

ЛАБОРАТОРНАЯ РАБОТА №1	Группа 39	2022
Построение логических схем в среде моделирования	ЯКОВЛЕВ ИЛЬЯ ИГОРЕВИЧ	

**Цель работы:** моделирование логических схем на элементах с памятью.

**Инструментарий и требования к работе:** работа выполняется в среде моделирования Logisim evolution.

### **Описание**

Составить и описать принцип работы двух схем: счётчика и регистра сдвига с линейной обратной связью

### **Вариант**

Вариант №125.

Схема счетчика: Синхронный суммирующий счетчик

Модуль счета: 12

Тип конфигурации регистра: Галуа

Конфигурация регистра: (20, 17)

### **Синхронный суммирующий счетчик**

Счетчиком называют цифровое устройство, предназначенное для подсчета входных сигналов (импульсов).

Суммирующий счетчик предназначен для выполнения счета символов в прямом направлении (т е для сложения).

Синхронный счетчик — счетчик, в котором триггеры переходят в новое состояние одновременно (синхронно).

## Описание вспомогательных логических схем, используемых для построения синхронного суммирующего счетчика

Триггер — класс электронных устройств, обладающих способностью длительно находиться в одном из двух устойчивых состояний и чередовать их под воздействием внешних сигналов

Триггер называется синхронизированным, если есть вход синхронизации, причем триггер может менять внутреннее состояние только при 1 на этом входе.

### Синхронизированный RS-trigger

Данный триггер является базовым, из него будут строится другие триггеры.

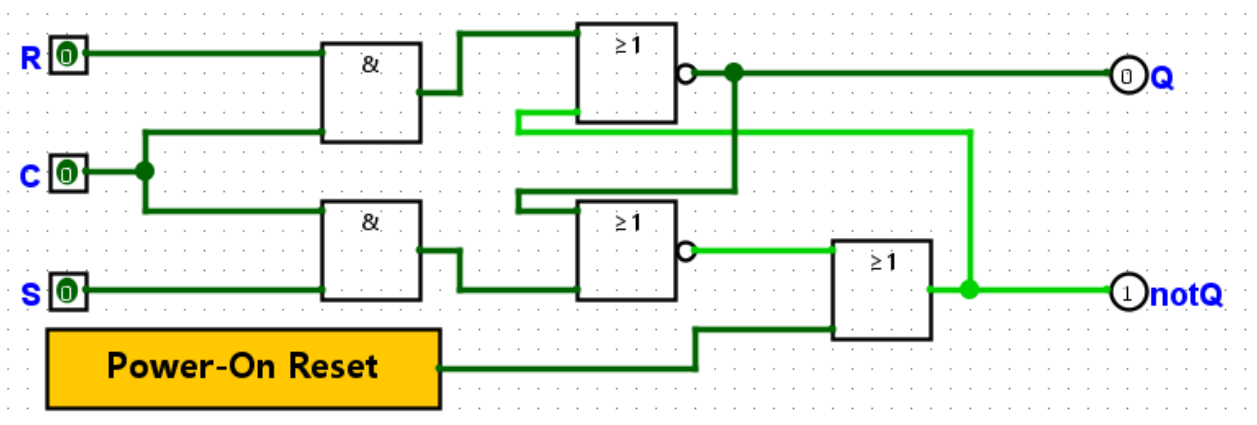
У него есть 3 входа: R, S, C (C – вход синхронизации), 2 выхода: Q, notQ ( $Q = \text{!notQ}$ )

Таблица истинности:

C	R	S	Q	notQ
1	0	0	safelastQ	safelastnotQ
1	0	1	1	0
1	1	0	0	1
1	1	1	undefined behavior	

При  $C = 0$  внутреннее состояние триггера не меняется.

Ниже представлена схема RS-trigger:



POR нужен для начальной инициализации триггера  $Q = 0$ ,  $\text{notQ} = 1$ .

