

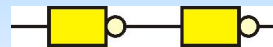


Последовательностная логика  
RS триггер

Sequential Logic  
SR Latch

# Что можно сделать из двух инверторов?

В рамках комбинаторной логики единственный вариант:



Без учета задержки только тривиальная функция

$$\textit{Output} = \textit{Input}$$

# Схема с обратной связью

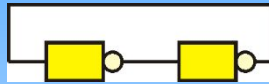
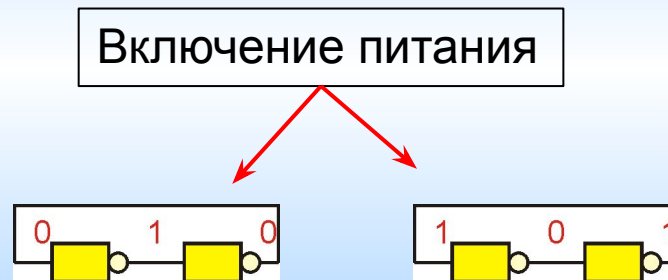
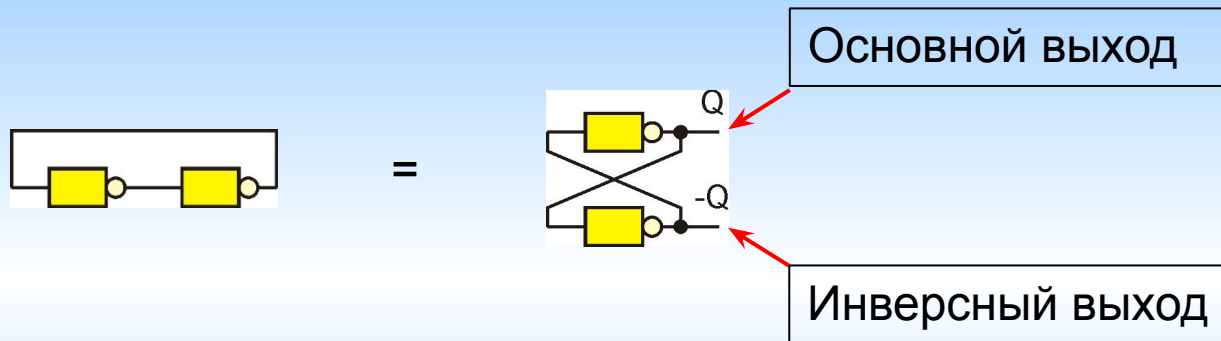


Схема управляет сама собой.

Возможны 2 устойчивых состояния



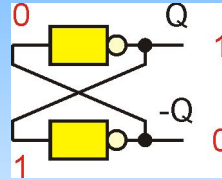
## Другое начертание схемы



Как управлять такой схемой?

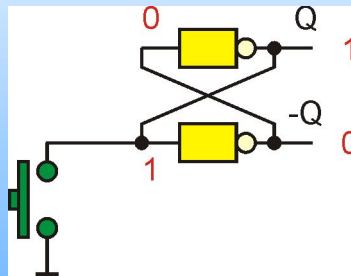
# Управление триггером

Допустим после включения питания триггер встал в положение 1  
(положение определяется по основному выходу):

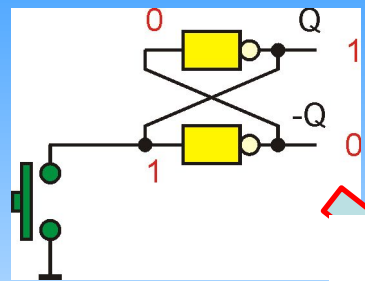


Требуется изменить состояние.

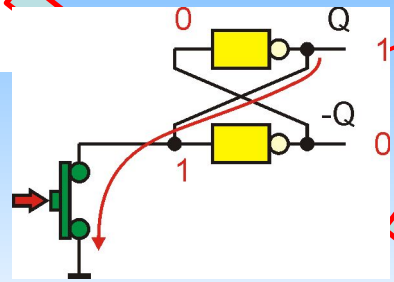
Можно подключить кнопку.



# Большая кнопка побеждает маленькую микросхему

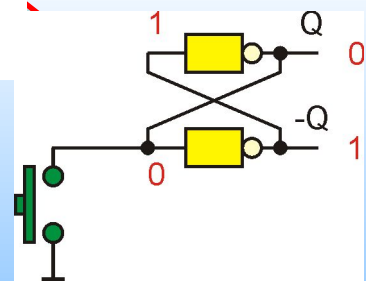
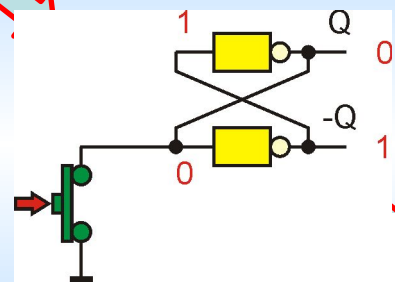


Начальное состояние



Ток короткого замыкания

Несколько наносекунд.  
Сгореть не успеет.



