

<i>Лабораторная работа №1</i>	<i>M3137</i>	<i>2022</i>
<i>Построение логических схем в среде моделирования</i>	<i>Булавко Тимофей Евгеньевич</i>	

Цель работы: моделирование логических схем на элементах с памятью.

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

Теоритическая часть

Счётчик:

Для начала дадим определение счётчику, Счётчик — устройство, на выходах которого получается двоичный код, определяемый числом поступивших импульсов. Основной параметр счётчика — модуль счёта — максимальное число единичных сигналов, которое может быть посчитано счётчиком.

Для создания счётчика обычно используют триггеры, чтобы сохранять предыдущие состояния.

Триггеры:

Дадим определение триггерам, которые мы будем использовать для построения нашего счётчика

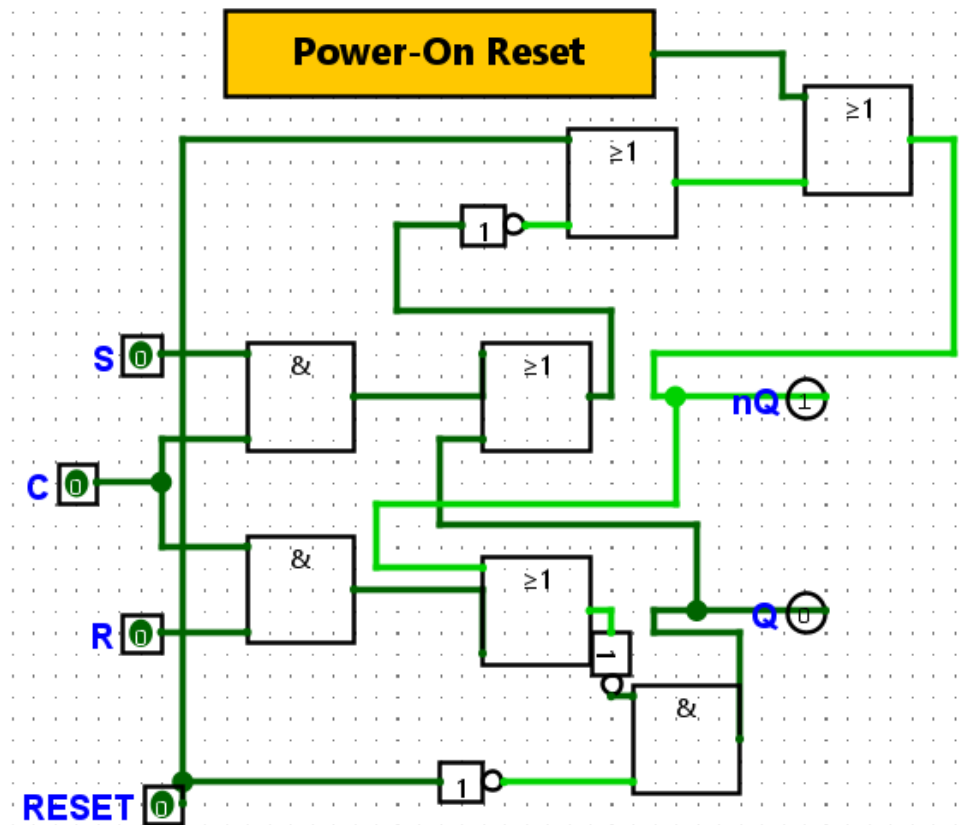
Триггеры — это такие устройства, которые могут долго находиться в одном из двух устойчивых состояний и чередовать их при подаче определенных сигналов на входы. Вся их ценность заключается в способности хранить биты, то есть состояние триггера, используя сразу несколько таких устройств мы получаем возможность хранить данные.

RS-триггер:

RS-триггер - это триггер с отдельной установкой состояний логического нуля и единицы (с отдельным запуском). Он имеет два информационных входа S и R. По входу S триггер устанавливается в состояние $Q=1$ ($nQ=0$), а по входу R - в состояние $Q=0$ ($nQ=1$).

