Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»(ИУ7)

НАПРАВЛЕНИЕ ПОДГОТОВКИ 09.03.04 «Программная инженерия»

О Т Ч Е Т
по лабораторной работе № 4

Название Методология разработки и верификации ускорителей вычислений на платформе Xilinx Alveo

Дисциплина Архитектура элекронно-вычислительных машин

Студент:	___________ подпись, дата	Козлова И. В. Фамилия, И.О.
Преподаватель:	___________ подпись, дата	Попов А. Ю. Фамилия, И. О.

Москва — 2021 г.

Оглавление

Цель работы

Изучение архитектуры гетерогенных вычислительных систем и технологии разработки ускорителей вычислений на базе ПЛИС фирмы Xilinx.

Для достижения поставленной цели необходимо выполнить следующие задачи:

- изучить основные сведения о платформе Xilinx Alveo U200;
- разработать RTL (Register Transfer Language, язык регистровых передач) описание ускорителя вычислений по индивидуальному варианту;
- выполнить генерацию ядра ускорителя;
- выполнить синтез и сборку бинарного модуля ускорителя;
- разработать и отладить тестирующее программное обеспечение на серверной хост-платформе;
- провести тесты работы ускорителя вычислений.

Основные теоретические сведения

В ходе лабораторной работы будет использован базовый шаблон так называемого RTL проекта VINC, который может быть создан в IDE Xilinx Vitis и САПР Xilinx Vivado. Шаблон VINC выполняет попарное сложение чисел исходного массива и сохраняет результаты во втором массиве. Проект VINC включает:

Проект ПО хоста, выполняющий инициализацию аппаратного ядра и его тестирование через OpenCL вызовы.

Синтезируемый RTL проект ядра ускорителя на языках Verilog и SystemVerilog.

Функциональный тест ускорителя VINC на языке SystemVerilog.

Все перечисленные проекты создаются автоматически посредством запуска мастера RTL проектов в IDE Xilinx Vitis, и могут далее модифицироваться как через тот же мастер, так и в ручном режиме в САПР Xilinx Vivado, или обычном текстовом редакторе. Ряд проектных процедур необходимо запустить из консоли ОС Linux.

Проект VINC представляет собой аппаратное устройство, связанное шиной AXI4 MM (Memory mapped) с DDR[i] памятью, и получающее настроечные параметры по интерфейсу AXI4 Lite от программного обеспечения хоста (см. рисунок 1). В рамках всей системы используется единое 64-х разрядное адресное пространство, в котором формируются адреса на всех AXI4 шинах.

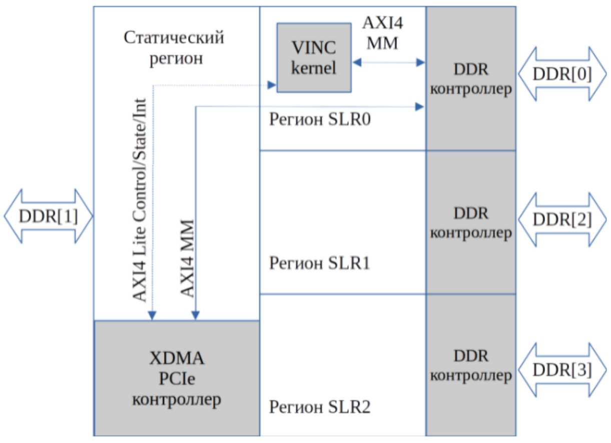


Рисунок 1 – Размещение проекта на ПЛИС xcu200-fsgd2104-2-e карты Alveo U200

В каждой карте U200 имеется возможность подключить ускоритель к любому DDR[i] контроллеру в том регионе, где будет размещен проект. Всего для пользователя доступны 3 динамических региона: SLR0,1,2, для которых выделены каналы локальной памяти DDR[0], DDR[2], DDR[3] соответственно. Вся подключенная память DDR[0..3] доступна со стороны статического региона, в котором размещена аппаратная часть XRT.

Память DDR[1] доступна для использования как в статическом регионе, так и в динамическом регионе SLR1.

Предполагается, что эта память может служить для организации эффективной подсистемы памяти ускорительной карты: буферизации данных, передаваемых между хост-системой и ускорителем.

Копии экранов моделирования исходного проекта VINC (исходная программа)

По умолчанию, в диаграмму (которую необходимо получить в приложении Vivado) добавлены сигналы шины AXI4 MM, представляющие собой 5 независимых каналов передачи сообщений, которые представлены в таблице 1.

Таблица 1 – Результаты замеров времени.

Канал передачи	Группы сигналов
Канал чтения адреса от ведущего к ведомому	m00_axi_ar*
Канал чтения данных от ведомого к ведущему	m00_axi_r*
Канал записи адреса записи от ведущего к ведомому	m00_axi_aw*
Канал запись данных от ведущего к ведомому	m00_axi_w*
Канал записи ответа от ведомого к ведущему	m00_axi_b*

Каналы позволяют сформировать конвейерные транзакции чтения и записи. Последовательность событий транзакции чтения можно представить следующим образом: ARVALID→ ARREADY→ RVALID→ RREADY.

Последовательность событий транзакции записи: AWVALID→ AWREADY → WVALID → WREADY → BVALID → BREADY.

На рисунке 2 приведена транзакция чтения данных вектора на шине AXI4 MM из DDR памяти.

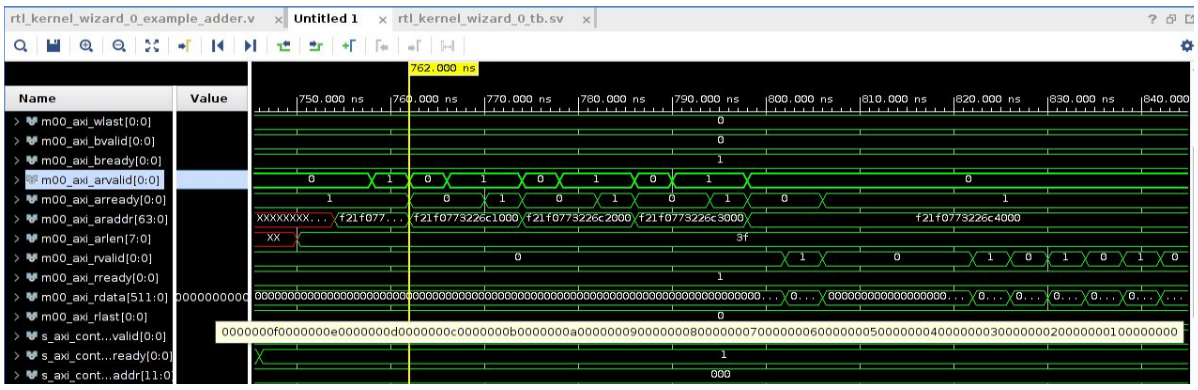


Рисунок 2 – Транзакция чтения данных вектора на шине AXI4 MM из DDR памяти

На рисунке 3 приведена транзакция записи результата инкремента данных на шине AXI4 MM.

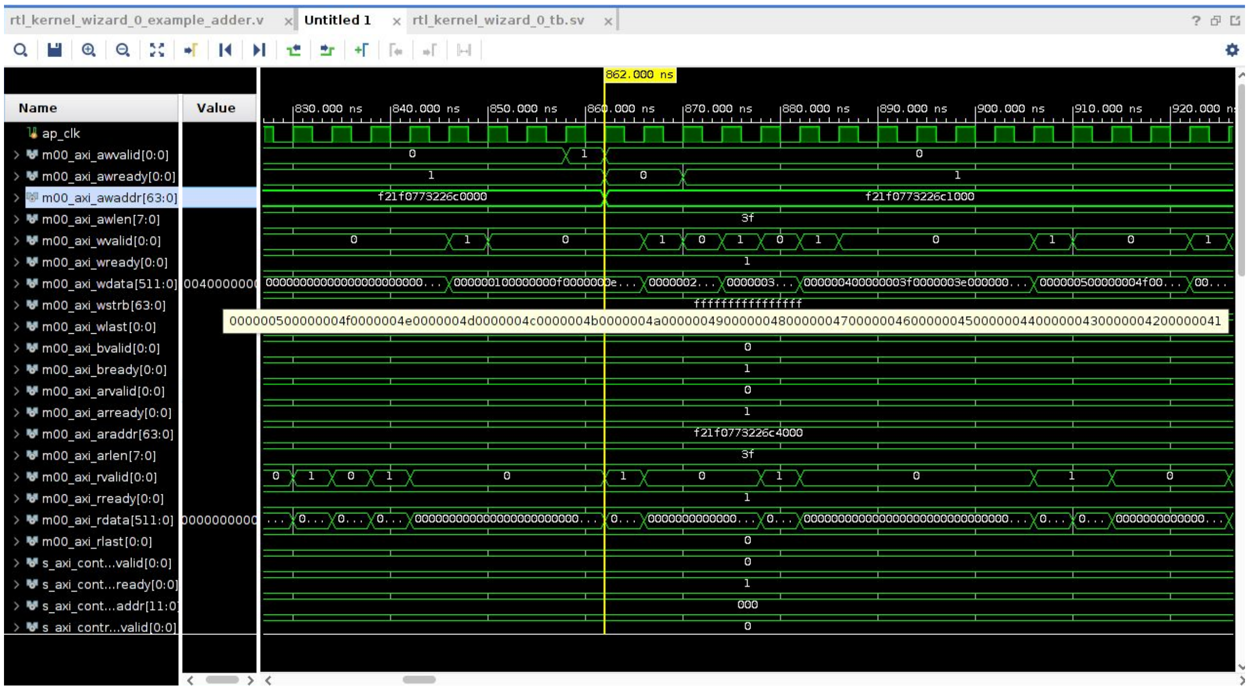
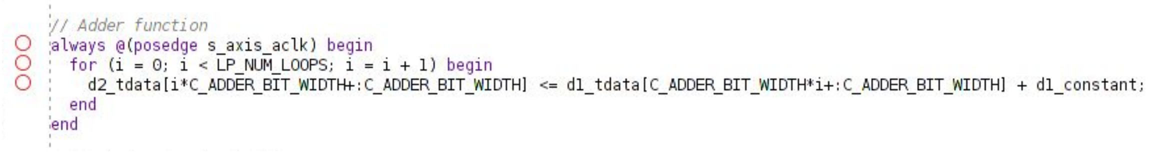


