Lab3\_z1

* В папке C:\Xilinx\_trn\HLS2023\lab3\_z1 создайте папку source (для исходных файлов и теста)
* Создайте на языке С++ (и сохраните в папку C:\Xilinx\_trn\HLS2023\lab3\_z1\source)
  + Функцию (файл lab3\_z1.сpp) ,

# include “lab3\_z1.h”

data\_ret lab3\_z1 (data\_in inA, data\_in inB, pnt\_in \*inC, pnt\_out \*res) {

data\_ret temp;

temp = inA \* \*inC;

\*res = temp + inB;

return temp;}

* + lab3\_z1.h файл: data\_ret – int; data\_in, pnt\_in – short; pnt\_out - int
  + тест lab3\_z1\_test.cpp для проверки работы функции. Тест должен обеспечивать
    - запуск функции 3 раза,
    - заполнение данных случайными числами.
    - Очистку \*res между запусками функции
    - проверку правильности вычисленного результата и формирование признака успешного/неуспешного выполнения для каждого запуска функции,
* Отладьте функцию и тест (при неправильном результате в любом из запусков функции должен сообщать об ошибке).
* Создайте скрипт lab3\_z1.tcl автоматизирующий процесс:
  + Создания проекта lab3\_z1,
    - * Микросхема: xa7a12tcsg325-1Q
      * Период тактового сигнала: 8нс, uncertainty 1нс.
  + Подключения файла lab3\_z1.cpp, lab3\_z1.h (папка source ),
  + Подключения файла lab3\_z1\_test.cpp (папка source),
  + Си моделирования
  + Создания решений
    - sol1,
      * Block Level I/O interface по умолчанию
      * Синтез
      * С/RTL cosimulation (с опцией полной трассировки)
    - sol2,
      * Block Level I/O interface ap\_ctrl\_chain
        + *Для упрощения рекомендуется задать эти настройки в GUI пакета и затем скопировать их из автоматически создаваемого пакетом файла directives.tcl в файл lab3\_z1.tcl*

*set\_directive\_interface -mode ap\_ctrl\_chain "lab3\_z1"*

* + - * Синтез
      * С/RTL cosimulation (с опцией полной трассировки)
* Отладьте и проверьте работу созданного скрипта.
* После выполнения скрипта откройте GUI
* Убедитесь, что созданы все решения
* Используя средства HLS сравните полученные решения
  + аппаратные затраты
  + результаты планирования
  + использованные интерфейсы
    - составьте таблицу использованных интерфейсов для Block Level I/O interface и Port Level I/O interface.
  + результаты планирования (Schedule viewer)
  + С/RTL cosimulation
  + Заполненную xls таблицу и полученный график
  + объясните полученные результаты.
* Оформите отчет, который должен включать
  + Задание
  + Раздел с описанием исходного кода функции
  + Раздел с описанием теста
  + Раздел с описание созданного командного файла
  + Раздел с описанием результатов сравнения решений (со снимком экрана из Vitis HLS)
  + Выводы
* Архив должен включать всю рабочую папку проекта, отчет и файл с электронной таблицей.