## D-trigger

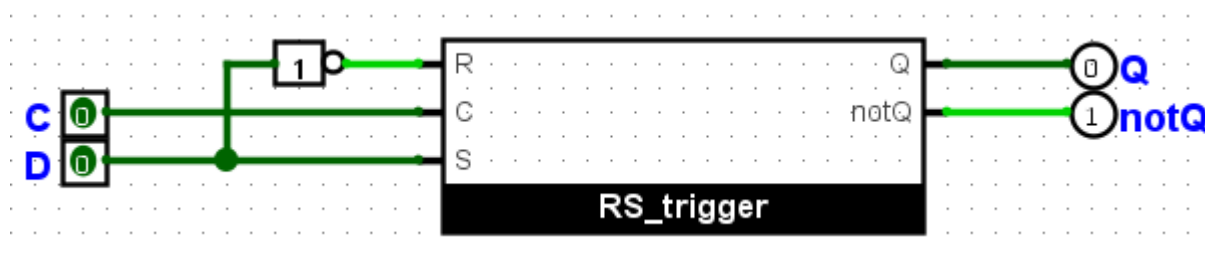
У него есть 2 входа: D, C (C – вход синхронизации), 2 выхода: Q, notQ ( $Q = \neg \text{notQ}$ )

Данный триггер является небольшой модификацией RS триггера, его таблица истинности выглядит так:

C	D	Q	notQ
1	0	0	1
1	1	1	0

При  $C = 0$  внутреннее состояние триггера не меняется.

Ниже представлена схема D-trigger (с использованием RS):



## JK-trigger

У него есть 3 входа: J, K, C (C – вход синхронизации), 2 выхода: Q, notQ ( $Q = \neg \text{notQ}$ )

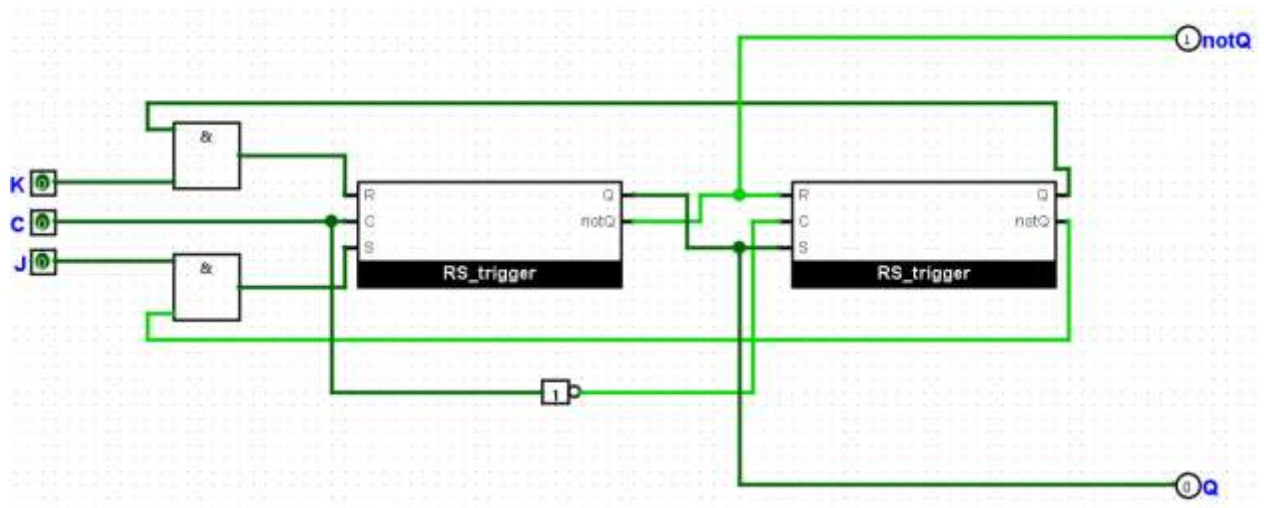
Данный триггер является модификацией RS триггера, но с определенным состоянием  $J = 1$ ,  $K = 1$  (в RS состояние  $R = 0$  и  $S = 0$  не определено), как инверсия сохраненного значения в Q и notQ.

Таблица истинности:

C	J	K	Q	notQ
1	0	0	safelastQ	safelastnotQ
1	0	1	0	1
1	1	0	1	0
1	1	1	inv	inv

При  $C = 0$  внутреннее состояние триггера не меняется.