Рисунок 3 – Транзакция записи результата инкремента данных на шине AXI4 MM

На рисунке 4 приведен фрагмент кода модуля rtl_kernel_wizard_0_example_a с выполнением инкрементирования данных.



```
// Adder function
always @(posedge s_axis_aclk) begin
  for (i = 0; i < LP_NUM_LOOPS; i = i + 1) begin
    d2_tdata[i*C_ADDER_BIT_WIDTH+:C_ADDER_BIT_WIDTH] <= d1_tdata[C_ADDER_BIT_WIDTH*i+:C_ADDER_BIT_WIDTH] + d1_constant;
  end
end
```

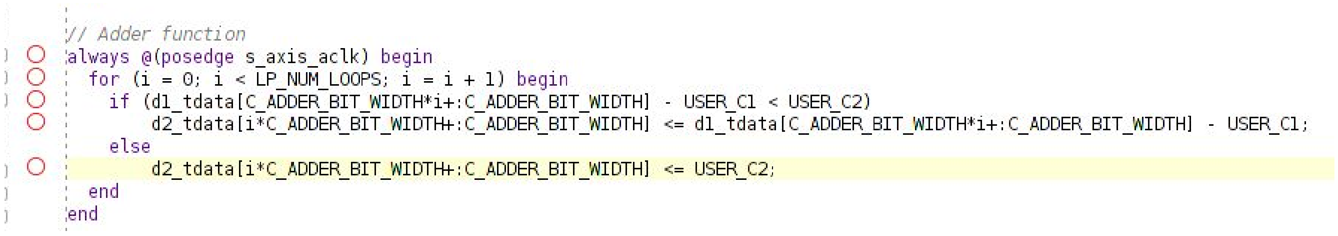
Рисунок 4 – Код модуля rtl_kernel_wizard_0_example_adder.v с выполнением инкрементирования данных

Копии экранов моделирования исходного проекта VINC (измененная программа)

В соответствии с вариантом 9 необходимо было изменить код проекта. Реализовать функцию 1

$$R[i] = min(A[i] - 4, 5) \tag{1}$$

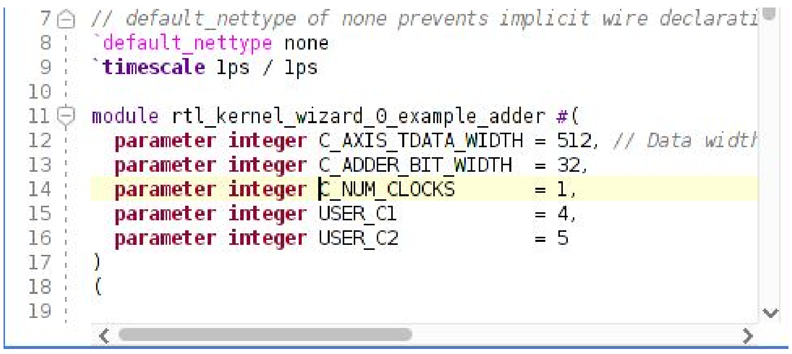
На рисунке 5 приведен код измененнной программы.



```verilog
// Adder function
always @(posedge s_axis_aclk) begin
  for (i = 0; i < LP_NUM_LOOPS; i = i + 1) begin
    if (d1_tdata[C_ADDER_BIT_WIDTH*i+:C_ADDER_BIT_WIDTH] - USER_C1 < USER_C2)
        d2_tdata[i*C_ADDER_BIT_WIDTH+:C_ADDER_BIT_WIDTH] <= d1_tdata[C_ADDER_BIT_WIDTH*i+:C_ADDER_BIT_WIDTH] - USER_C1;
    else
        d2_tdata[i*C_ADDER_BIT_WIDTH+:C_ADDER_BIT_WIDTH] <= USER_C2;
  end
end
```

Рисунок 5 – Измененный код модуля rtl_kernel_wizard_0_example_adder.v

Константы, которые используются в данном коде, представлены на рисунке 6.

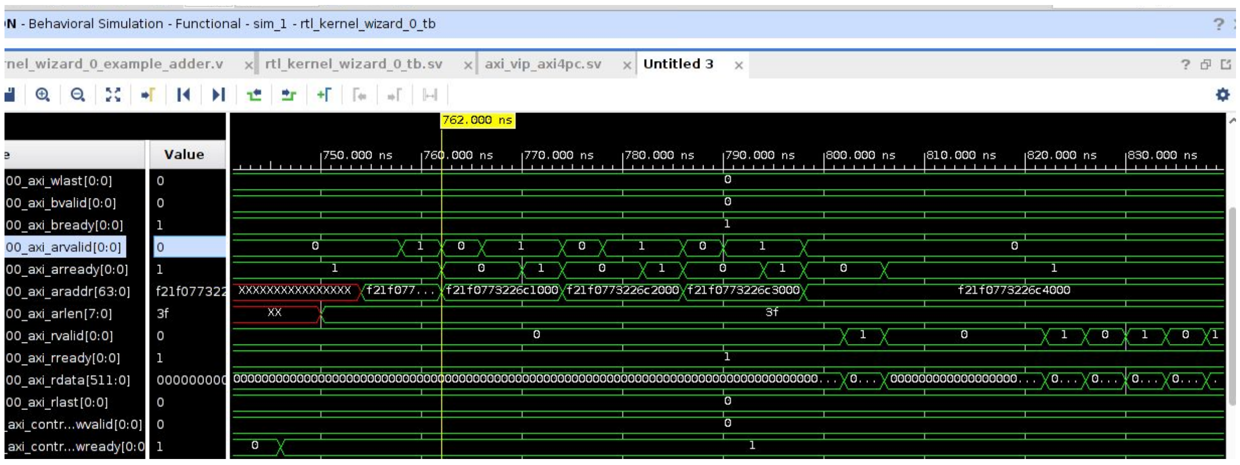


```verilog
// default_nettype of none prevents implicit wire declarati
`default_nettype none
`timescale 1ps / 1ps

module rtl_kernel_wizard_0_example_adder #(
  parameter integer C_AXIS_TDATA_WIDTH = 512, // Data widt
  parameter integer C_ADDER_BIT_WIDTH  = 32,
  parameter integer C_NUM_CLOCKS       = 1,
  parameter integer USER_C1            = 4,
  parameter integer USER_C2            = 5
)
(
```

Рисунок 6 – Константы

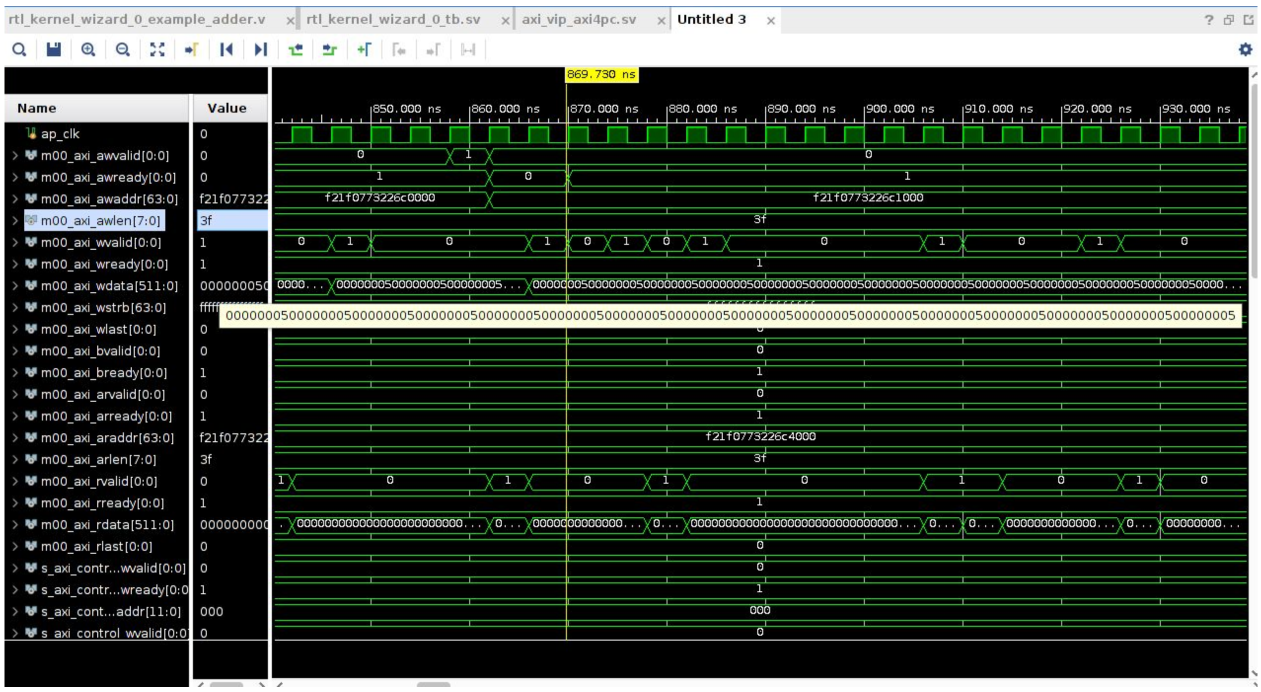
На рисунке 7 приведена транзакция чтения данных вектора на шине AXI4 MM из DDR памяти.



```
00_axi_wlast[0:0]        0
00_axi_bvalid[0:0]       0
00_axi_bready[0:0]       1
00_axi_arvalid[0:0]      0
00_axi_arready[0:0]      1
00_axi_araddr[63:0]      f21f077322
00_axi_arlen[7:0]        3f
00_axi_rvalid[0:0]       0
00_axi_rready[0:0]       1
00_axi_rdata[511:0]      000000000
00_axi_rlast[0:0]        0
axi_contr...wvalid[0:0]  0
axi_contr...wready[0:0   1
```

