Лабораторная работа №1	M3136	2022
Построение логических схем в среду моделирования	Исаев Максим Викторович	

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

1. Асинхронный вычитающий счётчик по модулю 18.
2. Регистр сдвига с линейной обратной связью в конфигурации Фибоначчи с отводной последовательностью (11, 9, 0).

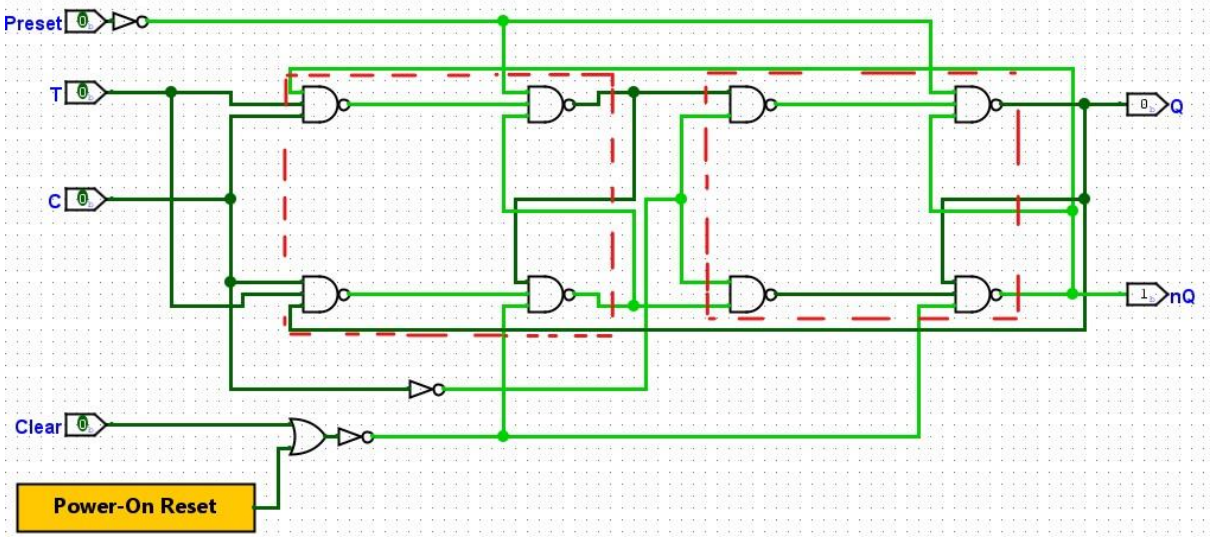
Счётчик.

В моём варианте нужно сделать асинхронный вычитающий счётчик по модулю 18. Счётчик — это элемент, который подсчитывает количество импульсов, поданных ему на вход. Счётчик по модулю n — это счётчик, который с переключается n раз, от 0 до n – 1, прежде чем обнулится. Вычитающий означает, что отсчёт ведётся в обратном направлении, от n - 1 до 0. Характеристика «асинхронный» говорит о том, что внутри счётчика сигнал распространяется последовательно, от одного триггера к другому.

T триггер.

Счётчик собран на основе синхронных T триггеров с асинхронным Reset/Clear. T триггер – это элемент запоминающий своё состояние и меняющая его по сигналу. Я использовал двухступенчатый вариант T триггера на двух RS триггерах (обведены красным на схеме). Ниже приведены таблица истинности и схема.

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}



Синхронность означает наличие синхронизирующего импульса C, по которому происходит переключение.

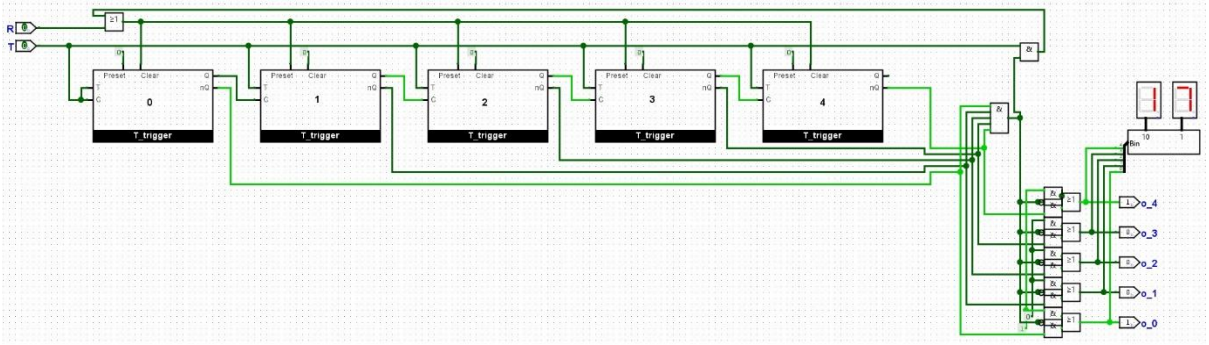
Асинхронные Reset и Clear позволяют в обход синхронизирующего импульса переключить триггер в одно из двух состояний, 1 и 0 соответственно. Ниже приведена таблица истинности для этих входов.