Ниже представлена схема JK-trigger (с использованием RS):



Принцип работы заключается в том, что правый RS триггер хранит предыдущее сохраненное в JK триггер значение (т е то, которое было сохранено перед новым, актуальным сохраненным значением), а левый RS триггер хранит актуальное значение JK триггера.

Инвертирование происходит по значению правого триггера. Он(правый триггер) обновляется при  $C = 0$ , а левый при  $C = 1$ , за счет этого достигается единичное инвертирование.

### My-trigger

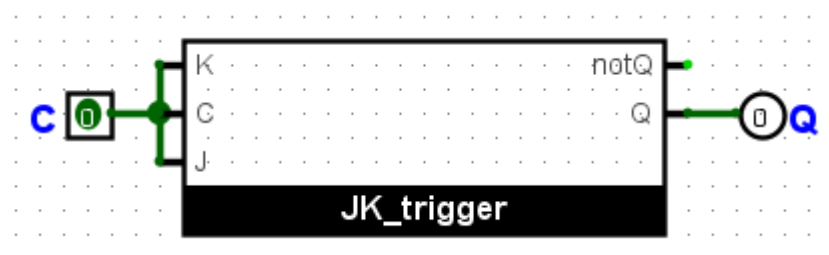
У него есть 1 входа: C (C – вход импульса, такта), 1 выход: Q.

Это моя модификация JK триггера, которая инвертирует свое ранее сохраненное значение Q при  $C = 1$ , при  $C = 0$  внутреннее состояние не меняется.

Таблица истинности:

C	Qold	Qnew
1	0	1
1	1	0

Ниже представлена схема my-trigger:



## Counter\_Basic\_El

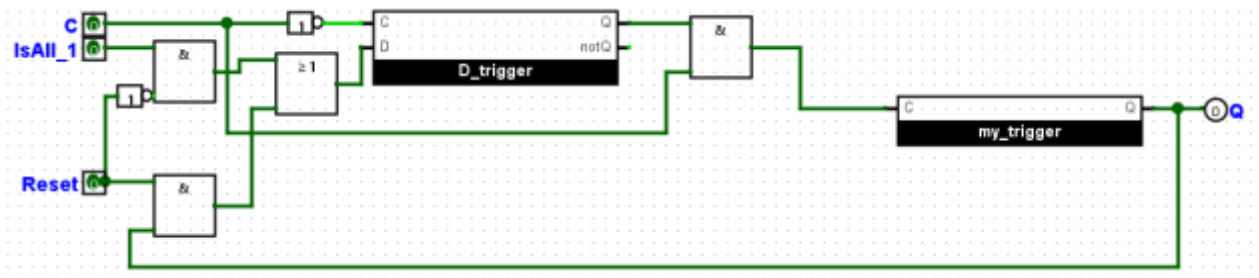
Это базовый элемент для построения синхронного суммирующего счетчика, отвечающий за хранение и увеличение разряда двоичного числа.

У него есть 3 входа: IsAll\_1, C, Reset (C – вход синхронизации), 1 выход: Q.

IsAll\_1 – Если все предыдущие разряды двоичного числа на предыдущем шаге 1, то IsAll\_1 = 1, иначе 0. Эта переменная отвечающая за перенос разряда.

Reset – Если Reset = 1, то внутреннее состояние элемента сбрасывается, а конкретно Q становится = 0.

Ниже представлена схема Counter\_Basic\_El:



Краткое описание принципа работы:

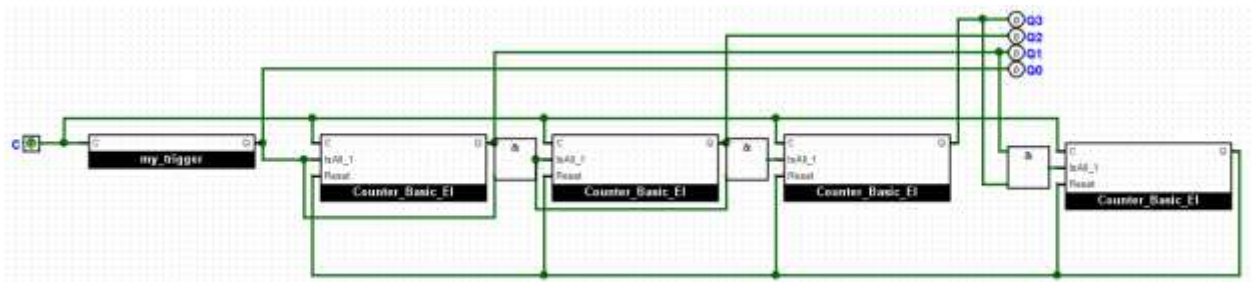
Происходит передача бита в my-trigger с задержкой на 1 такт, чтобы в my-trigger передавалось последнее переданное в IsAll\_1 при C = 0 значение. Это сделано, потому что при C = 0 значения на входах однозначно определены, а при C = 1 значения в процессе изменения, и может произойти такое, что за период, пока C = 1, на вход IsAll\_1 может быть подано 2 различных значения. Такая передача с задержкой реализована с помощью D-trigger.

При Reset = 0 элемент работает как указано выше. При Reset = 1 на вход D D-trigger подано Q. Значит, на следующем такте на вход my-trigger будет подано Q, что приведет Q в состояние 0.

## Counter

Это реализация синхронного суммирующего счетчика.

Ниже представлена схема Counter:



Краткое описание принципа работы:

C – тактовый вход

Изначально все триггеры содержат 0.

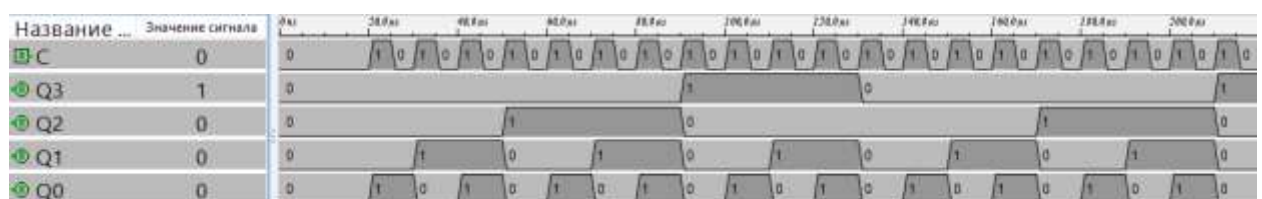
My-trigger изменяет свое значение каждый такт.

C помощью and вычисляю значение IsAll\_1 для каждого базового элемента. Соответственно первый базовый элемент изменяет свое значение каждый второй такт, второй базовый элемент каждый четвертый такт, третий базовый элемент каждый восьмой такт. Таким образом, они представляют из себя суммирующий счетчик.

Чтобы счет велся по модулю 12, использован еще один базовый элемент, в который при текущем значении 10 передается 1, при значении 11 на его выход передается 1 (т к у базового элемента задержка задержка в такт), а далее передается на Reset всех базовых элементов, что обнуляет их к на следующем, 12-ом, такте, что и является реализацией модуля 12. (my-trigger обнулять не нужно т е 12 – четное и на 12 такте его значение и так 0).

Счетчик является синхронным, ведь C подается всем элементам одновременно и все значения в базовых элементах вычисляются исходя из значений, поданных при C = 0, а, значит, элементы принимают новое значение при C = 1 независимо от других элементов.

Временная диаграмма counter:

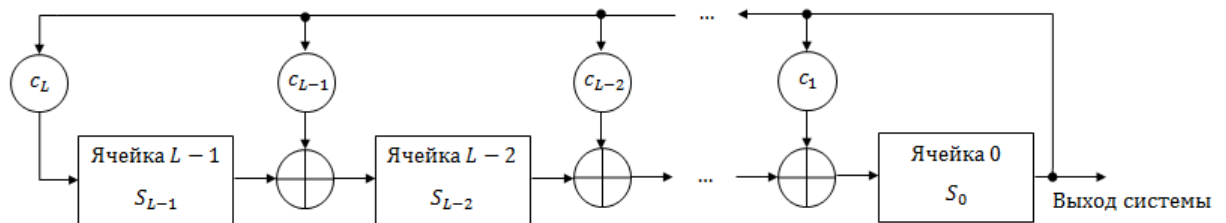


## Регистр сдвига с линейной обратной связью конфигурации Галуа

Регистр сдвига представляет собой упорядоченный набор триггеров (обычно D-триггеров), такой, что на каждом такте значения триггеров перемещаются в следующий с некоторым преобразованием.