Рисунок 7 – Транзакция чтения данных вектора на шине AXI4 MM из DDR памяти

На рисунке 8 приведена транзакция записи результата инкремента данных на шине AXI4 MM.



```
ap_clk                   0
m00_axi_awvalid[0:0]     0
m00_axi_awready[0:0]     0
m00_axi_awaddr[63:0]     f21f077322
m00_axi_awlen[7:0]       3f
m00_axi_wvalid[0:0]      1
m00_axi_wready[0:0]      1
m00_axi_wdata[511:0]     000000050
m00_axi_wstrb[63:0]      ffff
m00_axi_wlast[0:0]       0
m00_axi_bvalid[0:0]      0
m00_axi_bready[0:0]      1
m00_axi_arvalid[0:0]     0
m00_axi_arready[0:0]     1
m00_axi_araddr[63:0]     f21f077322
m00_axi_arlen[7:0]       3f
m00_axi_rvalid[0:0]      0
m00_axi_rready[0:0]      1
m00_axi_rdata[511:0]     000000000
m00_axi_rlast[0:0]       0
s_axi_contr...wvalid[0:0]  0
s_axi_contr...wready[0:0   1
s_axi_cont...addr[11:0]  000
s_axi_control_wvalid[0:0   0

0000000500000005000000050000000500000005000000050000000500000005000000050000000500000005000000050000000500000005000000050000000500000005
```

Рисунок 8 – Транзакция записи результата инкремента данных на шине AXI4 MM

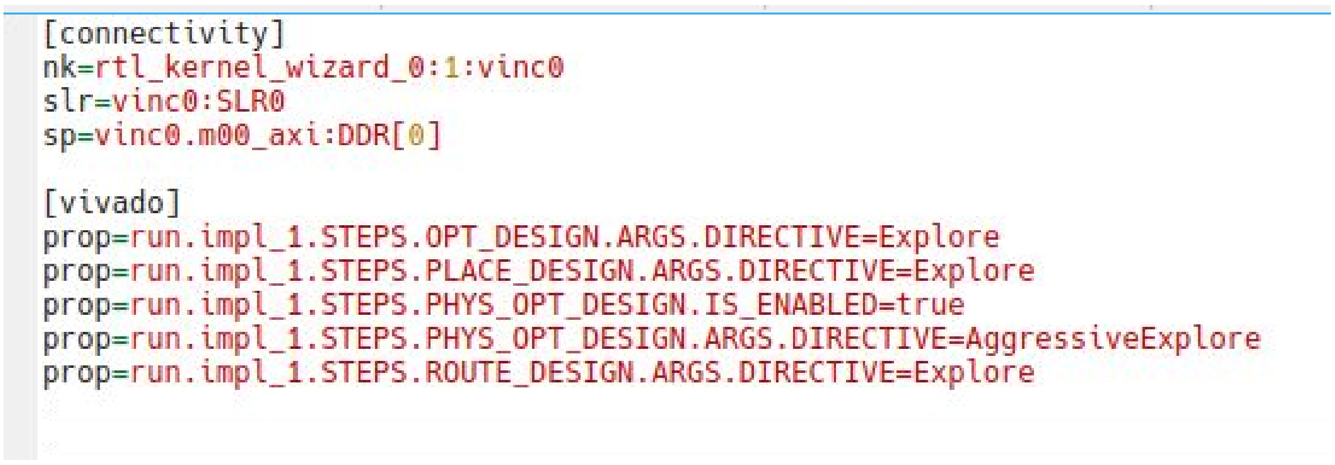
Сборка проекта

Для сборки проекта необходимо было написать конфигурационный файл.

В конфигурационом файле указывается основная информация для работы компилятора v++:

1. Количество и условные имена экземпляров ядер.
2. Тактовая частота работы ядра.
3. Для каждого ядра: выбор области SLR (SLR[0..2]), выбор DDR (DDR[0..3]) памяти, выбор высокопроизводительной памяти PLRAM(PLRAM[0,1,2]).
4. Параметры синтеза и оптимизации проекта.

На рисунке 9 представлен конфигурационный файл для сборки проекта.



```
[connectivity]
nk=rtl_kernel_wizard_0:1:vinc0
slr=vinc0:SLR0
sp=vinc0.m00_axi:DDR[0]

[vivado]
prop=run.impl_1.STEPS.OPT_DESIGN.ARGS.DIRECTIVE=Explore
prop=run.impl_1.STEPS.PLACE_DESIGN.ARGS.DIRECTIVE=Explore
prop=run.impl_1.STEPS.PHYS_OPT_DESIGN.IS_ENABLED=true
prop=run.impl_1.STEPS.PHYS_OPT_DESIGN.ARGS.DIRECTIVE=AggressiveExplore
prop=run.impl_1.STEPS.ROUTE_DESIGN.ARGS.DIRECTIVE=Explore
```

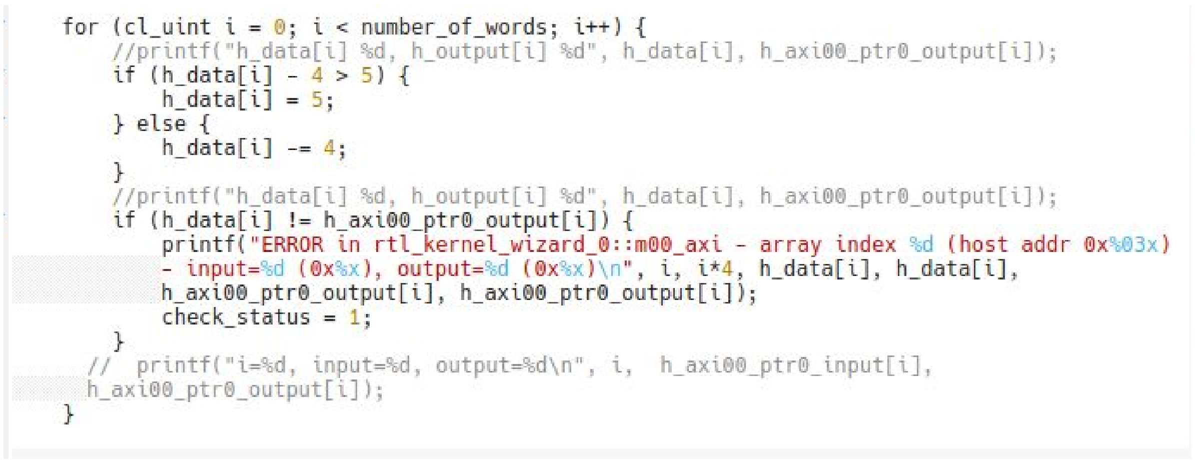
Рисунок 9 – Конфигурационный файл

Содержимое файлов v++*.log и *.xclbin.info. приведено в приложениях.

Тестирование

Для того, чтобы запустить тесты, необходимо изменить условие проверки в автоматически созданном программном модуле host_example.cpp.

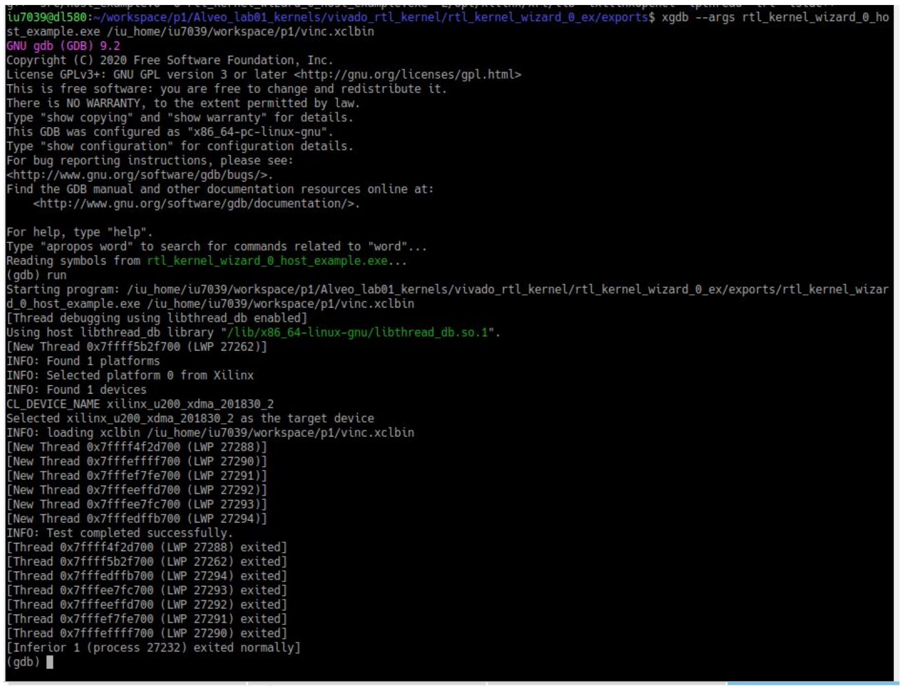
Часть кода модуля host_example.cpp приведена на рисунке 10. Было изменено условие проверки, на проверку, соответствующую моему варианту.



```
for (cl_uint i = 0; i < number_of_words; i++) {
    //printf("h_data[i] %d, h_output[i] %d", h_data[i], h_axi00_ptr0_output[i]);
    if (h_data[i] - 4 > 5) {
        h_data[i] = 5;
    } else {
        h_data[i] -= 4;
    }
    //printf("h_data[i] %d, h_output[i] %d", h_data[i], h_axi00_ptr0_output[i]);
    if (h_data[i] != h_axi00_ptr0_output[i]) {
        printf("ERROR in rtl_kernel_wizard_0::m00_axi - array index %d (host addr 0x%03x)
        - input=%d (0x%x), output=%d (0x%x)\n", i, i*4, h_data[i], h_data[i],
        h_axi00_ptr0_output[i], h_axi00_ptr0_output[i]);
        check_status = 1;
    }
  //  printf("i=%d, input=%d, output=%d\n", i,  h_axi00_ptr0_input[i],
  h_axi00_ptr0_output[i]);
}
```

