

- В папке C:\Xilinx_trn\HLS2023\lab2_z1 создать папку **source**
- Создать на языке C++ функцию (lab2_z1.cpp) поиска факториала (без рекурсии) – ниже пример псевдокода.
 - Сохранить ее в папку C:\Xilinx_trn\HLS2023\lab2_z1\source
 - Имя функции lab2_z1
 - Обязательно наличие lab2_z1.h файла с заданием типов данных (Сохранить в папку C:\Xilinx_trn\HLS2023\lab2_z1\source).
 - Базовый тип данных: n – unsigned char; возвращаемое значение – unsigned int (Обратите внимание на то, что n не должно быть больше 13, иначе будет переполнение результата)

Факториал	Рекурсия	без рекурсии
$\left\{ \begin{array}{l} n! = n * (n-1)!, \text{ при} \\ n > 0 \\ \\ 1, n=0 \end{array} \right.$	<pre>unsigned int Fact(n: unsigned char) begin if n=0 then Fact:=1 else Fact:=n*Fact(n-1) end;</pre>	<pre>unsigned int Fact(n: unsigned char) var F, i: unsigned int; begin F:=1; for i:=1 to n do F:=F*i; Fact:=F end;</pre>

- Создать на языке C++ тест (lab2_z1_test.cpp) для проверки работы функции - сохранить в папку C:\Xilinx_trn\HLS2023\lab2_z1\source. Тест должен обеспечивать
 - Чтение типов данных из lab2_z1.h файла
 - запуск функции 3 раза,
 - задание n случайным значением (новое значение для каждого запуска функции), распределенным от 5 до 13 (например так: n= (rand() %9) + 5)) . Обратите внимание на то, что n не должно быть больше 13, иначе будет переполнение результата
 - проверку правильности вычисленного результата (для проверки использовать **алгоритм с рекурсией**) и формирование признака успешного/неуспешного выполнения для каждого запуска функции,
- Отладить функцию и тест (при неправильном результате в любом из запусков функции должен сообщать об ошибке).
- Создать скрипт (сохранить в папку C:\Xilinx_trn\HLS2023\lab2_z1) автоматизирующий процесс:
 - Создания проекта lab2_z1,
 - Подключения файла lab2_z1.cpp (папка source),
 - Подключения файла lab2_z1_test.cpp (папка source),
 - Создания базового решения (ex_sol1), для которого
 - задается микросхема: xa7a12tcsg325-1q ,
 - задается clock period 4; clock_uncertainty 1 ,
 - выполняется Си моделирование.
 - Исследования в объеме 3-х решений - ex_sol[4:2]
 - Для всех решений задается микросхема: xa7a12tcsg325-1q
 - Для каждого решения отдельное требование к периоду тактового сигнала
 - Для ex_sol2 задается clock period 8; clock_uncertainty 1
 - Для ex_sol3 задается clock period 12; clock_uncertainty 1
 - Для ex_sol4 задается clock period 40; clock_uncertainty 1
 - Для каждого решения осуществить синтез.

- Для каждого решения осуществить моделирование cosim.
- Отладить и проверить работу созданного скрипта.
- открыть GUI
- убедиться, что созданы все решения
- используя средства HLS сравнить полученные решения.
- Составить электронную таблицу для сравнения решений (перенести в нее данные из HLS) и построить график, в котором для всех решений должны быть отражены: Iteration Interval (ns) – подсчитывается путем умножения Latency (cycles) на период тактового сигнала Estimated; использованные ресурсы (если значения какого-либо ресурса остаются неизменными для всех решений, то такой ресурс не следует отображать на временной диаграмме)
- Посмотреть, привести в отчете и сравнить временные диаграммы для решения ex_sol1 ... ex_sol4
- Оформить отчет, который должен включать
 - Задание
 - Раздел с описанием исходного кода функции
 - Раздел с описанием теста
 - Раздел с описанием созданного командного файла
 - Раздел с описанием результатов сравнения решений (со снимком экрана из vitis HLS)
 - Раздел с анализом результатов (со снимком экрана **с заполненной таблицей и полученным графиком**)
 - Анализ и выбор оптимального (критерий максимальная производительность) решения
 - Анализ и выбор оптимального (критерий минимальные аппаратные затраты) решения
 - Анализ и выбор оптимального (критерий максимальная производительность и минимальные аппаратные затраты) решения
 - Выводы
- Архив должен включать всю папку C:\Xilinx_trn\HLS2023\lab2_z1, (в папке ./doc должны быть: задание, отчет и файл с электронной таблицей).