Регистр сдвига с линейной обратной связью представляет собой регистр сдвига с определенной функцией, значение которой соответствует значению первого бита (в который не перемещается никакое значение другого триггера). Другими словами, доопределено преобразование для самого первого бита в цепочке.

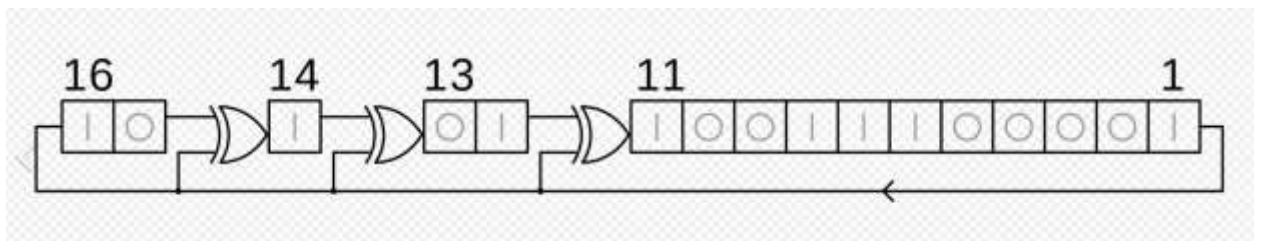
Конфигурация Галуа – определенный тип преобразований значений триггеров с использованием операции хог между некоторыми триггерами. Наглядное представление конфигурации Галуа:



Некоторые операции хог могут отсутствовать.

Примечание: ячейки (или триггеры) нумеруются справа налево от 0 до  $L - 1$ . Причем 0 соответствует выход системы.

Конфигурация регистра сдвига с линейной обратной связью определяет характеристический многочлен, который определяет период регистра сдвига. Для типа конфигурации Галуа конфигурации (16, 14, 13, 11) соответствует:



(рисунок справедлив для нумерации с 1)

Таким образом, наибольшее число в конфигурации определяет наименьшее возможное количество триггеров для реализации регистра с линейной обратной связью.

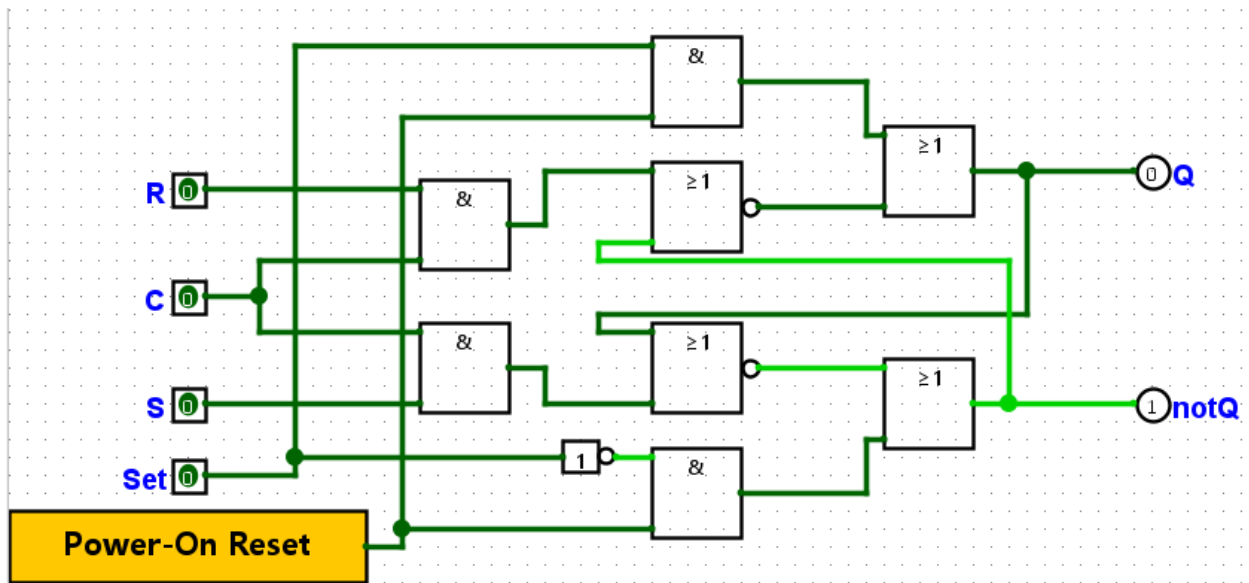
В моем случае с конфигурацией (20, 17) нужно будет использовать 21 триггер (т.к. нумерация с 0).