Рисунок 10 – host_example.cpp

Тесты запускались с помощью утилиты xgdb. Результаты тестирования приведены на рисунке 11.



```
iu7039@dl580:~/workspace/p1/Alveo_lab01_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports$ xgdb --args rtl_kernel_wizard_0_ho
st_example.exe /iu_home/iu7039/workspace/p1/vinc.xclbin
GNU gdb (GDB) 9.2
Copyright (C) 2020 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-pc-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from rtl_kernel_wizard_0_host_example.exe...
(gdb) run
Starting program: /iu_home/iu7039/workspace/p1/Alveo_lab01_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizar
d_0_host_example.exe /iu_home/iu7039/workspace/p1/vinc.xclbin
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
[New Thread 0x7ffff5b2f700 (LWP 27262)]
INFO: Found 1 platforms
INFO: Selected platform 0 from Xilinx
INFO: Found 1 devices
CL_DEVICE_NAME xilinx_u200_xdma_201830_2
Selected xilinx_u200_xdma_201830_2 as the target device
INFO: loading xclbin /iu_home/iu7039/workspace/p1/vinc.xclbin
[New Thread 0x7ffff4f2d700 (LWP 27288)]
[New Thread 0x7fffeffff700 (LWP 27290)]
[New Thread 0x7fffef7fe700 (LWP 27291)]
[New Thread 0x7fffeeffd700 (LWP 27292)]
[New Thread 0x7fffee7fc700 (LWP 27293)]
[New Thread 0x7fffedffb700 (LWP 27294)]
INFO: Test completed successfully.
[Thread 0x7ffff4f2d700 (LWP 27288) exited]
[Thread 0x7ffff5b2f700 (LWP 27262) exited]
[Thread 0x7fffedffb700 (LWP 27294) exited]
[Thread 0x7fffee7fc700 (LWP 27293) exited]
[Thread 0x7fffeeffd700 (LWP 27292) exited]
[Thread 0x7fffef7fe700 (LWP 27291) exited]
[Thread 0x7fffeffff700 (LWP 27290) exited]
[Inferior 1 (process 27232) exited normally]
(gdb) 
```

Рисунок 11 – Тестирование

По результатам можно увидеть, что все тесты выполнены успешно.

Контрольные вопросы

1. Назовите преимущества и недостатки XDMA и QDMA платформ.

Преимущества QDMA:

- позволяет передавать поток данных непосредственно в логику FPGA параллельно с их обработкой.
- предоставляет разработчикам прямое потоковое соединение с низкой задержкой между хостом и ядрами.
- включает высокопроизводительный DMA, который использует несколько очередей, оптимизированных как для передачи данных с высокой пропускной способностью, так и для передачи данных с большим количеством пакетов.

Недостатки XDMA:

- требует, чтобы данные сначала были полностью перемещены из памяти хоста в память FPGA (DDRx4 DIMM или PLRAM), прежде чем логика FPGA сможет начать обработку данных, что влияет на задержку на запуска задачи.

2. Назовите последовательность действий, необходимых для инициализации ускорителя со стороны хост-системы.

1. Хост получает все платформы.
2. Хост выбирает имя платформы Xilinx.
3. Хост получает Id устройства.
4. Хост получает информацию об устройстве.
5. Создается контекст для переменных.
6. Создается команда для устройста-ускорителя.

3. Какова процедура запуска задания на исполнения в ускорительном ядре VINC.

1. Данные из .xclbin копируются из ОЗУ в локальную память ускорителя посредством DMA.
2. В памяти устройства-ускорителя создается исполняемый файл.
3. Те данные, которые подлежат обработке, копируются из ОЗУ в локальную память усокрителя посредством DMA.
4. Указываются необходимые параметры и запускается программа на ускорителе.
5. В конце выполняется чтение готовых данных.

4. Опишите процесс линковки на основании содержимого файла v++_*.log.

1. Анализ профиля устройства. Анализ конфигурационного файла. Поиск необходимых интерфейсов.
2. FPGA linking synthesized kernels to platform
3. FPGA logic optimization (оптимизация логики ПЛИС) для минимизации задержки.
4. FPGA logic placement (размещение логики ПЛИС, то есть выбор конкретного мета для определенного логического блока).
5. FPGA routing (маршрутизация ПЛИС)
6. FPGA bitstream generation (генерация битового потока ПЛИС, то есть генерация файла [*.xclbin]).

Заключение

Изучены архитектуры гетерогенных вычислительных систем и технологии разработки ускорителей вычислений на базе ПЛИС фирмы Xilinx.

Были выполнены следующие задачи:

- изучены основные сведения о платформе Xilinx Alveo U200;
- разработано RTL (Register Transfer Language, язык регистровых передач) описание ускорителя вычислений по индивидуальному варианту;
- выполнена генерация ядра ускорителя;
- выполнены синтез и сборка бинарного модуля ускорителя;
- разработано и отлажено тестирующее программное обеспечение на серверной хост-платформе;
- проведены тесты работы ускорителя вычислений.

Поставленная цель достигнута.

