

- Создать на языке Си функцию,
  - Считывающую элементы массива D\_I, элементы массива имеют тип int (32 бита), размер массива 256 элемента, в функцию передается массив.
  - Считывающую элементы массива C\_I, элементы массива имеют тип int (32 бита), размер массива 16 элементов, в функцию передается указатель.
  - произведение i го элемента массива D\_I и j-го (от 0 до 15) элемента массива C\_I записывается в i элемент массива D\_O
  - Сделать ДВЕ реализации функции
    - Первая – обычное считывание (lab3\_z2s.c)
    - Вторая – считывание C\_I в буфер с помощью memcpy (lab3\_z2b.c)

По примеру (слайд 53 в лекции)

### > Standard Mode

- >> Each access to the bus results in a request then a read or write operation
- >> Multiple read or writes can be performed in a single transaction

#### Single read and write in Standard Mode

```
#include "bus.h"

void foo (int *d) {
    static int acc = 0;
    int i;

    acc += d[i];
    d[i] = acc;
}
```

#### Multiple reads and writes in Standard Mode

```
#include "bus.h"

void foo (int *d) {
    static int acc = 0;
    int i;

    for (i=0;i<4;i++) {
        acc += d[i];
        d[i] = acc;
    }
}
```

```
#include "bus.h"

void foo (int *d) {
    static int acc = 0;
    int i;

    for (i=0;i<4;i++) {
        acc += *(d+i);
        *(d+i) = acc;
    }
}
```

### > Burst Mode

- >> Use the memcpy command
- >> Copies data between array & a pointer argument
- >> The pointer argument can be a bus interface
  - This example uses a size of 4
  - This is more efficient for higher values

```
#include "bus.h"

void foo (int *d) {
    int buf1[4], buf2[4];
    int i;

    memcpy(buf1,d,4*sizeof(int));

    for (i=0;i<4;i++) {
        buf2[i] = buf1[3-i];
    }

    memcpy(d,buf2,4*sizeof(int));
}
```

- Создать на языке Си тест для проверки работы функции. Тест должен обеспечивать
  - запуск функции минимум 2 раза,
    - На вход функции поступает массив (созданный с использованием генератора случайных чисел), элементы массива имеют тип int (32 бита).
  - проверку правильности вычисленного результата и формирование признака успешного/неуспешного выполнения для каждого запуска функции,
- Отладить функции и тест (при неправильном результате в любом из запусков функции должен сообщать об ошибке).
- Создать скрипт автоматизирующий процесс:
  - Создания проекта lab3\_z2,

- Подключения файла lab3\_z2\_test.c (папка source),
  - задается микросхема: xa7a12tcs325-1q ,
  - задается clock period 6; clock\_uncertainty 0.1 ,
- Создания решения (
  - sol1, для которого Block Level I/O interface по умолчанию и используется функция первого варианта
    - Подключения файла lab3\_z2s.c (папка source ),
  - sol2, для которого Block Level I/O interface по умолчанию и используется функция второго варианта
    - Подключения файла lab3\_z2b.c (папка source ),
- Отладить и проверить работу созданного скрипта.
- После выполнения скрипта открыть GUI
- Убедиться, что созданы все решения
- Используя средства HLS сравнить полученные решения
  - аппаратные затраты
  - результаты планирования
  - использованные интерфейсы
  - привести временные диаграммы и объяснить в чем их отличия.
- Оформить отчет, который должен включать
  - Задание
  - Раздел с описанием исходного кода функций
  - Раздел с описанием теста
  - Раздел с описанием созданного командного файла
  - Раздел с описанием результатов сравнения решений (со снимком экрана из Vivado 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 рабочую папку проекта, отчет и файл с электронной таблицей.