**Описание вспомогательных логических схем, используемых для построения регистр сдвига с линейной обратной связью конфигурации Галуа**

### **Синхронизированный RS-trigger с модификацией**

Модификация состоит в том, что добавлен дополнительный вход Set, который определяет начальное значение триггера. В остальном реализация аналогична с описанной выше.

Ниже представлена схема RS-trigger с модификацией:

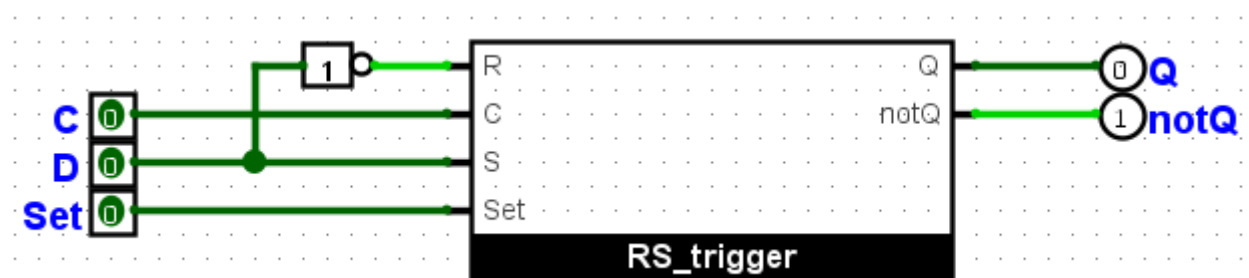


Изменено только то, что POR определяет не только  $\text{not}Q = 1$ , теперь что конкретно определяет POR зависит от Set.

### **D-trigger**

Абсолютно аналогичная реализация, только добавлен новый вход Set, идущий к Set RS-trigger.

Ниже представлена схема D-trigger:

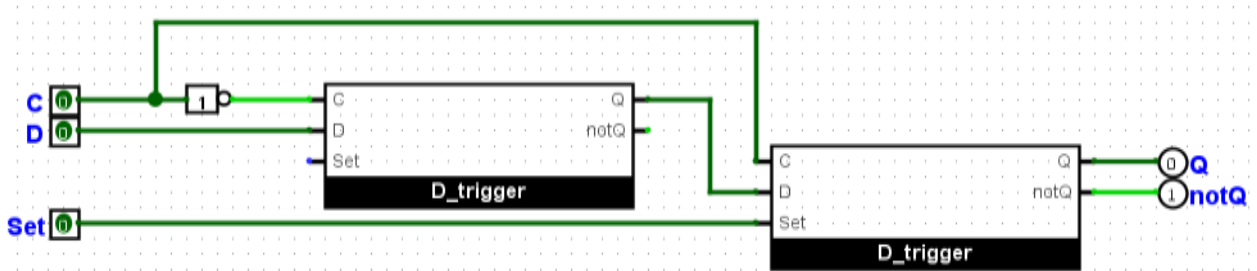




## New D-trigger

Суть изменения та же, что и в Counter\_Basic\_El (и по аналогичным причинам): передача бита происходит с задержкой в 1 такт. Это сделано, потому что при  $C = 0$  значения на входах однозначно определены, а при  $C = 1$  значения в процессе изменения, и может произойти такое, что за период, пока  $C = 1$ , на вход D может быть подано 2 различных значения.

