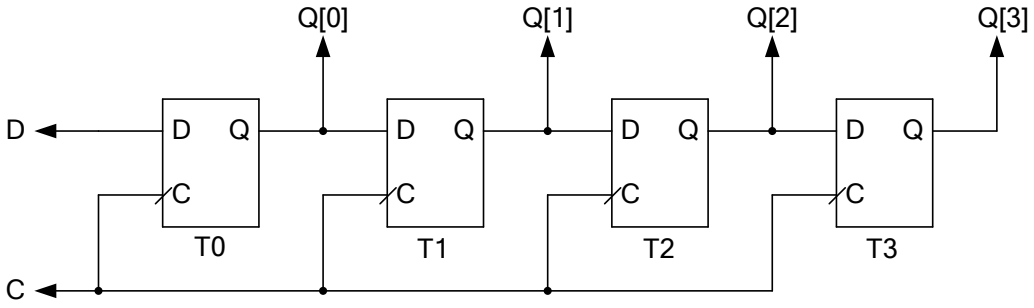
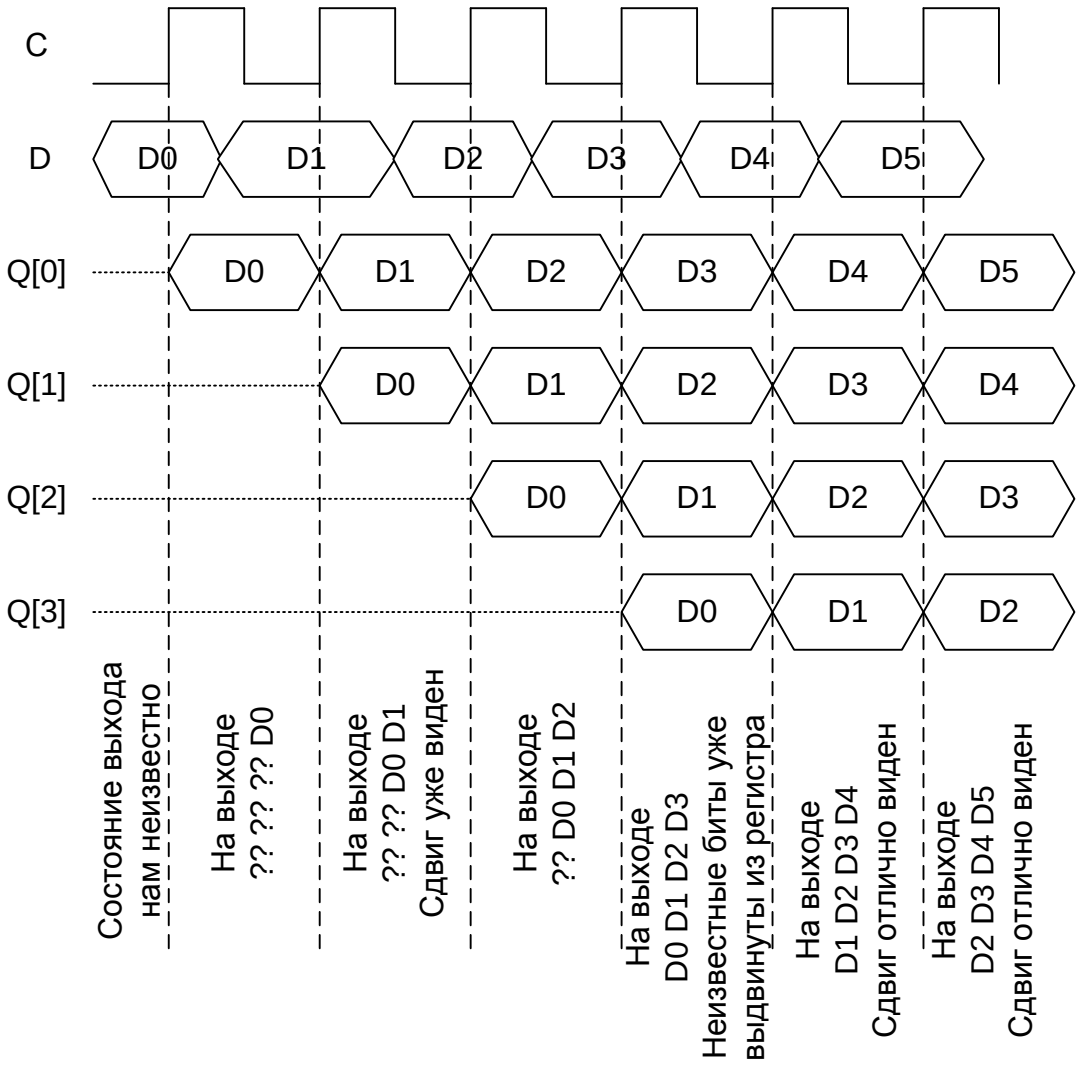
Регистры сдвига

Любой, кто связан с компьютером, чует нутром, что там, где есть что-то параллельное, явно есть и что-то последовательное. Не являются исключением регистры, правда, тут придумано другое название – регистр сдвига (Shift Register). Давайте посмотрим, что это за такая полезная штука.

