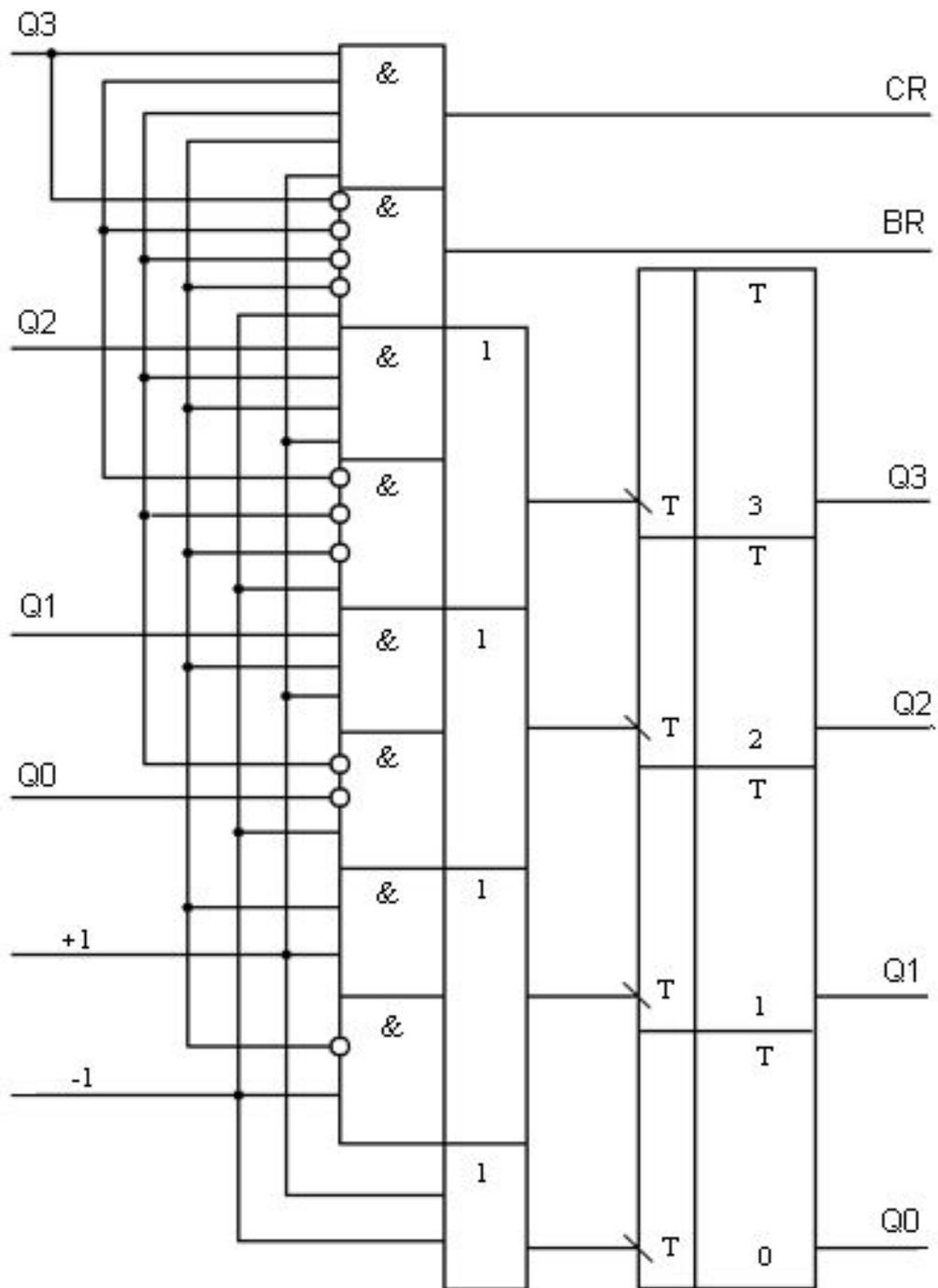


# Четырехразрядный вычитающий двоичный счетчик на Т-триггерах с последовательным переносом



# Четырехразрядный реверсивный двоичный счетчик на Т-триггерах с ускоренным переносом

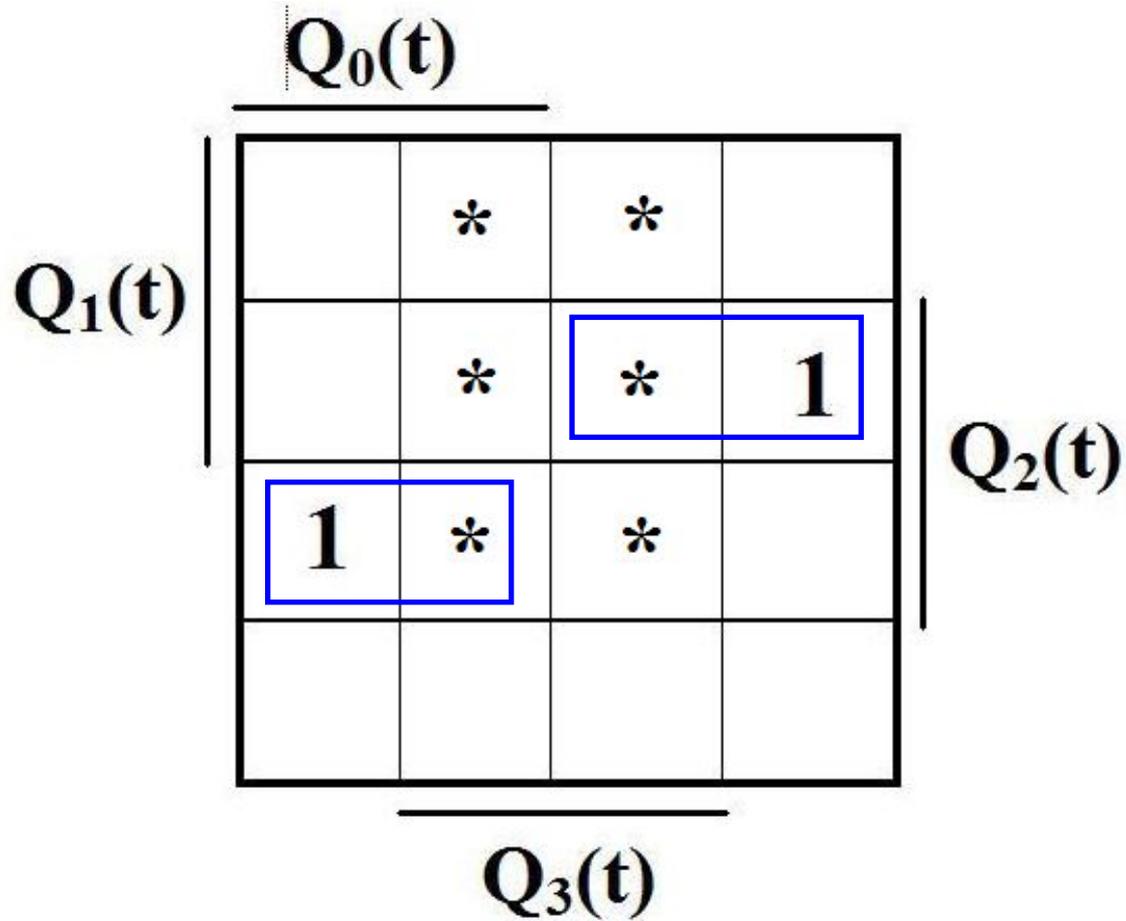




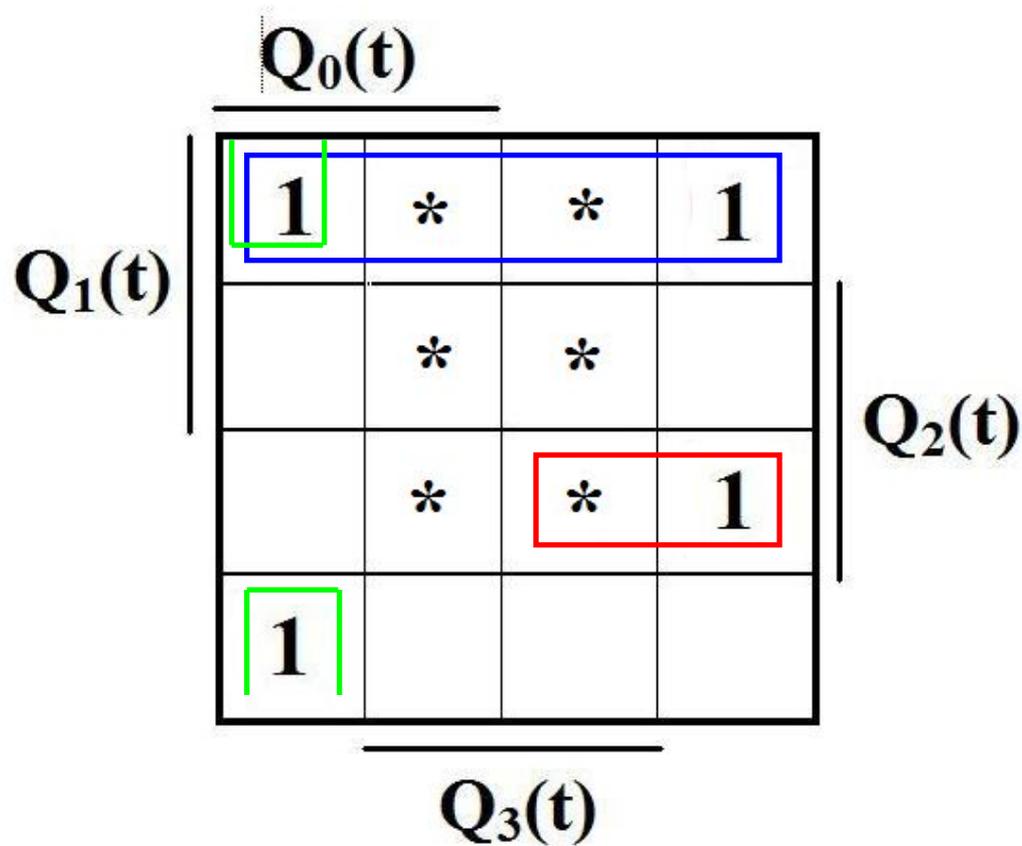