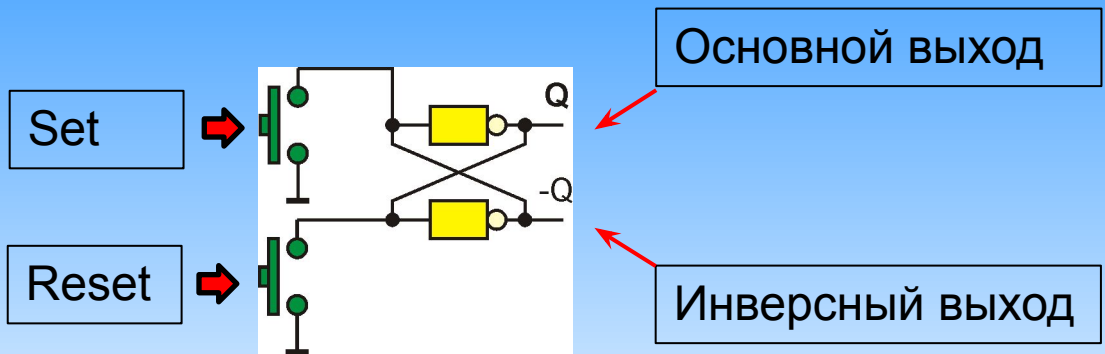
Большая кнопка побеждает маленькую микросхему



Новое состояние сохраняется и после отпущания кнопки. Это функция памяти.

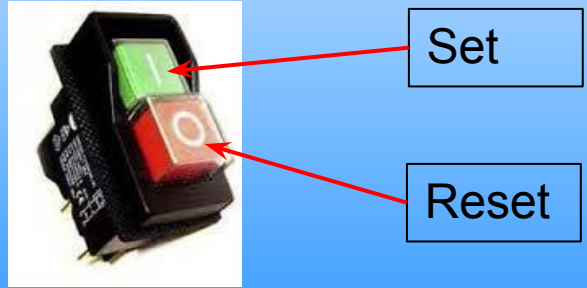
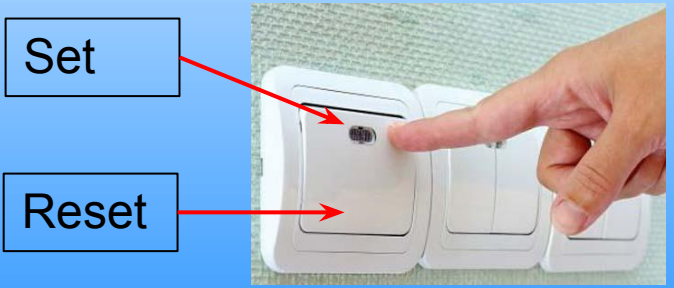
Дальнейшие нажатия кнопки не повлияют на состояние триггера

# Входы управления: Reset, Set.

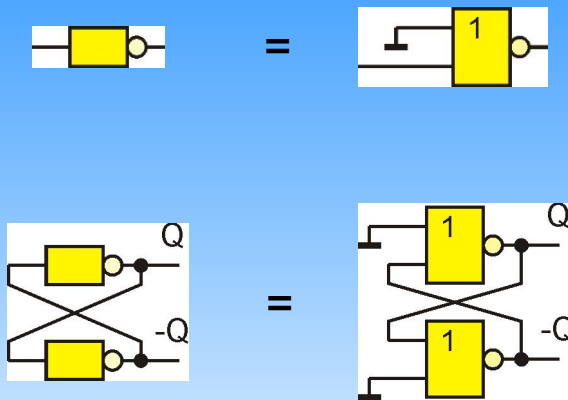


Кнопки		Выходы		Операция
Reset	Set	Q	-Q	
		$Q_{n-1}$	$-Q_{n-1}$	Хранение
	Нажатие	1	0	Установка
Нажатие		0	1	Сброс
Нажатие	Нажатие	КЗ	КЗ	Запрещенная комбинация, приводящая к разрушению инверторов