Включим триггеры следующим образом:



Исходно состояние регистра нам не известно. Но вот мы подали импульс на вход C. В этот момент, в триггер T0 защёлкнулось то, что было на входе D. И, по всем канонам работы триггера, оно попало на выход Q[0]. При следующем импульсе C, данные с Q[0] запрыгнут в триггер T1 (переедут в Q[1]), а на Q[0] попадут новые данные со входа. Через 4 импульса, у нас будет известно состояние всех четырёх выходов Q. Причём Q[0] будет содержать состояние входа D 0 импульсов назад, Q[1] – 1 импульс назад, и т.д. Графически, это можно представить так:

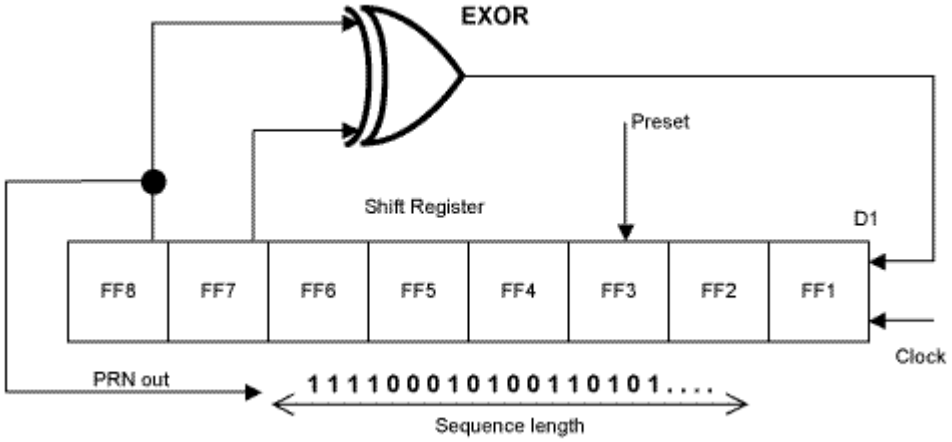


Практические применения регистров сдвига

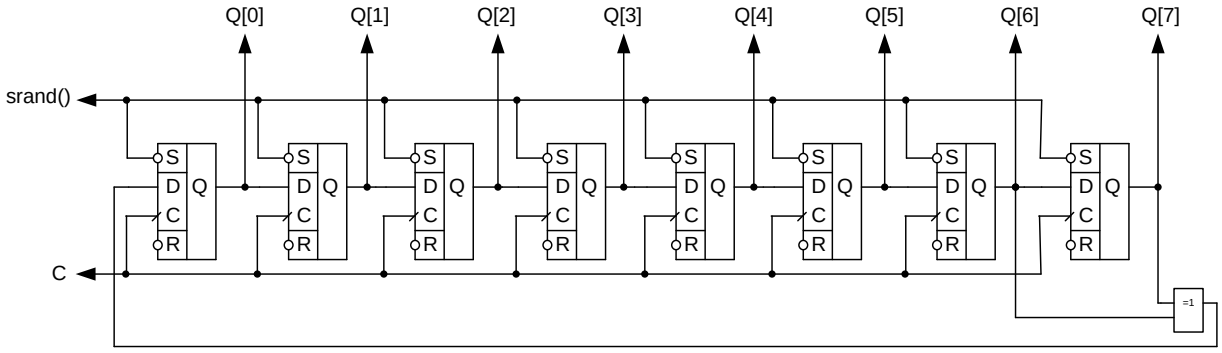
Собственно, с теорией на сегодня всё. Регистр сдвига, как и параллельный регистр – штука очень простая, про него больше пока что рассказать нечего (дойдём до мультиплексоров – ещё дорасскажу). А пока что – давайте рассмотрим пару практических схем с использованием регистра сдвига.

М-последовательность

Изучая прикладную математику, мы почти месяц потратили на М-последовательности. Каких только замечательных свойств у неё нет. Жаль только, что программно реализовать её сложно. Однако, кто нам мешает сделать её аппаратно? Допустим, мы делаем какой-то быстродействующий процессор, у которого есть команда «Выдать очередное псевдослучайное число». Как нам быть? Идём на Гугля, пишем «М-последовательность», смотрим результаты. Например, этот: http://vrtp.ru/index.php?showtopic=9500 . Прекрасно, нам там почти всё и нарисовано, вот картинка с сайта.