Приложение 1

Листинг 1 – Содержимое файла host_example.cpp

```
// This is a generated file. Use and modify at your own risk.
////////////////////////////////////////////////////////////////////////////////

/*******************************************************************************
Vendor: Xilinx
Associated Filename: main.c
#Purpose: This example shows a basic vector add +1 (constant) by manipulating
#          memory inplace.
*******************************************************************************/

#include <fcntl.h>
#include <stdio.h>
#include <iostream>
#include <stdlib.h>
#include <string.h>
#include <math.h>
#ifdef _WINDOWS
#include <io.h>
#else
#include <unistd.h>
#include <sys/time.h>
#endif
#include <assert.h>
#include <stdbool.h>
#include <sys/types.h>
#include <sys/stat.h>
#include <CL/opencl.h>
#include <CL/cl_ext.h>
#include "xclhal2.h"

////////////////////////////////////////////////////////////////////////////////

#define NUM_WORKGROUPS (1)
#define WORKGROUP_SIZE (256)
#define MAX_LENGTH 8192
#define MEM_ALIGNMENT 4096
#if defined(VITIS_PLATFORM) && !defined(TARGET_DEVICE)
#define STR_VALUE(arg)      #arg
#define GET_STRING(name) STR_VALUE(name)
#define TARGET_DEVICE GET_STRING(VITIS_PLATFORM)
#endif

////////////////////////////////////////////////////////////////////////////////

cl_uint load_file_to_memory(const char *filename, char **result)
{
    cl_uint size = 0;
    FILE *f = fopen(filename, "rb");
    if (f == NULL) {
        *result = NULL;
        return -1; // -1 means file opening fail
    }
    fseek(f, 0, SEEK_END);
    size = ftell(f);
    fseek(f, 0, SEEK_SET);
    *result = (char *)malloc(size+1);
    if (size != fread(*result, sizeof(char), size, f)) {
        free(*result);
        return -2; // -2 means file reading fail
    }
    fclose(f);
    (*result)[size] = 0;
    return size;
}

int main(int argc, char** argv)
{

    cl_int err;                            // error code returned from api calls
    cl_uint check_status = 0;
    const cl_uint number_of_words = 4096; // 16KB of data


    cl_platform_id platform_id;          // platform id
    cl_device_id device_id;              // compute device id
    cl_context context;                 // compute context
    cl_command_queue commands;          // compute command queue
    cl_program program;                 // compute programs
    cl_kernel kernel;                   // compute kernel

    cl_uint* h_data;                                // host memory for input vector
    char cl_platform_vendor[1001];
    char target_device_name[1001] = TARGET_DEVICE;

    cl_uint* h_axi00_ptr0_output = (cl_uint*)aligned_alloc(MEM_ALIGNMENT,MAX_LENGTH * sizeof(cl_uint*)); // host memory for output vector
    cl_mem d_axi00_ptr0;                            // device memory used for a vector

    if (argc != 2) {
        printf("Usage: %s xclbin\n", argv[0]);
        return EXIT_FAILURE;
    }

    // Fill our data sets with pattern
    h_data = (cl_uint*)aligned_alloc(MEM_ALIGNMENT,MAX_LENGTH * sizeof(cl_uint*));
    for(cl_uint i = 0; i < MAX_LENGTH; i++) {
        h_data[i]  = i;
        h_axi00_ptr0_output[i] = 0;

    }

    // Get all platforms and then select Xilinx platform
    cl_platform_id platforms[16];       // platform id
    cl_uint platform_count;
    cl_uint platform_found = 0;
    err = clGetPlatformIDs(16, platforms, &platform_count);
    if (err != CL_SUCCESS) {
        printf("ERROR: Failed to find an OpenCL platform!\n");
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    }
    printf("INFO: Found %d platforms\n", platform_count);

    // Find Xilinx Plaftorm
    for (cl_uint iplat=0; iplat<platform_count; iplat++) {
        err = clGetPlatformInfo(platforms[iplat], CL_PLATFORM_VENDOR, 1000, (void *)cl_platform_vendor,
            NULL);
        if (err != CL_SUCCESS) {
            printf("ERROR: clGetPlatformInfo(CL_PLATFORM_VENDOR) failed!\n");
            printf("ERROR: Test failed\n");
            return EXIT_FAILURE;
        }
        if (strcmp(cl_platform_vendor, "Xilinx") == 0) {
            printf("INFO: Selected platform %d from %s\n", iplat, cl_platform_vendor);
            platform_id = platforms[iplat];
            platform_found = 1;
        }
    }
    if (!platform_found) {
        printf("ERROR: Platform Xilinx not found. Exit.\n");
        return EXIT_FAILURE;
    }

    // Get Accelerator compute device
    cl_uint num_devices;
    cl_uint device_found = 0;
    cl_device_id devices[16];  // compute device id
    char cl_device_name[1001];
    err = clGetDeviceIDs(platform_id, CL_DEVICE_TYPE_ACCELERATOR, 16, devices, &num_devices);
    printf("INFO: Found %d devices\n", num_devices);
    if (err != CL_SUCCESS) {
        printf("ERROR: Failed to create a device group!\n");
        printf("ERROR: Test failed\n");
        return -1;
    }

    //iterate all devices to select the target device.
    for (cl_uint i=0; i<num_devices; i++) {
        err = clGetDeviceInfo(devices[i], CL_DEVICE_NAME, 1024, cl_device_name, 0);
        if (err != CL_SUCCESS) {
            printf("ERROR: Failed to get device name for device %d!\n", i);
            printf("ERROR: Test failed\n");
            return EXIT_FAILURE;
        }
        printf("CL_DEVICE_NAME %s\n", cl_device_name);
        if(strcmp(cl_device_name, target_device_name) == 0) {
            device_id = devices[i];
            device_found = 1;
            printf("Selected %s as the target device\n", cl_device_name);
        }
    }

    if (!device_found) {
        printf("ERROR:Target device %s not found. Exit.\n", target_device_name);
        return EXIT_FAILURE;
    }

    // Create a compute context
    //
    context = clCreateContext(0, 1, &device_id, NULL, NULL, &err);
    if (!context) {
        printf("ERROR: Failed to create a compute context!\n");
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    }

    // Create a command commands
    commands = clCreateCommandQueue(context, device_id, CL_QUEUE_PROFILING_ENABLE |
        CL_QUEUE_OUT_OF_ORDER_EXEC_MODE_ENABLE, &err);
    if (!commands) {
        printf("ERROR: Failed to create a command commands!\n");
        printf("ERROR: code %i\n",err);
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    }

    cl_int status;

    // Create Program Objects
    // Load binary from disk
    unsigned char *kernelbinary;
    char *xclbin = argv[1];

    //------------------------------------------------------------------------------
    // xclbin
    //------------------------------------------------------------------------------
    printf("INFO: loading xclbin %s\n", xclbin);
    cl_uint n_i0 = load_file_to_memory(xclbin, (char **) &kernelbinary);
    if (n_i0 < 0) {
        printf("ERROR: failed to load kernel from xclbin: %s\n", xclbin);
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    }

    size_t n0 = n_i0;

    // Create the compute program from offline
    program = clCreateProgramWithBinary(context, 1, &device_id, &n0,
    (const unsigned char **) &kernelbinary, &status, &err);
    free(kernelbinary);

    if ((!program) || (err!=CL_SUCCESS)) {
        printf("ERROR: Failed to create compute program from binary %d!\n", err);
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    }


    // Build the program executable
    //
    err = clBuildProgram(program, 0, NULL, NULL, NULL, NULL);
    if (err != CL_SUCCESS) {
        size_t len;
        char buffer[2048];

        printf("ERROR: Failed to build program executable!\n");
        clGetProgramBuildInfo(program, device_id, CL_PROGRAM_BUILD_LOG, sizeof(buffer), buffer, &len);
        printf("%s\n", buffer);
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    }

    // Create the compute kernel in the program we wish to run
    //
    kernel = clCreateKernel(program, "rtl_kernel_wizard_0", &err);
    if (!kernel || err != CL_SUCCESS) {
        printf("ERROR: Failed to create compute kernel!\n");
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    }

    // Create structs to define memory bank mapping
    cl_mem_ext_ptr_t mem_ext;
    mem_ext.obj = NULL;
    mem_ext.param = kernel;


    mem_ext.flags = 1;
    d_axi00_ptr0 = clCreateBuffer(context,  CL_MEM_READ_WRITE | CL_MEM_EXT_PTR_XILINX,  sizeof(cl_uint) *
        number_of_words, &mem_ext, &err);
    if (err != CL_SUCCESS) {
        std::cout << "Return␣code␣for␣clCreateBuffer␣flags=" << mem_ext.flags << ":␣" << err << std::endl;
    }


    if (!(d_axi00_ptr0)) {
        printf("ERROR:␣Failed␣to␣allocate␣device␣memory!\n");
        printf("ERROR:␣Test␣failed\n");
        return EXIT_FAILURE;
    }


    err = clEnqueueWriteBuffer(commands, d_axi00_ptr0, CL_TRUE, 0, sizeof(cl_uint) * number_of_words,
        h_data, 0, NULL, NULL);
    if (err != CL_SUCCESS) {
        printf("ERROR:␣Failed␣to␣write␣to␣source␣array␣h_data:␣d_axi00_ptr0:␣%d!\n", err);
        printf("ERROR:␣Test␣failed\n");
        return EXIT_FAILURE;
    }


    // Set the arguments to our compute kernel
    // cl_uint vector_length = MAX_LENGTH;
    err = 0;
    cl_uint d_num = 0;
    err |= clSetKernelArg(kernel, 0, sizeof(cl_uint), &d_num); // Not used in example RTL logic.
    err |= clSetKernelArg(kernel, 1, sizeof(cl_mem), &d_axi00_ptr0);

    if (err != CL_SUCCESS) {
        printf("ERROR:␣Failed␣to␣set␣kernel␣arguments!␣%d\n", err);
        printf("ERROR:␣Test␣failed\n");
        return EXIT_FAILURE;
    }

    size_t global[1];
    size_t local[1];
    // Execute the kernel over the entire range of our 1d input data set
    // using the maximum number of work group items for this device

    global[0] = 1;
    local[0] = 1;
    err = clEnqueueNDRangeKernel(commands, kernel, 1, NULL, (size_t*)&global, (size_t*)&local, 0, NULL,
        NULL);
    if (err) {
        printf("ERROR:␣Failed␣to␣execute␣kernel!␣%d\n", err);
        printf("ERROR:␣Test␣failed\n");
        return EXIT_FAILURE;
    }

    clFinish(commands);


    // Read back the results from the device to verify the output
    //
    cl_event readevent;

    err = 0;
    err |= clEnqueueReadBuffer( commands, d_axi00_ptr0, CL_TRUE, 0, sizeof(cl_uint) * number_of_words,
        h_axi00_ptr0_output, 0, NULL, &readevent );


    if (err != CL_SUCCESS) {
        printf("ERROR:␣Failed␣to␣read␣output␣array!␣%d\n", err);
        printf("ERROR:␣Test␣failed\n");
        return EXIT_FAILURE;
    }
    clWaitForEvents(1, &readevent);
    // Check Results

    for (cl_uint i = 0; i < number_of_words; i++) {
        //printf("h_data[i] %d, h_output[i] %d", h_data[i], h_axi00_ptr0_output[i]);
        if (h_data[i] - 4 > 5) {
            h_data[i] = 5;
        } else {
            h_data[i] -= 4;
        }
        //printf("h_data[i] %d, h_output[i] %d", h_data[i], h_axi00_ptr0_output[i]);
        if (h_data[i] != h_axi00_ptr0_output[i]) {
            printf("ERROR in rtl_kernel_wizard_0::m00_axi - array index %d (host addr 0x%03x) - input=%d (0x%x), output=%d (0x%x)\n", i, i*4, h_data[i], h_data[i], h_axi00_ptr0_output[i], h_axi00_ptr0_output[i]);
            check_status = 1;
        }
        //  printf("i=%d, input=%d, output=%d\n", i,  h_axi00_ptr0_input[i], h_axi00_ptr0_output[i]);
    }



    //------------------------------------------------------------------------------
    // Shutdown and cleanup
    //------------------------------------------------------------------------------
    clReleaseMemObject(d_axi00_ptr0);
    free(h_axi00_ptr0_output);



    free(h_data);
    clReleaseProgram(program);
    clReleaseKernel(kernel);
    clReleaseCommandQueue(commands);
    clReleaseContext(context);

    if (check_status) {
        printf("ERROR: Test failed\n");
        return EXIT_FAILURE;
    } else {
        printf("INFO: Test completed successfully.\n");
        return EXIT_SUCCESS;
    }


} // end of main
```

