# Лабораторная работа №1.

Это первая лабораторная работа. В этой работе Вам необходимо написать свой модуль пересинхронизации между двумя клоковыми доменами. В действительности, в этой работе не будет два полностью асинхронных домена, так что необходимости в использовании пересинхронизаторов в этом случае нет. Но т.к. это не реальный проект, а образовательный, все же предлагаю его сделать.

1. Создайте два делителя частоты clk0 и clk1 с разной выходной частотой. Можете либо создать два блока, либо один с использованием параметра деления частоты.

2. Используя библиотеку IP модулей (tools->IP Library, появится окно в правой части Quartus) создайте блок ROM памяти с одним входом:

   (a) Установите размер одного слова 16 бит, размер памяти 16 слов.

   (b) Создайте файл начальной инициализации ROM памяти (расширение .hex). Пока пропишите там любые произвольные значения. Удобно пользоваться редактором Vim, после открытия пропишите опцию :%!xxd для переходя в режим hex. Укажите этот файл на этапе создания IP блока RAM.

   (c) Завершите генерацию блока со стандартными опциями. Понаблюдайте, что этот файл появился в файлах проекта. Для этого переключитесь в дереве иерархии проекта в режим всех файлов.

3. Запишите в файл инициализации ROM памяти код Грея. В каждом слове из 16 бит 4 числа по 4 бита (т.е. 4 hex числа). При этом каждое hex число является любо 0 либо 1. Т.е. в файле должны быть строки типа 0000, 0001, 0011, 0010, ..., при этом каждый ноль либо единица занимают четыре бита.

4. Создайте блок пересинхронизации. Необходимо создать блок описанный в статье `https://pdf4pro.com/view/clock-domain-crossing-cdc-design-amp-verification-2b742.html` на странице 27. Входы данных при этом должны быть размера 16 бит.

5. Создайте блок вывода четырех hex чисел на четыре семисегментных индикатора. Скорее всего Вам потребуется написать еще один делитель клока.

6. Подключением блоков и написанием небольших вспомогательных блоков реализуйте следующий функционал: часть схемы, работающая по clk0 читает значение из памяти и через блок пересинхронизатора отправляет прочитанное значение в часть схемы, работающую по clk1. Далее, часть схемы работающей от clk0 ждет, пока к ней не придет ack из блока пересинхронизации от клокового домена clk1. После этого загружает следующее значение и далее все повторяет. Не забудьте, что необходимо правильно управлять сигналами пересинхронизатора в clk0. При этом часть схемы, работающая от clk1, выводит полученные значения на на семисегментный индикатор. Позаботьтесь, чтобы период clk1 был таким, чтобы значения на семисегментном индикаторе хорошо воспринимались глазами.

7. Загрузите прошивку на ПЛИС. Проверьте, что при разных соотношениях клоков clk0 и clk1 (clk0 больше или меньше clk1) значения не теряются и успешно передаются.

8. Вышлите мне код проекта и видео с работой для случая clk0 больше clk1 и clk0 меньше clk1.