SHORT USER GUIDE ADD TWO VALUES

Исходные данные

1. Есть два компьютера
    a. Сервер VM-XILINX (Windows 7). К нему подключаемся через RDP. Тут работаем в Vivado HLS и собираем битстрим в Vivado.
        i. Пароль для всех студентов 123456
    b. Сервер Kraft (Ubuntu). К нему подключен FPGA блок. Заходим на неге через Putty SSH через Windows машину VM-XILINX. Тут запускаем реальное прикладное приложение (хост-программу).
2. Исходные файлы
    a. На сервере Kraft - установленная Rosta SDK. Расположение - /opt/RostaSDK
        i. Готовая исходная хост-программа (общая) /opt/RostaSDK/lin/training
        ii. Утилиты (реконфигурация FPGA) /opt/RostaSDK/lin/utilities
    b. На VM-XILINX в вашей домашней директории две копии одного проекта rc_pcie_base_2000_11 и готовый собранный битстрим

Порядок действий

1. Запустить готовое приложение (готовый битстрим и хост-программу)
    a. Скопировать готовый битстрим с VM-XILINX на сервер Kraft
    b. На сервере Kraft запустить mc и скопировать хост-программу /opt/RostaSDK/lin/training/add_test.cpp /opt/RostaSDK/lin/training/add_test.h в свою домашнюю директорию
    c. Запустить утилиту конфигурации с сконфигурировать FPGA скопированным на Kraft битстримом
    d. Запустить хост-программу add_test
2. Самостоятельно собрать проект FPGA без изменений в проекте
    a. Проекты лежат в папке xilprojects (есть ярлык на рабочем столе)
3. Изучение и внесение изменений в приложение

Сборка FPGA проекта, генерация файла конфигурации ПЛИС

1. Перейти в папку проекта rc_pcie_base_2000_11
2. Перейти в папку project
3. Запустить командный файл create_project.bat. Дождаться его завершения. Должно появиться приглашение командной строки vivado.
4. В командной строке Vivado набрать start_gui
5. Запустить генерацию файла конфигурации *.bit
6. После завершения генерации файла прошивки, скопировать его на Kraft

Генерация файла займет довольно много времени (~40 минут), поэтому можно, не дожидаясь завершения, переходить к пункту "Внесение изменений в проект". Затем, после завершения генерации прошивки, перейти к пункту "Запуск теста".

Запуск теста

Подготовка

1. Убедиться, что в системе обнаруживается хотя бы одна плата RC-47

```
$ cd /opt/RostaSDK/lin/utilities
$ ./rc47_find_devices
```

Пример вывода информации о найденной плате

```
/opt/RostaSDK/lin/utilities$  ./rc47_find_devices
Searching for RC47 boards
Found 1 board
Bus Address Chip ID
RC47 #0:::
System Chip 11:00.0 b007
Virtex-7 C0 0F:00.0 2000
Virtex-7 C1 0E:00.0 2000
Virtex-7 C2 12:00.0 2000
Virtex-7 C3 13:00.0 2000
PLX 8732 0C:00.0 8732
RC47 # 0 SN = 220003
```

Реконфигурация ПЛИС

1. Для того, чтобы реконфигурировать ПЛИС Virtex-7, с хост-компьютера, необходимо набрать в командной строке

```
$ cd /opt/RostaSDK/lin/utilities
$ ./rc47_config -b [board number] [chip_select] -f [path_to_file]
```

где board number - номер платы в системе, chip_select - любая комбинация C0, C1, C2, C3 или all, path_to_file - путь к файлу прошивки *.bit

Пример запуска утилиты:

```
/opt/RostaSDK/lin/utilities$   ./rc47_config -b 0 -v all -f
$ROSTA_SDK_DIR/bit-files/rc_pcie_base_2000_8.bit
Searching for RC47 boards
Found 1 board
                Bus Address     Chip ID
RC47 #0:::
      System Chip     0A:00.0       b007
      Virtex-7 C0     0C:00.0       2000
      Virtex-7 C1     0D:00.0       2000
      Virtex-7 C2     09:00.0       2000
      Virtex-7 C3     08:00.0       2000
Board selected: 0

-------------
Entered rc47_config_v7
Will now configure: C0 C1 C2 C3
C3 device diabled      08:00.0
C2 device diabled      09:00.0
C1 device diabled      0d:00.0
C0 device diabled      0c:00.0

load_bitstream: bit_file_path = $ROSTA_SDK_DIR/bit-files/rc_pcie_base_2000_8.bit
file length = 4994915
Start configuration...
Config End OK
Wait for link
Rescan PCI Express bus after reconfig...
Searching for RC47 boards
                Bus Address     Chip ID
RC47 #0:::
      System Chip     0A:00.0       b007
      Virtex-7 C0     0C:00.0       2000
      Virtex-7 C1     0D:00.0       2000
      Virtex-7 C2     09:00.0       2000
      Virtex-7 C3     08:00.0       2000
setpci -s c:00.0 68.L=00003820
setpci -s d:00.0 68.L=00003820
setpci -s 9:00.0 68.L=00003820
setpci -s 8:00.0 68.L=00003820
```