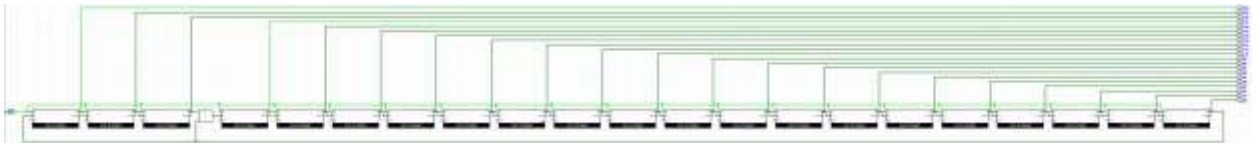
Ниже представлена схема New D-trigger:



## LFSR (linear feedback shift register)

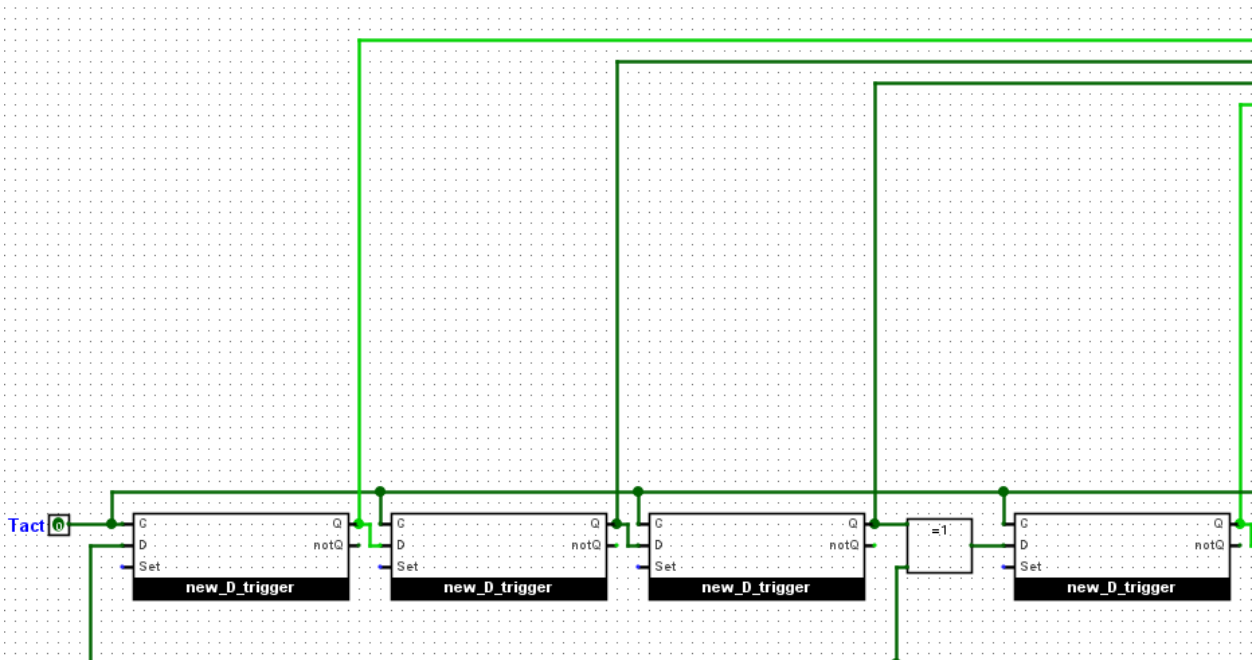
Как уже было сказано, для конфигурации (20, 17) нужно будет использовать 21 триггер, причем только между 18 и 17 будет хор.