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

Состояние Reset = 1, Clear = 1 является невалидным. Так же стоит отметить, что сигналы с входов Reset и Clear инвертируются, потому что иначе бы они были active low. Power-On Reset подаёт короткий импульс в начале симуляции и устанавливает триггер в состояние 0, иначе начальное состояние было бы не определено.

Собственно счётчик.

На ниже рисунке представлена схема счётчика.

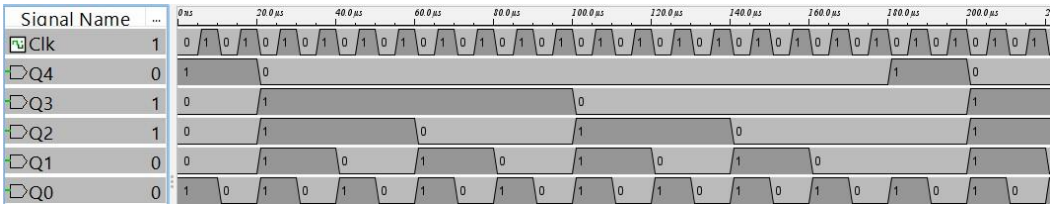


Несколько T триггеров соединены последовательно и каждый из них кодирует один разряд в двоичной системе. Видно, что входы Clock подключены последовательно, выход предыдущего является входом следующего. Это и даёт асинхронность переключения – триггеры переключаются по очереди. Из-за асинхронности переключение занимает время, зависящее от количества триггеров, а также во время переключения выходные данные не валидны, т. к. не все переключения ещё произошли.

Переключение триггера происходит на falling edge, т. е. при переходе сигнала C с 1 на 0. Обратим внимание, что входы Clock триггеров подключены к выходам Q, а выходные данные мы берём с $\overline{Q}$, это обеспечивает, то, что счётчик является вычитающим. Число, кодируемое выходами Q в порядке от 0 к 4, возрастает, а число кодируемое $\overline{Q}$ в порядке от 4 к 0 убывает, т. к. они являются побитовыми инверсиями друг друга.

Временная диаграмма.

Временная диаграмма счётчика сделана при помощи внутренних средств Logisim evolution. Она приведена ниже.



Первая строка показывает сигнал, генерируемый внешним осциллятором и подаваемый на вход счётчику. Видно, что в нулевой момент двоичное значение счётчика 0b10001, что эквивалентно 17 (n – 1). Так же заметно что переключение происходит на falling edge, как описывалось раньше. Когда состояние доходит до 0b00000 то происходит сброс обратно на 0b10001, потому -1 эквивалентно 17 по модулю 18.

Куда тыкать?

В файле counter.circ, помимо счётчика и триггера, есть схема main, в которой можно проверить его работоспособность. На корпусе счётчика есть HEX Digits Display на котором отображается текущее состояние. При запуске счётчик сразу готов к работе. Вы можете сбросить его в любой момент, нажав Reset.

Регистр сдвига с линейной обратной связью.