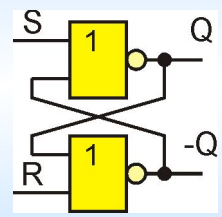
## Аналоги



# RS триггер на элементах ИЛИ-НЕ



Схема



Обозначение



Таблица истинности

Входы		Выходы		Операция
Reset	Set	Q	-Q	
0	0	$Q_{n-1}$	$-Q_{n-1}$	Хранение
0	1	1	0	Установка
1	0	0	1	Сброс
1	1	0	0	Нежелательная комбинация



# Нежелательное состояние. Почему?

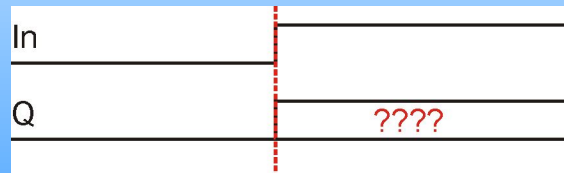
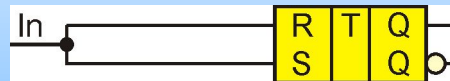
Таблица истинности

Входы		Выходы		Операция
Reset	Set	Q	-Q	
0	0	$Q_{n-1}$	$-Q_{n-1}$	Хранение
0	1	1	0	Установка
1	0	0	1	Сброс
1	1	0	0	Нежелательная комбинация



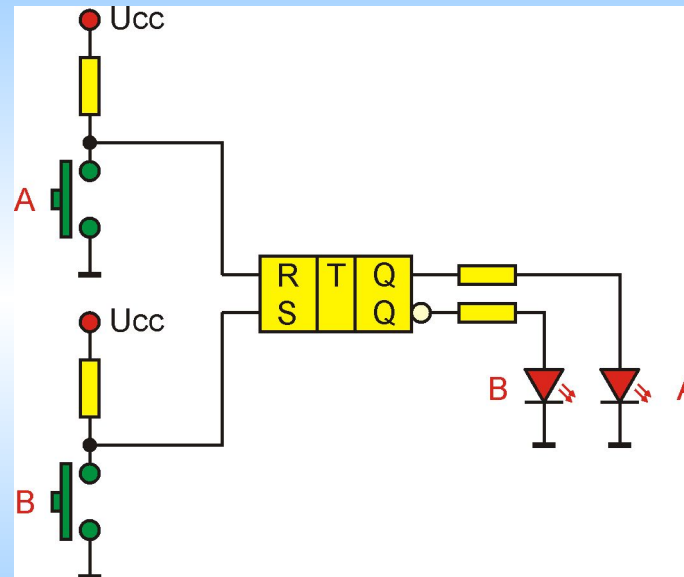
При такой комбинации ничего не сгорит,  
НО

Переход из  $R=1, S=1$  в  $R=0, S=0$  приведет к неопределенному состоянию триггера.



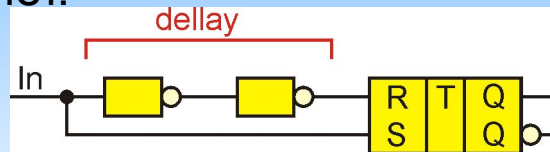
Гонки

# Гонки. Игра «кто быстрее?»

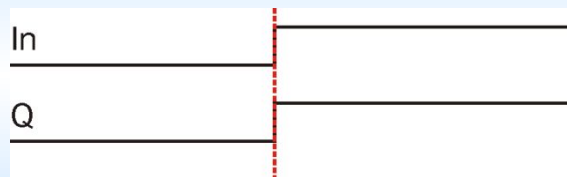


Если контролировать порядок снятия сигналов Set и Reset то использовать нежелательную комбинацию входных сигналов вполне можно.

На следующей схеме неопределенности нет.

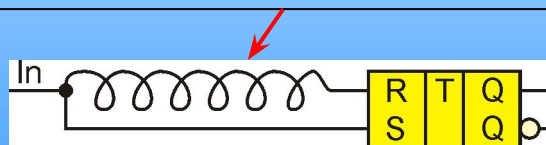


Сигнал Set обгоняет сигнал Reset

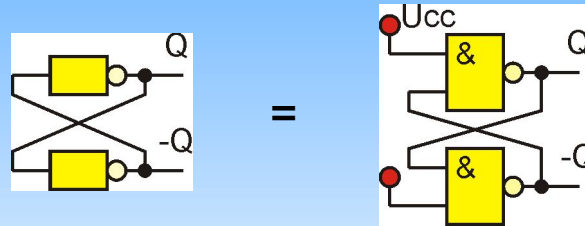


Играют роль не только задержки в логических элементах, но и в соединительных линиях.

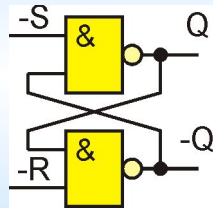
Электрическая длина этого проводника больше.