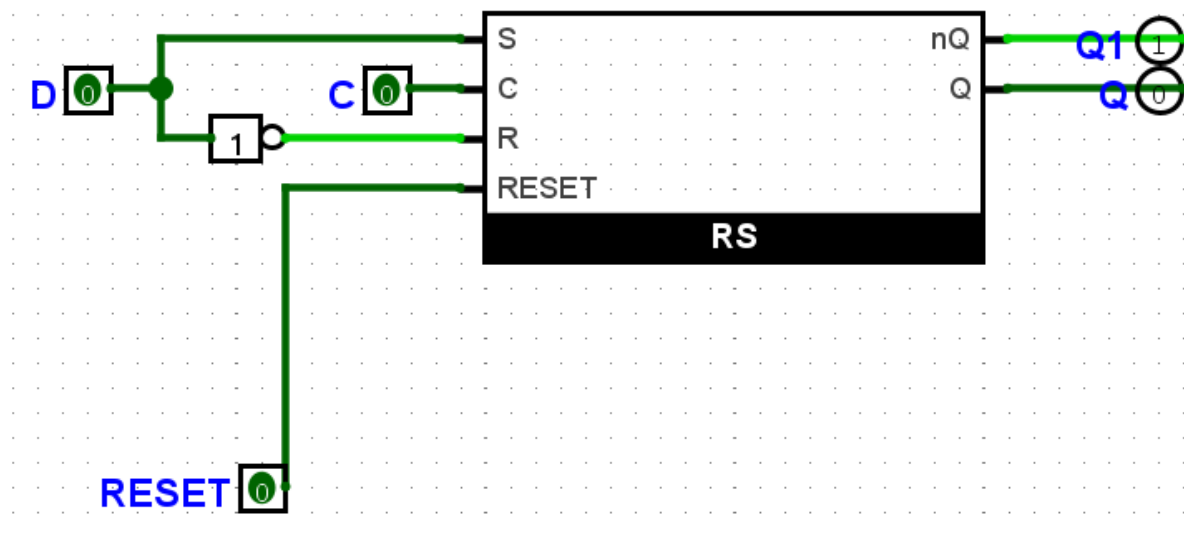
Синхронный RS-триггер:

Синхронные RS-триггеры имеют дополнительный С-вход для подачи управляющих (тактовых, синхронизирующих) импульсов, под действием которых происходит изменение их состояния. Сигналы на информационные S- и R-входы выставляются до подачи синхроимпульса. При нажатии RESET Q сбрасывается на 0 в независимости от R и S.



D-триггер:

D-триггером называется триггер с одним информационным входом, работающий так, что сигнал на выходе после переключения равен сигналу на входе D до переключения, т. е. $Q_{n+1} = D_n$. Основное назначение D-триггеров - задержка сигнала, поданного на вход D. Он имеет информационный вход D (вход данных) и вход синхронизации C.

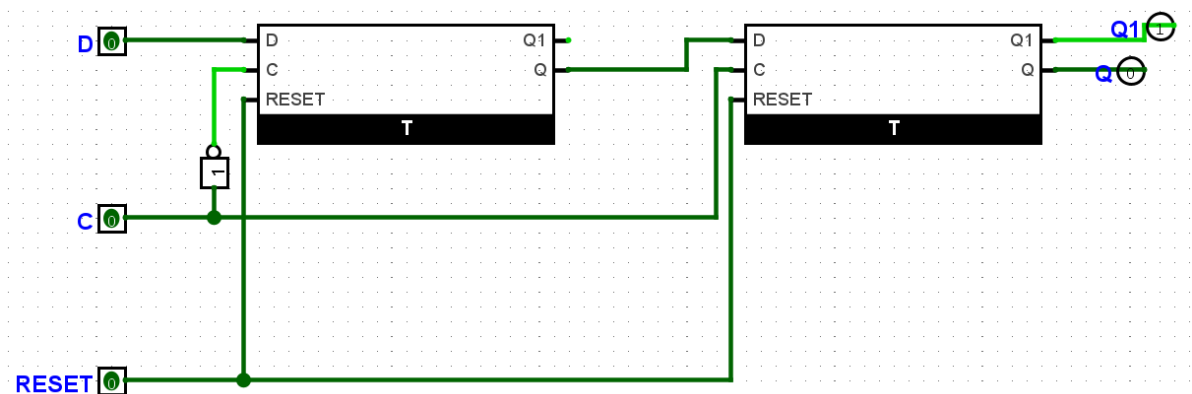


Динамический D-триггер:

Основным недостатком простейшего D-триггера (защелки), рассмотренного в предыдущей статье, является наличие режима "прозрачности". Пока на входе синхронизации присутствует высокий потенциал, триггер записывает входную информацию. Для того, чтобы избежать прохождения входного сигнала на выход схемы приходится на вход триггера подавать очень узкие импульсы.