Запуск теста без изменений

1. Скопировать хост-программу /opt/RostaSDK/lin/training к себе в домашнюю директорию
2. Если тест не скомпилирован, скомпилировать его

```
$ cd $HOME_DIR/ddr_hls
$ make -B
```

3. Запустить тест, убедиться, что тест не выдает ошибок.

```
$ cd $HOME_DIR/training
$ ./add_test  -b [board] -v [chip_select]
```

где board - номер платы в системе , chip_select - C0, C1, C2 или C3

Пример запуска теста:

```
$ ./add_test -b 0 -v C1
Searching for RC47 boards
Found 1 board
                Bus Address     Chip ID
RC47 #0:::
      System Chip     0C:00.0       b007
      Virtex-7 C0     0E:00.0       2000
      Virtex-7 C1     0F:00.0       2000
      Virtex-7 C2     0B:00.0       2000
      Virtex-7 C3     0A:00.0       2000
      PLX 8732        08:00.0       8732

Selected:
Board number    :::     0
Virtex Chip     :::     C1

REG BASE TEST on iteration 0 completed successfully!
REG BASE TEST on iteration 1 completed successfully!
REG BASE TEST on iteration 2 completed successfully!
REG BASE TEST on iteration 3 completed successfully!
REG BASE TEST on iteration 4 completed successfully!
REG BASE TEST on iteration 5 completed successfully!
REG BASE TEST on iteration 6 completed successfully!
REG BASE TEST on iteration 7 completed successfully!
REG BASE TEST on iteration 8 completed successfully!
REG BASE TEST on iteration 9 completed successfully!
```

Алгоритм работы хост-программы:

1. Хост-программа записывает случайное число в переменную write_value.
2. Вызывается функция RD_WriteDeviceReg32 из библиотеки PCI Express API, которая инициирует запись центральным процессором в ПЛИС в регистр TestRegIn User IP
3. Внутри ПЛИС в блоке User IP значение TestRegIn инкрементируется на 1 и записывается в регистр TestRegOut и выставляется бит готовности.
4. Вызывается функция RD_ReadDeviceReg32 из библиотеки PCI Express API, которая инициирует чтение центральным процессором из ПЛИС из регистра TestRegOut User IP в переменную read_value.
5. Сравниваются значения, хранящиеся в переменных write_value и read_value.

Пункты 1-5 выполняются заданное количество итераций

Внесение изменений в проект

В данном пункте необходимо добавить новую функциональность в проект. Необходимо добавить возможность записи по двум адресам в регистры, а также возможность чтения по третьему адресу.

name	addr	val
reg1	addr_reg1	val1
reg2	addr_reg2	val2
regs	addr_regs	val1+val2

Для этого в файле user_ip.vhd необходимо внести соответствующие изменения.

Рекомендуемые значения адресов регистров:

addr	addr_val
addr_reg1	0x4C
addr_reg2	0x50
addr_regs	0x54

После внесения всех изменений необходимо выполнить все пункты из раздела "Запуск теста / Сборка FPGA проекта, генерация файла конфигурации ПЛИС".

Внесение изменений в хост-программу.

В хост-программе необходимо добавить запись в регистры по адресам addr_reg1, addr_reg2 , и чтение из регистра по адресу addr_regs по аналогии с примером. Также необходимо добавить проверку корректности результатов.

> Внимание! Адреса в проекте и в хост-программе должны отличаться на 0x100

Рекомендуемые значения адресов регистров:

addr	addr_val
addr_reg1	0x14C
addr_reg2	0x150
addr_regs	0x154

Запуск теста с новым проектом

Выполнить подпункты "Реконфигурация ПЛИС" и "Запуск HLS теста" из пункта "Запуск теста"

Убедиться в отсутствии ошибок.

> Результатом данной работы являются
>
> add_test.cpp
>
> user_ip.vhd
>
> Необходимо показать работу хост-программы преподавателю.