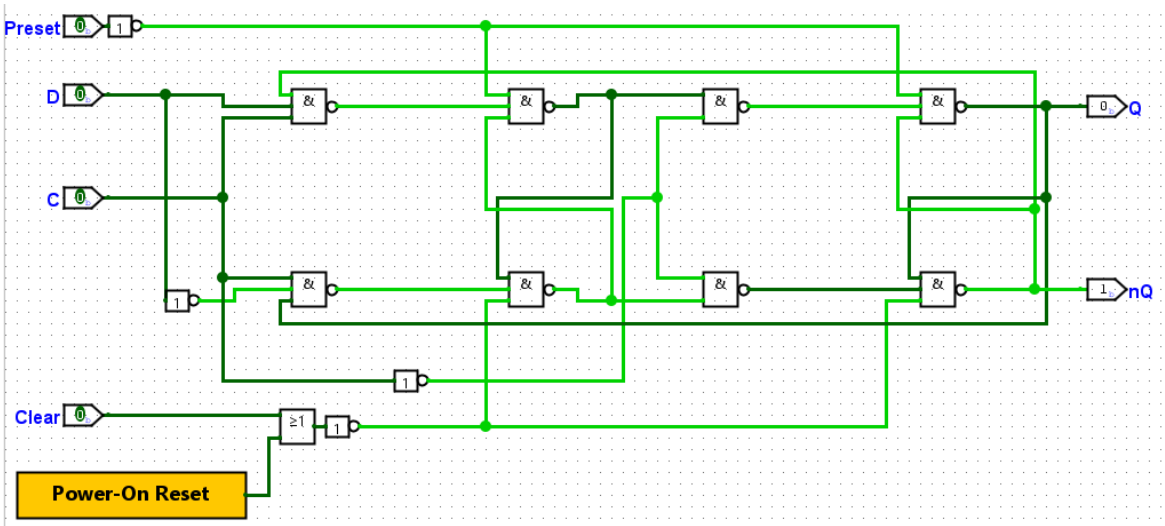
В моём варианте нужно сделать регистр сдвига в конфигурации Фибоначчи с отводной последовательностью (11, 9, 0). Регистр — это последовательно соединенные триггеры, которые хранят какую-то последовательность и по сигналу сдвигает её, передавая хранимые значения на вход следующим. В конфигурации Фибоначчи новый бит определяется суммированием значений находившихся на определённый позициях (вообще оно высчитывается как сумма значений в степени равной индексу ячейки, но у нас только 0 и 1, а они в любой степени равны себе). Эти позиции задаются конфигурацией отводной последовательности. В моём случае это 11, 9 и 0 позиции. Суммирование по модулю 2 производим при помощи XOR.

Для построение регистра я использовал D триггеры.

D триггер.

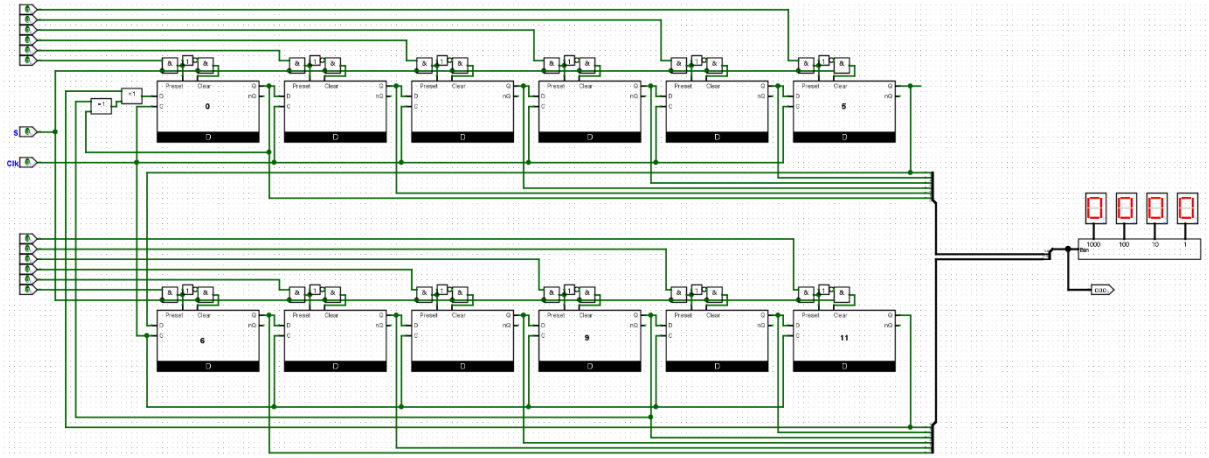
D триггер хранит своё состояние и по сигналу C меняет его на то, которое ему подали на вход D. Он построен аналогично T триггеру из счётчика и так же оборудован асинхронным сбросом. Ниже приведена его таблица истинности и схема.

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



Регистр.

Ниже приведена схема регистра.



Я использую 12 D триггеров, т. к. это минимально необходимое количество, для моей конфигурации. Все они подключены последовательно и по сигналу предают значение следующему. Значения триггеров с номерами 11, 9 и 0 складываются XOR’ом и подаются на вход в 0 триггер.

Куда тыкать?

В файле lfsr.circ есть схема main, где можно протестировать регистр. Вы можете задать любое начальное состояние подав нужный сигнал на 12 контактов и затем нажав set. На корпусе расположен HEX Digits Display, на котором отображается текущее значение в регистре. Побитовое состояние регистра можно увидеть на LED Bar подключённом к 12 битному выходу.