# **Синтез счетчиков**

# Построим суммирующий счетчик по модулю 10 с шагом 3

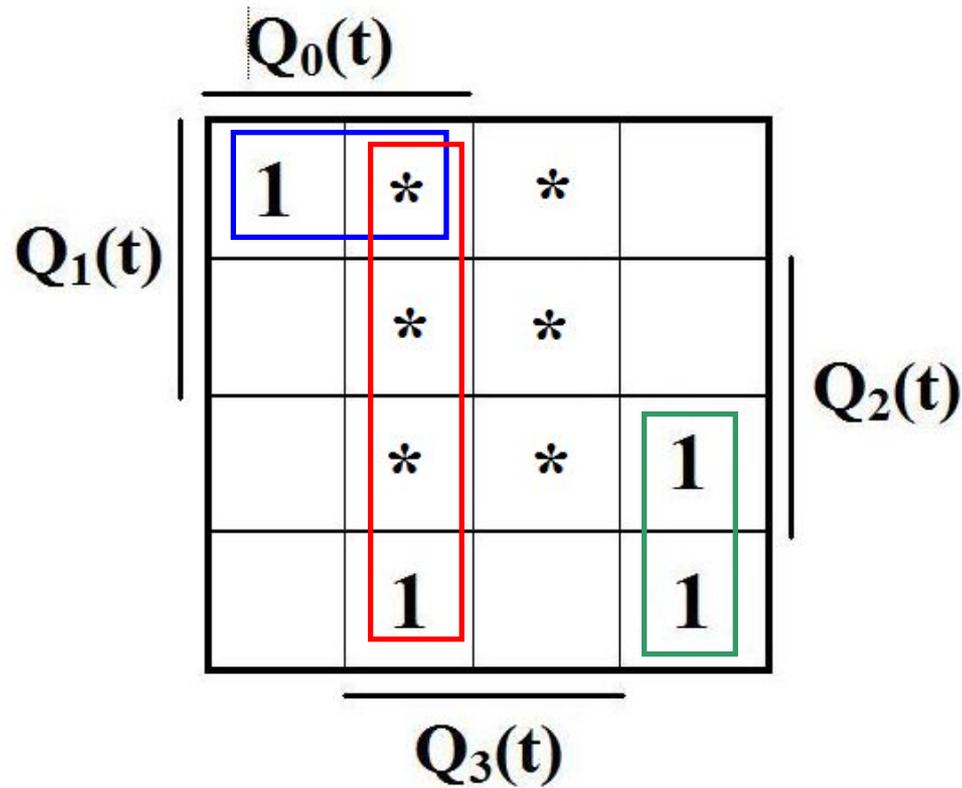
$Q_3(t)$	$Q_2(t)$	$Q_1(t)$	$Q_0(t)$	$Q_3(t+1)$	$Q_2(t+1)$	$Q_1(t+1)$	$Q_0(t+1)$
0	0	0	0	0	0	1	1
0	0	0	1	0	1	0	0
0	0	1	0	0	1	0	1
0	0	1	1	0	1	1	0
0	1	0	0	0	1	1	1
0	1	0	1	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	1	0	0	0	0
1	0	0	0	0	0	0	1
1	0	0	1	0	0	1	0
1	0	1	0	*	*	*	*
1	0	1	1	*	*	*	*
1	1	0	0	*	*	*	*
1	1	0	1	*	*	*	*
1	1	1	0	*	*	*	*
1	1	1	1	*	*	*	*



$$Q_3(t+1) = Q_2(t) \& \overline{Q_1(t)} \& Q_0(t) + \\ + Q_2(t) \& Q_1(t) \& \overline{Q_0(t)}$$