Значит, если наш процессор восьмиразрядный, нам в него потребуется добавить элементарный блок:



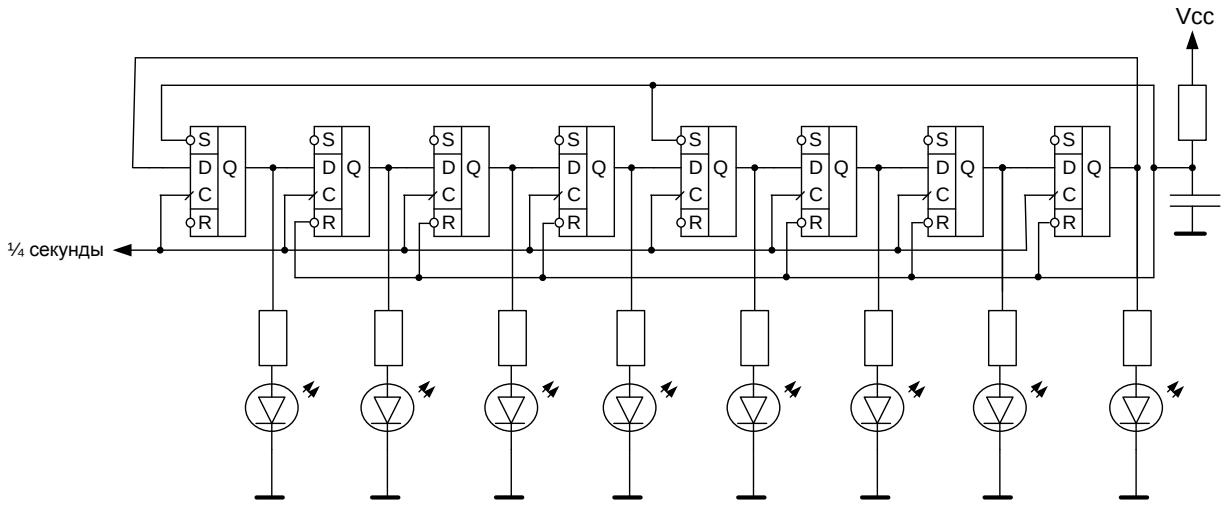
Собственно, всё. Пусть в нашем гипотетическом ассемблере есть команда srand без аргументов (можно и с аргументами, но схема будет сложнее), которая устанавливает все триггеры в 1, а также команда rand, которая

1) Считывает содержимое Q[7..0] и выдаёт его, как результат
2) Подаёт 8 импульсов C для того, чтобы следующая команда смогла считать новое значение.

Быстродействие такого процессора будет желать лучшего (подумать только! 8 тактов на команду!), однако, при желании, обратные связи в регистре можно закрутить так, что всё будет выполняться за 1 такт, но к регистрам сдвига, о которых мы сейчас говорим, это не имеет никакого отношения. Зато мы сделали наше первое устройство на регистре сдвига, и оно, кроме того, может быть применено в нашем гипотетическом самодельном процессоре, а это уже немало.

Бегущие огни

Неумолимо приближается Новый год. Давайте сделаем что-нибудь для него. Ну хотя бы простейшие бегущие огни. Огонь бежит из лампочки в лампочку, так что здесь наверняка также не обошлось без регистра сдвига.



Справа стоит уже известный нам смывной бачок, он даст кратковременный ноль при подаче питания. В результате, на тех триггерах, которые подключены входом S к сигналу Reset, загорится огонь, а которые нет – огонь будет потушен. В нашей схеме загорится два огня. Как только сигнал Reset будет снят, регистр сдвига начнёт свою работу. С выхода последнего триггера, сигнал вновь подаётся на вход первого, так что огни будут бегать по замкнутому кругу.

Где взять импульсы с частотой ¼ секунды – мы рассматривали в статье про счётчики. Вот так легко и непринуждённо мы разработали симпатичную мигалочку, основанную на всё том же регистре сдвига. В былые времена, у неё была бы одна сложность – слишком много микросхем потребовалось бы на реализацию (одна микросхема К555ТМ2 содержит всего два триггера, бывают микросхемы - регистры сдвига, но всё равно, пришлось бы отдельно взять микросхему регистра сдвига и отдельно микросхему счётчика, в общем, жуть), но во времена ПЛИС, микросхема практически всегда одна – сама ПЛИС. Если у неё достаточная ёмкость, то все наши изыски в неё поместятся, ещё и останется немного места для других функций. Наша задача – всего лишь придумать схему и нарисовать её в редакторе. Остальное – проблемы компилятора.

Ну, вроде, по триггерам, в общих чертах, всё. Статья получилась короткая, но мешать регистры сдвига с параллельными не хотелось, так как они всё-таки разные. Поэтому пока поставим на триггерах точку и перейдём к составным логическим функциям. Правда, сделаем это уже в следующий раз.