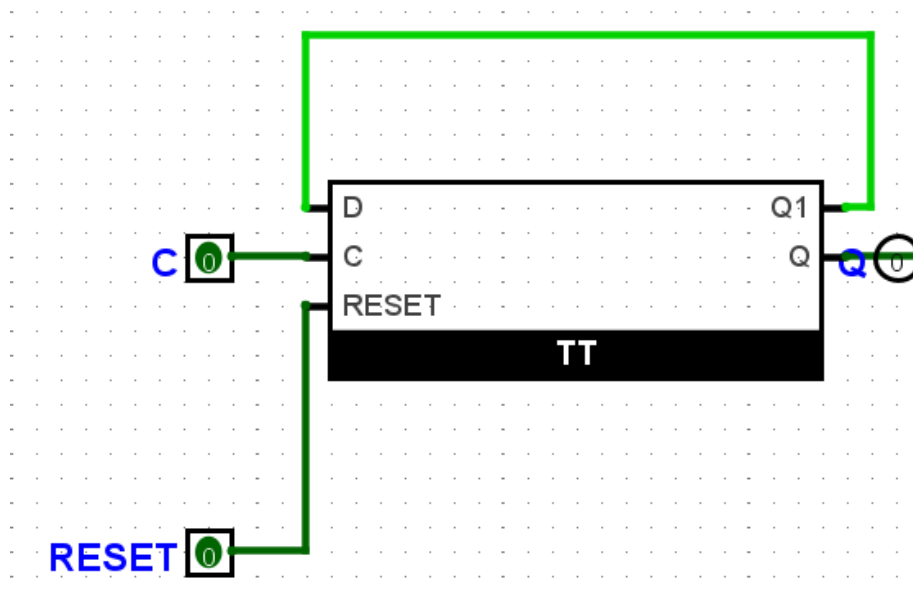
Фронт сигнала синхронизации, в отличие от высокого (или низкого) потенциала, не может длиться продолжительное время. В идеальном случае длительность фронта импульса равна нулю. Поэтому в триггере, запоминающем входную информацию по фронту не нужно предъявлять требования к длительности тактового сигнала.

Динамический D триггер, запоминающий входную информацию по фронту, может быть построен из двух статических D триггеров. Сигнал синхронизации C будем подавать на статические D триггеры в противофазе.