# RS триггер на элементах И-НЕ



Схема



Обозначение



Таблица истинности

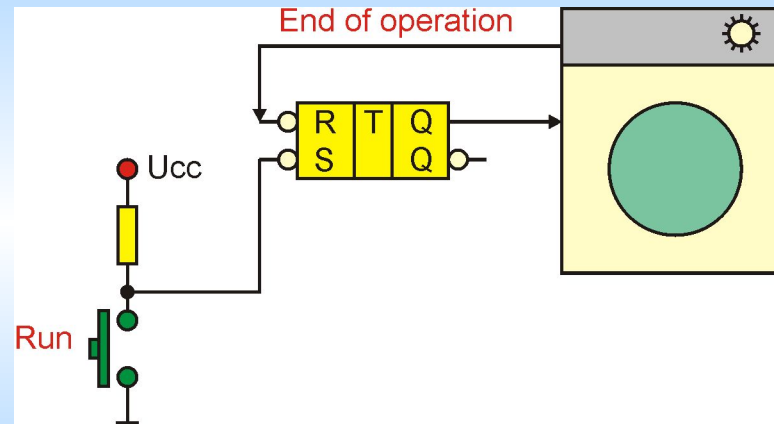
Входы		Выходы		Операция
$\bar{R}$	$\bar{S}$	$Q$	$\bar{Q}$	
1	1	$Q_{n-1}$	$\bar{Q}_{n-1}$	Хранение
1	0	1	0	Установка
0	1	0	1	Сброс
0	0	0	0	Нежелательная комбинация

# Применение RS триггеров 1

RS триггер является основой для построения всех других типов триггеров

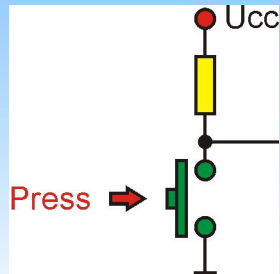
## Применение RS триггеров 2

Включение из одного места, а выключение из другого.

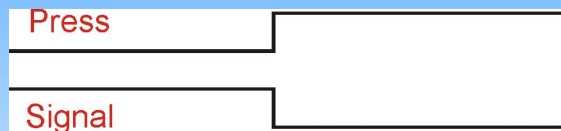


# Применение RS триггеров 3

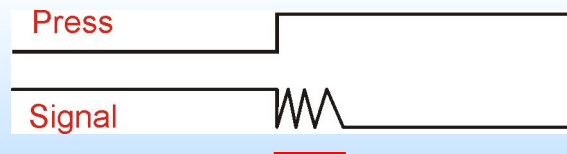
## Дребезг контактов Switch Bounce



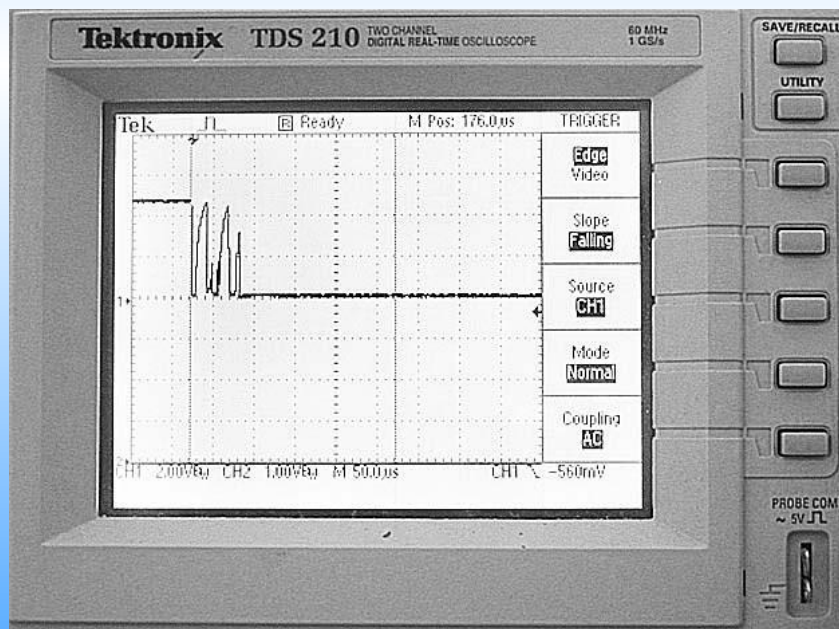
Идеальный случай



В реальности так:



10÷100 мс



# Применение RS триггеров 3

## Устранение дребезга контактов Switch Debouncing