Ниже представлена схема LFSR:

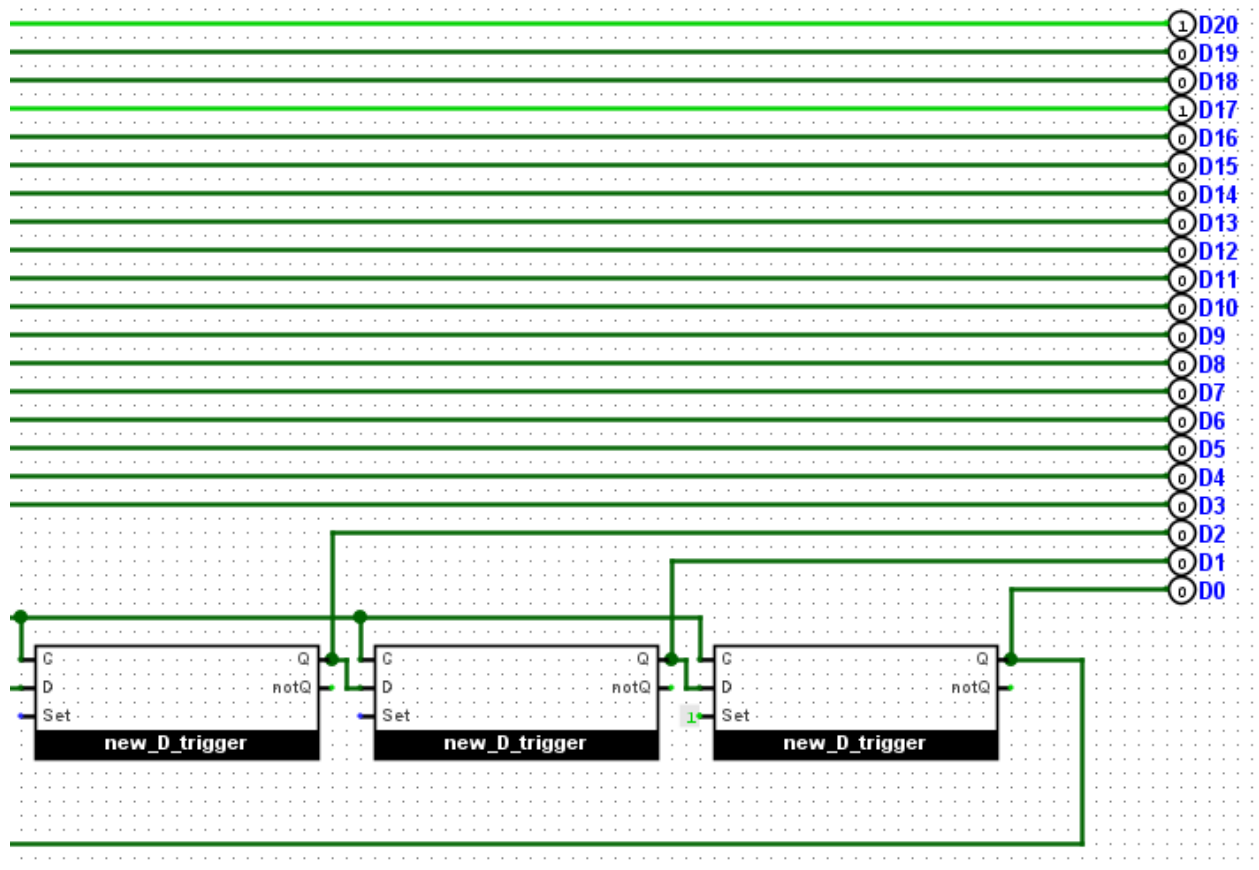


Из-за громоздкости схемы для большей наглядности отдельно представлю начало схемы и конец.

Ниже представлено начало схемы LFSR:



Ниже представлен конец схемы LFSR:



Каждый такт значения передаются от старшего триггера к младшему. В качестве исключения триггер 20, в который передается значение с триггера 0. И только передача с 18 триггера на 17 происходит с помощью операции xor, остальные значения передаются без изменений.

Примечание: если бы изначально значения во всех триггерах были 0, то каждый такт ничего бы не изменялось, из-за в D-trigger была добавлена возможность определить начальное состояние. В моей схеме триггеру 0 было задано начальное значение 1, а остальным 0.