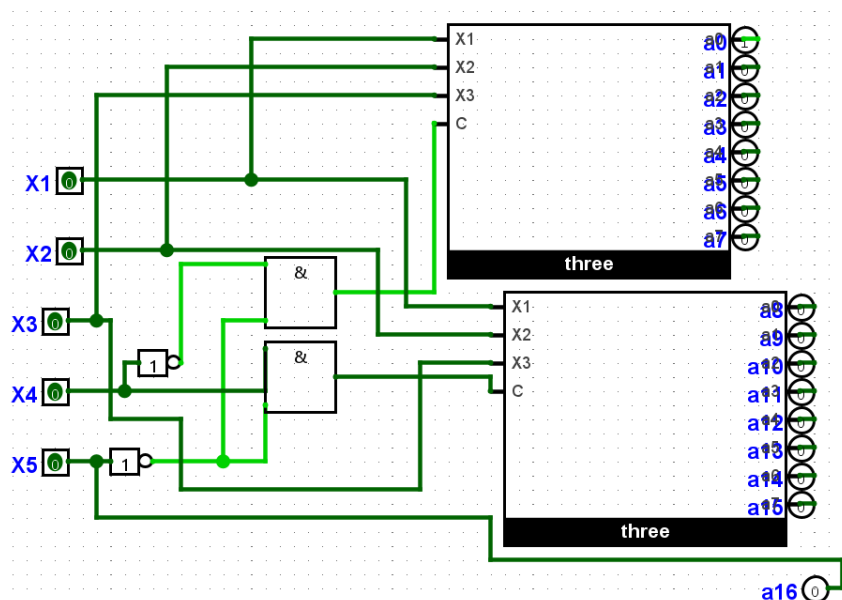
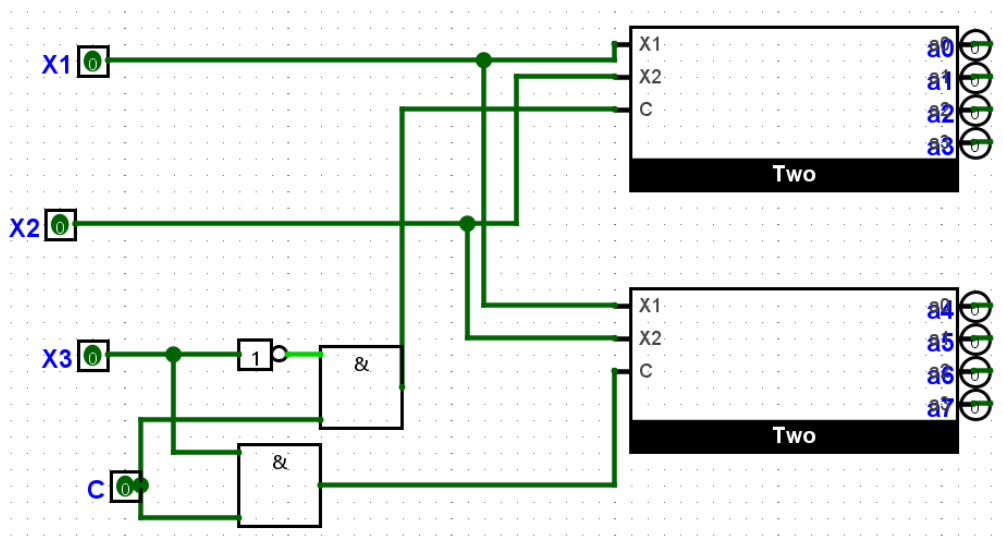
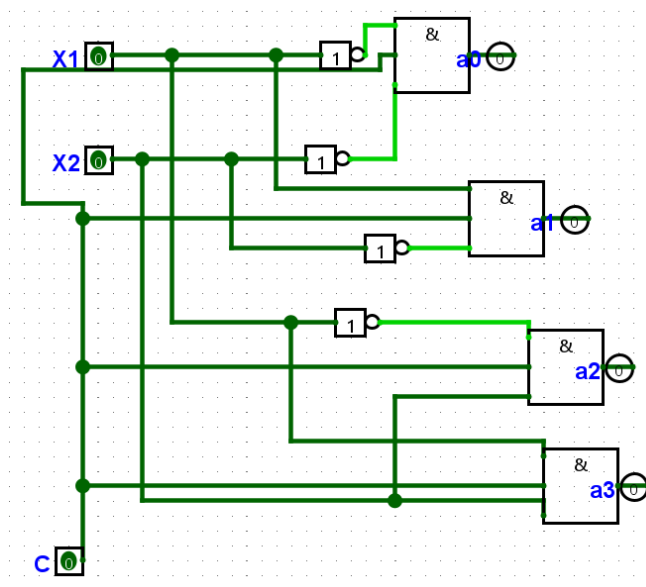
Делитель частоты:

Делитель частоты - электронное устройство, уменьшающее в целое число раз частоту подводимых к нему периодических колебаний.