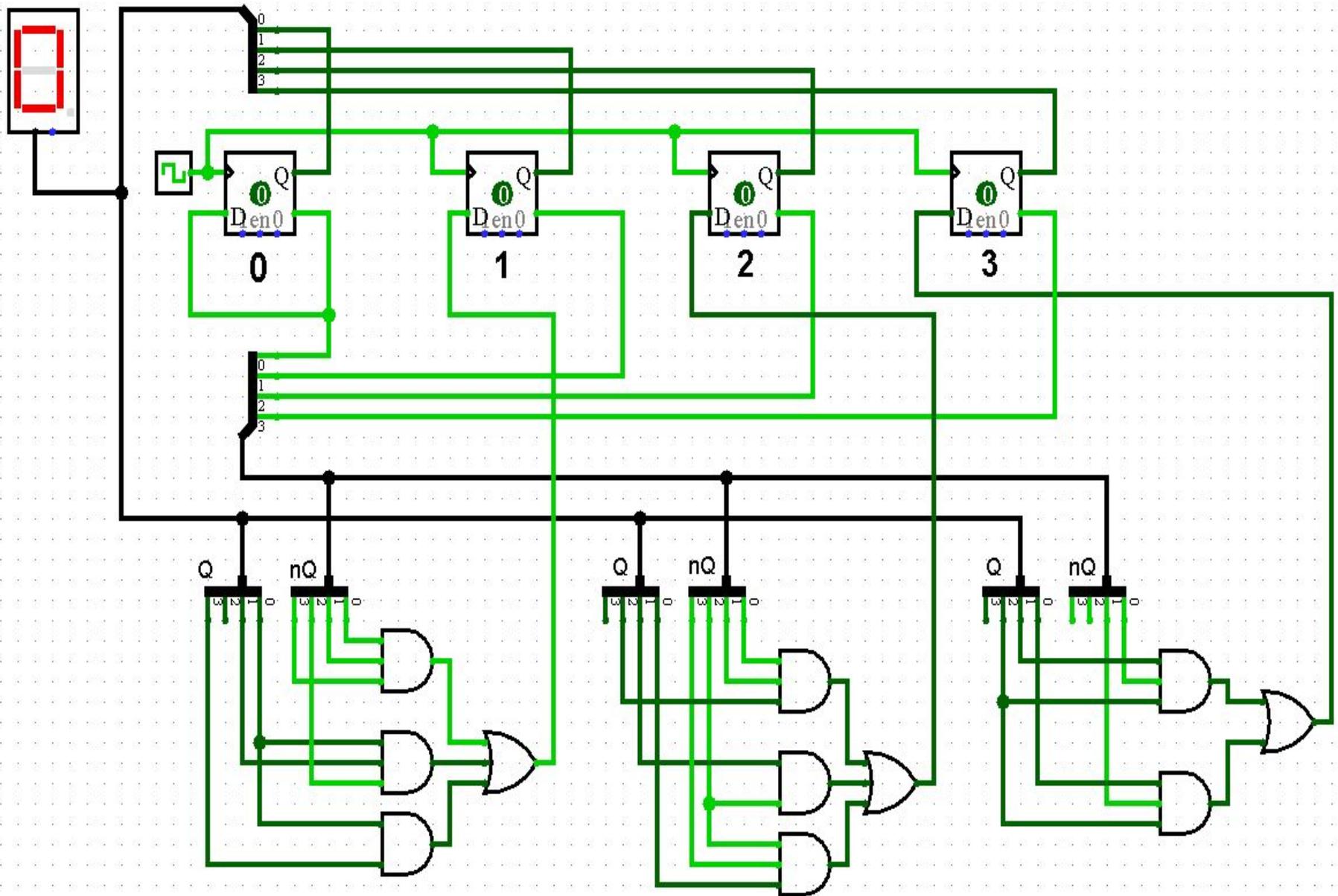
$$Q_2(t+1) = \overline{Q_2(t)} \& Q_1(t) + \overline{Q_3(t)} \& \overline{Q_2(t)} \& Q_0(t) + \\ + Q_2(t) \& \overline{Q_1(t)} \& \overline{Q_0(t)}$$