Приложение 2

Листинг 2 – Содержимое log-файла

```
INFO: [v++ 60-1306] Additional information associated with this v++ link can be found at:
Reports: /iu_home/iu7039/workspace/p1/_x/reports/link
Log files: /iu_home/iu7039/workspace/p1/_x/logs/link
INFO: [v++ 60-1548] Creating build summary session with primary output /iu_home/iu7039/workspace/p1/vinc.xclbin.link_summary, at Sat Nov 20 11:42:41 2021
INFO: [v++ 60-1316] Initiating connection to rulecheck server, at Sat Nov 20 11:42:42 2021
INFO: [v++ 60-1315] Creating rulecheck session with output '/iu_home/iu7039/workspace/p1/_x/reports/link/v++_link_vinc_guidance.html', at Sat Nov 20 11:43:00 2021
INFO: [v++ 60-895]   Target platform: /opt/xilinx/platforms/xilinx_u200_xdma_201830_2/xilinx_u200_xdma_201830_2.xpfm
INFO: [v++ 60-1578]   This platform contains Device Support Archive '/opt/xilinx/platforms/xilinx_u200_xdma_201830_2/hw/xilinx_u200_xdma_201830_2.dsa'
INFO: [v++ 74-74] Compiler Version string: 2020.2
INFO: [v++ 60-1302] Platform 'xilinx_u200_xdma_201830_2.xpfm' has been explicitly enabled for this release.
INFO: [v++ 60-629] Linking for hardware target
INFO: [v++ 60-423]   Target device: xilinx_u200_xdma_201830_2
INFO: [v++ 60-1332] Run 'run_link' status: Not started
INFO: [v++ 60-1443] [11:43:51] Run run_link: Step system_link: Started
INFO: [v++ 60-1453] Command Line: system_link --xo /iu_home/iu7039/workspace/p1/Alveo_lab01_kernels/src/vitis_rtl_kernel/rtl_kernel_wizard_0/rtl_kernel_wizard_0.xo --config /iu_home/iu7039/workspace/p1/_x/link/int/syslinkConfig.ini --xpfm /opt/xilinx/platforms/xilinx_u200_xdma_201830_2/xilinx_u200_xdma_201830_2.xpfm --target hw --output_dir /iu_home/iu7039/workspace/p1/_x/link/int --temp_dir /iu_home/iu7039/workspace/p1/_x/link/sys_link
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link
INFO: [SYSTEM_LINK 60-1316] Initiating connection to rulecheck server, at Sat Nov 20 11:44:04 2021
INFO: [SYSTEM_LINK 82-70] Extracting xo v3 file /iu_home/iu7039/workspace/p1/Alveo_lab01_kernels/src/vitis_rtl_kernel/rtl_kernel_wizard_0/rtl_kernel_wizard_0.xo
INFO: [SYSTEM_LINK 82-53] Creating IP database /iu_home/iu7039/workspace/p1/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml
INFO: [SYSTEM_LINK 82-38] [11:44:06] build_xd_ip_db started: /data/Xilinx/Vitis/2020.2/bin/build_xd_ip_db -ip_search 0  -sds-pf /iu_home/iu7039/workspace/p1/_x/link/sys_link/xilinx_u200_xdma_201830_2.hpfm -clkid 0 -ip /iu_home/iu7039/workspace/p1/_x/link/sys_link/iprepo/mycompany_com_kernel_rtl_kernel_wizard_0_1_0,rtl_kernel_wizard_0 -o /iu_home/iu7039/workspace/p1/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml
INFO: [SYSTEM_LINK 82-37] [11:44:36] build_xd_ip_db finished successfully
Time (s): cpu = 00:00:31 ; elapsed = 00:00:30 . Memory (MB): peak = 1557.891 ; gain = 0.000 ; free physical = 264561 ; free virtual = 290855
INFO: [SYSTEM_LINK 82-51] Create system connectivity graph
INFO: [SYSTEM_LINK 82-102] Applying explicit connections to the system connectivity graph: /iu_home/iu7039/workspace/p1/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml
INFO: [SYSTEM_LINK 82-38] [11:44:36] cfgen started: /data/Xilinx/Vitis/2020.2/bin/cfgen  -nk rtl_kernel_wizard_0:1:vinc0 -slr vinc0:SLR0 -sp vinc0.m00_axi:DDR[0] -dmclkid 0 -r /iu_home/iu7039/workspace/p1/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml -o /iu_home/iu7039/workspace/p1/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml
INFO: [CFGEN 83-0] Kernel Specs:
INFO: [CFGEN 83-0]   kernel: rtl_kernel_wizard_0, num: 1  {vinc0}
INFO: [CFGEN 83-0] Port Specs:
INFO: [CFGEN 83-0]   kernel: vinc0, k_port: m00_axi, sptag: DDR[0]
INFO: [CFGEN 83-0] SLR Specs:
INFO: [CFGEN 83-0]   instance: vinc0, SLR: SLR0
INFO: [CFGEN 83-2228] Creating mapping for argument vinc0.axi00_ptr0 to DDR[0] for directive vinc0.m00_axi:DDR[0]
INFO: [SYSTEM_LINK 82-37] [11:45:01] cfgen finished successfully
Time (s): cpu = 00:00:25 ; elapsed = 00:00:25 . Memory (MB): peak = 1557.891 ; gain = 0.000 ; free physical = 264560 ; free virtual = 290846
INFO: [SYSTEM_LINK 82-52] Create top-level block diagram
INFO: [SYSTEM_LINK 82-38] [11:45:01] cf2bd started: /data/Xilinx/Vitis/2020.2/bin/cf2bd  --linux --trace_buffer 1024 --input_file /iu_home/iu7039/workspace/p1/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml --ip_db /iu_home/iu7039/workspace/p1/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml --cf_name dr --working_dir /iu_home/iu7039/workspace/p1/_x/link/sys_link/_sysl/.xsd --temp_dir /iu_home/iu7039/workspace/p1/_x/link/sys_link --output_dir /iu_home/iu7039/workspace/p1/_x/link/int --target_bd pfm_dynamic.bd
INFO: [CF2BD 82-31] Launching cf2xd: cf2xd -linux -trace-buffer 1024 -i /iu_home/iu7039/workspace/p1/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml -r /iu_home/iu7039/workspace/p1/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml -o dr.xml
INFO: [CF2BD 82-28] cf2xd finished successfully
INFO: [CF2BD 82-31] Launching cf_xsd: cf_xsd -disable-address-gen -bd pfm_dynamic.bd -dn dr -dp /iu_home/iu7039/workspace/p1/_x/link/sys_link/_sysl/.xsd
INFO: [CF2BD 82-28] cf_xsd finished successfully
INFO: [SYSTEM_LINK 82-37] [11:45:16] cf2bd finished successfully
Time (s): cpu = 00:00:13 ; elapsed = 00:00:14 . Memory (MB): peak = 1557.891 ; gain = 0.000 ; free physical = 264525 ; free virtual = 290816
INFO: [v++ 60-1441] [11:45:16] Run run_link: Step system_link: Completed
```

```
Time (s): cpu = 00:01:24 ; elapsed = 00:01:25 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
    physical = 264582 ; free virtual = 290868
INFO: [v++ 60-1443] [11:45:16] Run run_link: Step cf2sw: Started
INFO: [v++ 60-1453] Command Line: cf2sw -sdsl /iu_home/iu7039/workspace/p1/_x/link/int/sdsl.dat -rtd /
    iu_home/iu7039/workspace/p1/_x/link/int/cf2sw.rtd -nofilter /iu_home/iu7039/workspace/p1/_x/link/int/
    cf2sw_full.rtd -xclbin /iu_home/iu7039/workspace/p1/_x/link/int/xclbin_orig.xml -o /iu_home/iu7039/
    workspace/p1/_x/link/int/xclbin_orig.1.xml
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link
INFO: [v++ 60-1441] [11:45:33] Run run_link: Step cf2sw: Completed
Time (s): cpu = 00:00:16 ; elapsed = 00:00:17 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
    physical = 264551 ; free virtual = 290847
INFO: [v++ 60-1443] [11:45:33] Run run_link: Step rtd2_system_diagram: Started
INFO: [v++ 60-1453] Command Line: rtd2SystemDiagram
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link
INFO: [v++ 60-1441] [11:45:40] Run run_link: Step rtd2_system_diagram: Completed
Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:08 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
    physical = 263964 ; free virtual = 290259
INFO: [v++ 60-1443] [11:45:40] Run run_link: Step vpl: Started
INFO: [v++ 60-1453] Command Line: vpl -t hw -f xilinx_u200_xdma_201830_2 --remote_ip_cache /iu_home/iu7039
    /workspace/p1/.ipcache --output_dir /iu_home/iu7039/workspace/p1/_x/link/int --log_dir /iu_home/iu7039
    /workspace/p1/_x/logs/link --report_dir /iu_home/iu7039/workspace/p1/_x/reports/link --config /iu_home
    /iu7039/workspace/p1/_x/link/int/vplConfig.ini -k /iu_home/iu7039/workspace/p1/_x/link/int/kernel_info
    .dat --webtalk_flag Vitis --temp_dir /iu_home/iu7039/workspace/p1/_x/link --no-info --iprepo /iu_home/
    iu7039/workspace/p1/_x/link/int/xo/ip_repo/mycompany_com_kernel_rtl_kernel_wizard_0_1_0 --messageDb /
    iu_home/iu7039/workspace/p1/_x/link/run_link/vpl.pb /iu_home/iu7039/workspace/p1/_x/link/int/dr.bd.tcl
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link

****** vpl v2020.2 (64-bit)
**** SW Build (by xbuild) on 2020-11-18-05:13:29
** Copyright 1986-2020 Xilinx, Inc. All Rights Reserved.

INFO: [VPL 60-839] Read in kernel information from file '/iu_home/iu7039/workspace/p1/_x/link/int/
    kernel_info.dat'.
INFO: [VPL 74-74] Compiler Version string: 2020.2
INFO: [VPL 60-423]   Target device: xilinx_u200_xdma_201830_2
INFO: [VPL 60-1032] Extracting hardware platform to /iu_home/iu7039/workspace/p1/_x/link/vivado/vpl/.local
    /hw_platform
WARNING: /data/Xilinx/Vitis/2020.2/tps/lnx64/jre9.0.4 does not exist.
[11:50:52] Run vpl: Step create_project: Started
Creating Vivado project.
[11:51:19] Run vpl: Step create_project: Completed
[11:51:19] Run vpl: Step create_bd: Started
[11:53:03] Run vpl: Step create_bd: RUNNING...
[11:54:41] Run vpl: Step create_bd: RUNNING...
[11:56:18] Run vpl: Step create_bd: RUNNING...
[11:58:08] Run vpl: Step create_bd: RUNNING...
[11:59:44] Run vpl: Step create_bd: RUNNING...
[12:00:20] Run vpl: Step create_bd: Completed
[12:00:20] Run vpl: Step update_bd: Started
[12:00:22] Run vpl: Step update_bd: Completed
[12:00:22] Run vpl: Step generate_target: Started
[12:01:58] Run vpl: Step generate_target: RUNNING...
[12:03:27] Run vpl: Step generate_target: RUNNING...
[12:04:52] Run vpl: Step generate_target: RUNNING...