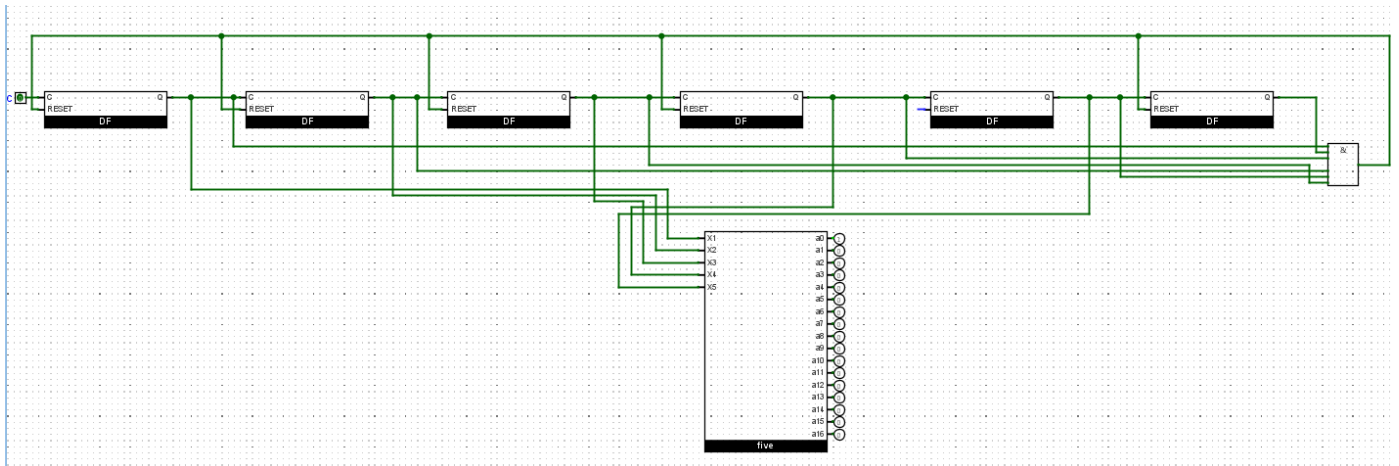
Декодер:

Суть декодера проста – он переводит данное число из двоичной системы счисления в десятичную, и делает он это посредством передачи единицы на выход, номер которого равен десятичному представлению исходного двоичного числа.

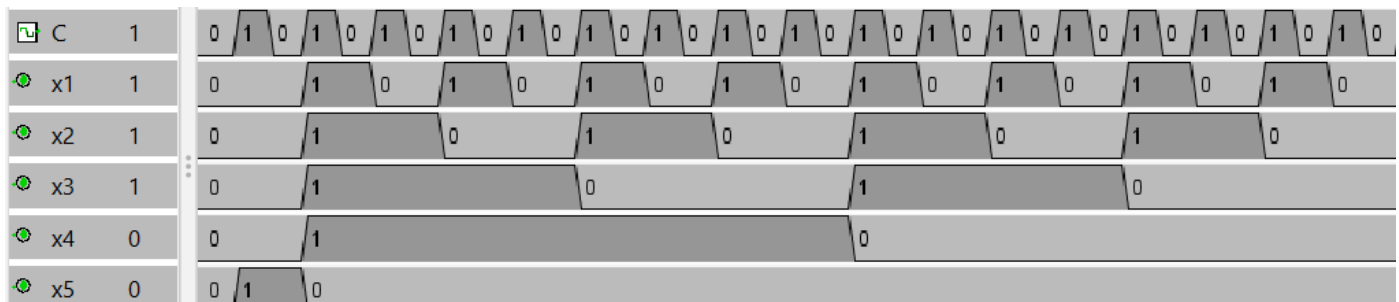


Реализация счётчика

Мой вариант был реализовать асинхронный вычитающий счётчик по модулю 17. Чтобы реализовать сам счётчик нужно подключить несколько делителей частоты подряд, в моём случае это 6. Такой счётчик будет работать по модулю степени 2. Чтобы счётчик работал по модулю 17, нужно – когда все биты равны 1 занести с помощью RESET нужное нам число, в моём случае это 16. Мой счётчик возвращает число в двоичной системе счисления поэтому к нему я подключил декодировщик в десятичную систему.



Временная диаграмма счётчика:



Реализация регистра

В моём варианте требуется реализовать регистр с типом конфигурации Фибоначчи и конфигурацией (6, 1, 0) (конфигурация реализована ссылаясь на [источник](#)). По конфигурации мы узнаем что 0-й бит это бит ввода, а 0-й и 6-й это биты отвода. Если подключить динамические биты в цикл то получится схема делающая циклические сдвиги битов. Так в моей случае тип конфигурации Фибоначчи нужно все биты отвода провести через исключаящее или, и отправить в бит ввода. Чтобы установить начальное значение битов нужно использовать d0, d1, d2, d3, d4, d5, d0 – младший бит, d5 – старший.