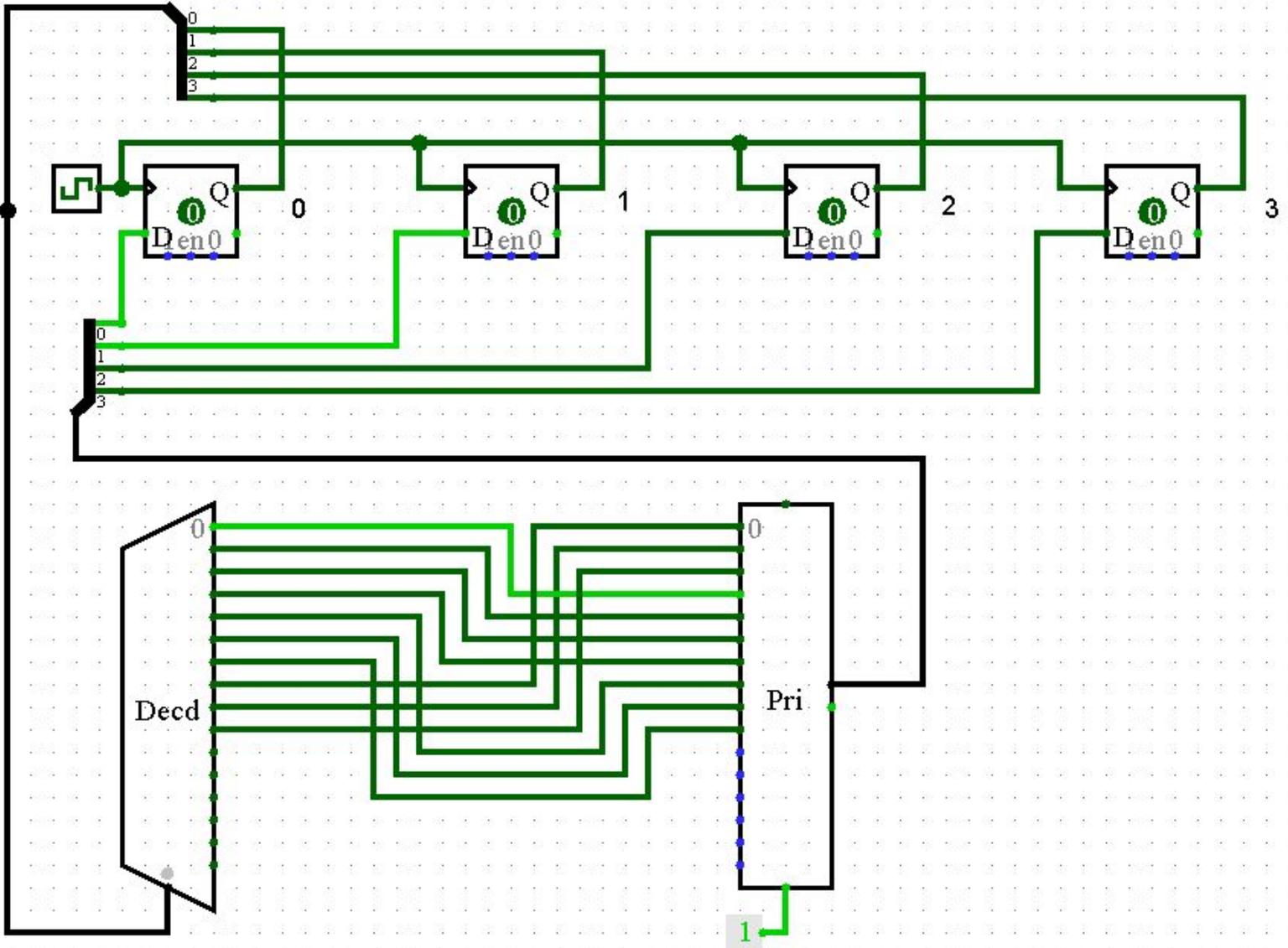
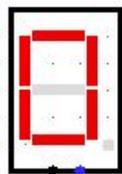


$$Q_1(t+1) = \overline{Q_3(t)} \& \overline{Q_1(t)} \& \overline{Q_0(t)} + \overline{Q_2(t)} \& Q_1(t) \& Q_0(t) + Q_3(t) \& Q_0(t)$$

$Q_0(t)$	$Q_0(t+1)$
0	1
1	0
0	1
1	0
0	1
1	0
0	1
1	0
0	1
1	0

$$Q_0(t+1) = \overline{Q_0(t)}$$





Спасибо за внимание !