[12:06:24] Run vpl: Step generate_target: RUNNING...
[12:07:51] Run vpl: Step generate_target: RUNNING...
[12:09:26] Run vpl: Step generate_target: RUNNING...
[12:10:56] Run vpl: Step generate_target: RUNNING...
[12:11:30] Run vpl: Step generate_target: Completed
[12:11:30] Run vpl: Step config_hw_runs: Started
[12:13:02] Run vpl: Step config_hw_runs: Completed
[12:13:02] Run vpl: Step synth: Started
[12:15:38] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:16:13] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:16:51] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:17:29] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:18:07] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:18:43] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:19:22] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:19:57] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:20:36] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:21:12] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:21:52] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:22:27] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[12:23:06] Block-level synthesis in progress, 5 of 66 jobs complete, 3 jobs running.
[12:23:42] Block-level synthesis in progress, 6 of 66 jobs complete, 2 jobs running.
[12:24:21] Block-level synthesis in progress, 6 of 66 jobs complete, 7 jobs running.
[12:24:56] Block-level synthesis in progress, 6 of 66 jobs complete, 8 jobs running.
[12:25:35] Block-level synthesis in progress, 6 of 66 jobs complete, 8 jobs running.
[12:26:13] Block-level synthesis in progress, 8 of 66 jobs complete, 6 jobs running.
[12:26:51] Block-level synthesis in progress, 8 of 66 jobs complete, 6 jobs running.
[12:27:27] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
```

```
[12:28:05] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
[12:28:41] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
[12:29:19] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
[12:29:55] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
[12:30:33] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
[12:31:10] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
[12:31:49] Block-level synthesis in progress, 12 of 66 jobs complete, 4 jobs running.
[12:32:25] Block-level synthesis in progress, 13 of 66 jobs complete, 3 jobs running.
[12:33:04] Block-level synthesis in progress, 13 of 66 jobs complete, 8 jobs running.
[12:33:40] Block-level synthesis in progress, 13 of 66 jobs complete, 8 jobs running.
[12:34:18] Block-level synthesis in progress, 14 of 66 jobs complete, 7 jobs running.
[12:34:54] Block-level synthesis in progress, 15 of 66 jobs complete, 6 jobs running.
[12:35:33] Block-level synthesis in progress, 15 of 66 jobs complete, 8 jobs running.
[12:36:09] Block-level synthesis in progress, 16 of 66 jobs complete, 7 jobs running.
[12:36:47] Block-level synthesis in progress, 16 of 66 jobs complete, 7 jobs running.
[12:37:23] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
[12:38:02] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
[12:38:38] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
[12:39:17] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
[12:39:53] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
[12:40:31] Block-level synthesis in progress, 19 of 66 jobs complete, 5 jobs running.
[12:41:08] Block-level synthesis in progress, 19 of 66 jobs complete, 5 jobs running.
[12:41:47] Block-level synthesis in progress, 19 of 66 jobs complete, 8 jobs running.
[12:42:23] Block-level synthesis in progress, 19 of 66 jobs complete, 8 jobs running.
[12:43:02] Block-level synthesis in progress, 20 of 66 jobs complete, 7 jobs running.
[12:43:40] Block-level synthesis in progress, 22 of 66 jobs complete, 5 jobs running.
[12:44:18] Block-level synthesis in progress, 22 of 66 jobs complete, 6 jobs running.
[12:44:54] Block-level synthesis in progress, 23 of 66 jobs complete, 7 jobs running.
[12:45:32] Block-level synthesis in progress, 23 of 66 jobs complete, 7 jobs running.
[12:46:09] Block-level synthesis in progress, 23 of 66 jobs complete, 8 jobs running.
[12:46:47] Block-level synthesis in progress, 23 of 66 jobs complete, 8 jobs running.
[12:47:23] Block-level synthesis in progress, 23 of 66 jobs complete, 8 jobs running.
[12:48:01] Block-level synthesis in progress, 23 of 66 jobs complete, 8 jobs running.
[12:48:38] Block-level synthesis in progress, 23 of 66 jobs complete, 8 jobs running.
[12:49:17] Block-level synthesis in progress, 23 of 66 jobs complete, 8 jobs running.
[12:49:54] Block-level synthesis in progress, 25 of 66 jobs complete, 6 jobs running.
[12:50:32] Block-level synthesis in progress, 25 of 66 jobs complete, 6 jobs running.
[12:51:10] Block-level synthesis in progress, 25 of 66 jobs complete, 8 jobs running.
[12:51:49] Block-level synthesis in progress, 28 of 66 jobs complete, 5 jobs running.
[12:52:26] Block-level synthesis in progress, 28 of 66 jobs complete, 5 jobs running.
[12:53:06] Block-level synthesis in progress, 29 of 66 jobs complete, 7 jobs running.
[12:53:42] Block-level synthesis in progress, 30 of 66 jobs complete, 6 jobs running.
[12:54:20] Block-level synthesis in progress, 31 of 66 jobs complete, 6 jobs running.
[12:54:58] Block-level synthesis in progress, 31 of 66 jobs complete, 7 jobs running.
[12:55:34] Block-level synthesis in progress, 31 of 66 jobs complete, 8 jobs running.
[12:56:10] Block-level synthesis in progress, 31 of 66 jobs complete, 8 jobs running.
[12:56:48] Block-level synthesis in progress, 31 of 66 jobs complete, 8 jobs running.
[12:57:25] Block-level synthesis in progress, 31 of 66 jobs complete, 8 jobs running.
[12:58:04] Block-level synthesis in progress, 31 of 66 jobs complete, 8 jobs running.
[12:58:40] Block-level synthesis in progress, 33 of 66 jobs complete, 6 jobs running.
[12:59:19] Block-level synthesis in progress, 33 of 66 jobs complete, 6 jobs running.
[12:59:56] Block-level synthesis in progress, 33 of 66 jobs complete, 8 jobs running.
[13:00:33] Block-level synthesis in progress, 35 of 66 jobs complete, 6 jobs running.
[13:01:10] Block-level synthesis in progress, 37 of 66 jobs complete, 4 jobs running.
[13:01:50] Block-level synthesis in progress, 37 of 66 jobs complete, 8 jobs running.
[13:02:26] Block-level synthesis in progress, 40 of 66 jobs complete, 5 jobs running.
[13:03:05] Block-level synthesis in progress, 42 of 66 jobs complete, 3 jobs running.
[13:03:42] Block-level synthesis in progress, 42 of 66 jobs complete, 6 jobs running.
[13:04:20] Block-level synthesis in progress, 42 of 66 jobs complete, 8 jobs running.
[13:04:57] Block-level synthesis in progress, 44 of 66 jobs complete, 6 jobs running.
[13:05:36] Block-level synthesis in progress, 44 of 66 jobs complete, 6 jobs running.
[13:06:13] Block-level synthesis in progress, 44 of 66 jobs complete, 8 jobs running.
[13:06:52] Block-level synthesis in progress, 45 of 66 jobs complete, 7 jobs running.
[13:07:29] Block-level synthesis in progress, 46 of 66 jobs complete, 6 jobs running.
[13:08:08] Block-level synthesis in progress, 46 of 66 jobs complete, 7 jobs running.
[13:08:45] Block-level synthesis in progress, 47 of 66 jobs complete, 7 jobs running.
[13:09:24] Block-level synthesis in progress, 48 of 66 jobs complete, 6 jobs running.
[13:10:01] Block-level synthesis in progress, 48 of 66 jobs complete, 7 jobs running.
[13:10:38] Block-level synthesis in progress, 48 of 66 jobs complete, 8 jobs running.
[13:11:15] Block-level synthesis in progress, 51 of 66 jobs complete, 5 jobs running.
[13:11:54] Block-level synthesis in progress, 53 of 66 jobs complete, 3 jobs running.
[13:12:32] Block-level synthesis in progress, 53 of 66 jobs complete, 7 jobs running.
[13:13:10] Block-level synthesis in progress, 54 of 66 jobs complete, 7 jobs running.
[13:13:46] Block-level synthesis in progress, 58 of 66 jobs complete, 3 jobs running.
[13:14:26] Block-level synthesis in progress, 58 of 66 jobs complete, 6 jobs running.
[13:15:03] Block-level synthesis in progress, 58 of 66 jobs complete, 8 jobs running.
[13:15:42] Block-level synthesis in progress, 60 of 66 jobs complete, 6 jobs running.
[13:16:21] Block-level synthesis in progress, 60 of 66 jobs complete, 6 jobs running.
[13:16:59] Block-level synthesis in progress, 60 of 66 jobs complete, 6 jobs running.
[13:17:37] Block-level synthesis in progress, 60 of 66 jobs complete, 6 jobs running.
[13:18:17] Block-level synthesis in progress, 60 of 66 jobs complete, 6 jobs running.
[13:18:55] Block-level synthesis in progress, 61 of 66 jobs complete, 5 jobs running.
```

```
[13:19:35] Block-level synthesis in progress, 61 of 66 jobs complete, 5 jobs running.
[13:20:13] Block-level synthesis in progress, 62 of 66 jobs complete, 4 jobs running.
[13:20:53] Block-level synthesis in progress, 63 of 66 jobs complete, 3 jobs running.
[13:21:31] Block-level synthesis in progress, 63 of 66 jobs complete, 3 jobs running.
[13:22:11] Block-level synthesis in progress, 64 of 66 jobs complete, 2 jobs running.
[13:22:48] Block-level synthesis in progress, 64 of 66 jobs complete, 2 jobs running.
[13:23:29] Block-level synthesis in progress, 64 of 66 jobs complete, 2 jobs running.
[13:24:08] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
[13:24:48] Block-level synthesis in progress, 66 of 66 jobs complete, 0 jobs running.
[13:25:26] Block-level synthesis in progress, 66 of 66 jobs complete, 0 jobs running.
[13:26:06] Top-level synthesis in progress.
[13:26:44] Top-level synthesis in progress.
[13:27:25] Top-level synthesis in progress.
[13:28:03] Top-level synthesis in progress.
[13:28:43] Top-level synthesis in progress.
[13:29:21] Top-level synthesis in progress.
[13:30:01] Top-level synthesis in progress.
[13:30:38] Top-level synthesis in progress.
[13:31:19] Top-level synthesis in progress.
[13:31:57] Top-level synthesis in progress.
[13:32:38] Top-level synthesis in progress.
[13:33:16] Top-level synthesis in progress.
[13:33:56] Top-level synthesis in progress.
[13:34:33] Top-level synthesis in progress.
[13:35:14] Top-level synthesis in progress.
[13:35:51] Top-level synthesis in progress.
[13:36:31] Top-level synthesis in progress.
[13:36:55] Run vpl: Step synth: Completed
[13:36:55] Run vpl: Step impl: Started
[14:39:18] Finished 2nd of 6 tasks (FPGA linking synthesized kernels to platform). Elapsed time: 02h 53m
    26s

[14:39:18] Starting logic optimization..
[14:45:53] Phase 1 Generate And Synthesize MIG Cores
[15:23:12] Phase 2 Generate And Synthesize Debug Cores
[15:49:15] Phase 3 Retarget
[15:52:00] Phase 4 Constant propagation
[15:53:22] Phase 5 Sweep
[15:58:08] Phase 6 BUFG optimization
[16:00:12] Phase 7 Shift Register Optimization
[16:00:55] Phase 8 Post Processing Netlist
[16:14:55] Finished 3rd of 6 tasks (FPGA logic optimization). Elapsed time: 01h 35m 37s

[16:14:55] Starting logic placement..
[16:19:40] Phase 1 Placer Initialization
[16:19:40] Phase 1.1 Placer Initialization Netlist Sorting
[16:31:40] Phase 1.2 IO Placement/ Clock Placement/ Build Placer Device
[16:39:35] Phase 1.3 Build Placer Netlist Model
[16:52:12] Phase 1.4 Constrain Clocks/Macros
[16:53:37] Phase 2 Global Placement
[16:53:37] Phase 2.1 Floorplanning
[16:57:14] Phase 2.1.1 Partition Driven Placement
[16:57:14] Phase 2.1.1.1 PBP: Partition Driven Placement
[16:59:27] Phase 2.1.1.2 PBP: Clock Region Placement
[17:04:04] Phase 2.1.1.3 PBP: Compute Congestion
[17:05:23] Phase 2.1.1.4 PBP: UpdateTiming
[17:06:52] Phase 2.1.1.5 PBP: Add part constraints
[17:06:52] Phase 2.2 Update Timing before SLR Path Opt
[17:07:34] Phase 2.3 Global Placement Core
[17:39:57] Phase 2.3.1 Physical Synthesis In Placer
[17:53:24] Phase 3 Detail Placement
[17:53:24] Phase 3.1 Commit Multi Column Macros
[17:54:12] Phase 3.2 Commit Most Macros & LUTRAMs
[17:59:15] Phase 3.3 Small Shape DP
[17:59:15] Phase 3.3.1 Small Shape Clustering
[18:01:34] Phase 3.3.2 Flow Legalize Slice Clusters
[18:02:18] Phase 3.3.3 Slice Area Swap
[18:07:36] Phase 3.4 Place Remaining
[18:08:19] Phase 3.5 Re-assign LUT pins
[18:09:49] Phase 3.6 Pipeline Register Optimization
[18:09:49] Phase 3.7 Fast Optimization
[18:14:21] Phase 4 Post Placement Optimization and Clean-Up
[18:14:21] Phase 4.1 Post Commit Optimization
[18:23:07] Phase 4.1.1 Post Placement Optimization
[18:23:53] Phase 4.1.1.1 BUFG Insertion
[18:23:53] Phase 1 Physical Synthesis Initialization
[18:26:48] Phase 4.1.1.2 BUFG Replication
[18:29:44] Phase 4.1.1.3 Replication
[18:36:11] Phase 4.2 Post Placement Cleanup
[18:36:56] Phase 4.3 Placer Reporting
[18:36:56] Phase 4.3.1 Print Estimated Congestion
[18:38:24] Phase 4.4 Final Placement Cleanup
```

```
[19:46:20] Finished 4th of 6 tasks (FPGA logic placement). Elapsed time: 03h 31m 24s

[19:46:20] Starting logic routing..
[19:51:30] Phase 1 Build RT Design
[20:01:57] Phase 2 Router Initialization
[20:02:44] Phase 2.1 Fix Topology Constraints
[20:02:44] Phase 2.2 Pre Route Cleanup
[20:03:26] Phase 2.3 Global Clock Net Routing
[20:06:31] Phase 2.4 Update Timing
[20:19:59] Phase 2.5 Update Timing for Bus Skew
[20:19:59] Phase 2.5.1 Update Timing
[20:25:04] Phase 3 Initial Routing
[20:25:04] Phase 3.1 Global Routing
[20:29:31] Phase 4 Rip-up And Reroute
[20:29:31] Phase 4.1 Global Iteration 0
[20:55:43] Phase 4.2 Global Iteration 1
[21:08:17] Phase 4.3 Global Iteration 2
[21:12:46] Phase 5 Delay and Skew Optimization
[21:12:46] Phase 5.1 Delay CleanUp
[21:12:46] Phase 5.1.1 Update Timing
[21:19:22] Phase 5.2 Clock Skew Optimization
[21:20:08] Phase 6 Post Hold Fix
[21:20:08] Phase 6.1 Hold Fix Iter
[21:20:51] Phase 6.1.1 Update Timing
[21:25:57] Phase 7 Route finalize
[21:26:38] Phase 8 Verifying routed nets
[21:28:07] Phase 9 Depositing Routes
[21:31:51] Phase 10 Route finalize
[21:32:33] Phase 11 Post Router Timing
[21:39:06] Finished 5th of 6 tasks (FPGA routing). Elapsed time: 01h 52m 46s

[21:39:06] Starting bitstream generation..
[23:33:58] Creating bitmap...
[00:21:31] Writing bitstream ./pfm_top_i_dynamic_region_my_rm_partial.bit...
[00:22:18] Finished 6th of 6 tasks (FPGA bitstream generation). Elapsed time: 02h 43m 11s
[00:26:12] Run vpl: Step impl: Completed
[00:26:30] Run vpl: FINISHED. Run Status: impl Complete!
INFO: [v++ 60-1441] [00:27:12] Run run_link: Step vpl: Completed
Time (s): cpu = 00:41:43 ; elapsed = 12:41:31 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
      physical = 230667 ; free virtual = 258886
INFO: [v++ 60-1443] [00:27:12] Run run_link: Step rtdgen: Started
INFO: [v++ 60-1453] Command Line: rtdgen
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link
INFO: [v++ 60-991] clock name 'clkwiz_kernel_clk_out1' (clock ID '0') is being mapped to clock name '
      DATA_CLK' in the xclbin
INFO: [v++ 60-991] clock name 'clkwiz_kernel2_clk_out1' (clock ID '1') is being mapped to clock name '
      KERNEL_CLK' in the xclbin
INFO: [v++ 60-1230] The compiler selected the following frequencies for the runtime controllable kernel
      clock(s) and scalable system clock(s): Kernel (DATA) clock: clkwiz_kernel_clk_out1 = 300, Kernel (
      KERNEL) clock: clkwiz_kernel2_clk_out1 = 500
INFO: [v++ 60-1453] Command Line: cf2sw -a /iu_home/iu7039/workspace/p1/_x/link/int/address_map.xml -sdsl
      /iu_home/iu7039/workspace/p1/_x/link/int/sdsl.dat -xclbin /iu_home/iu7039/workspace/p1/_x/link/int/
      xclbin_orig.xml -rtd /iu_home/iu7039/workspace/p1/_x/link/int/vinc.rtd -o /iu_home/iu7039/workspace/p1
      /_x/link/int/vinc.xml
INFO: [v++ 60-1652] Cf2sw returned exit code: 0
INFO: [v++ 60-2311] HPISystemDiagram::writeSystemDiagramAfterRunningVivado, rtdInputFilePath: /iu_home/
      iu7039/workspace/p1/_x/link/int/vinc.rtd
INFO: [v++ 60-2312] HPISystemDiagram::writeSystemDiagramAfterRunningVivado, systemDiagramOutputFilePath: /
      iu_home/iu7039/workspace/p1/_x/link/int/systemDiagramModelSlrBaseAddress.json
INFO: [v++ 60-1618] Launching
INFO: [v++ 60-1441] [00:27:27] Run run_link: Step rtdgen: Completed
Time (s): cpu = 00:00:14 ; elapsed = 00:00:15 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
      physical = 230695 ; free virtual = 258914
INFO: [v++ 60-1443] [00:27:27] Run run_link: Step xclbinutil: Started
INFO: [v++ 60-1453] Command Line: xclbinutil --add-section DEBUG_IP_LAYOUT:JSON:/iu_home/iu7039/workspace/
      p1/_x/link/int/debug_ip_layout.rtd --add-section BITSTREAM:RAW:/iu_home/iu7039/workspace/p1/_x/link/
      int/partial.bit --force --target hw --key-value SYS:dfx_enable:true --add-section :JSON:/iu_home/
      iu7039/workspace/p1/_x/link/int/vinc.rtd --append-section :JSON:/iu_home/iu7039/workspace/p1/_x/link/
      int/appendSection.rtd --add-section CLOCK_FREQ_TOPOLOGY:JSON:/iu_home/iu7039/workspace/p1/_x/link/int/
      vinc_xml.rtd --add-section BUILD_METADATA:JSON:/iu_home/iu7039/workspace/p1/_x/link/int/vinc_build.rtd
       --add-section EMBEDDED_METADATA:RAW:/iu_home/iu7039/workspace/p1/_x/link/int/vinc.xml --add-section
      SYSTEM_METADATA:RAW:/iu_home/iu7039/workspace/p1/_x/link/int/systemDiagramModelSlrBaseAddress.json --
      output /iu_home/iu7039/workspace/p1/vinc.xclbin
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link
XRT Build Version: 2.8.743 (2020.2)
 Build Date: 2020-11-16 00:19:11
 Hash ID: 77d5484b5c4daa691a7f78235053fb036829b1e9
 Creating a default 'in-memory' xclbin image.

 Section: 'DEBUG_IP_LAYOUT'(9) was successfully added.
 Size   : 440 bytes
 Format : JSON
```

```
File    : '/iu_home/iu7039/workspace/p1/_x/link/int/debug_ip_layout.rtd'

Section: 'BITSTREAM'(0) was successfully added.
Size    : 43430742 bytes
Format : RAW
File    : '/iu_home/iu7039/workspace/p1/_x/link/int/partial.bit'

Section: 'MEM_TOPOLOGY'(6) was successfully added.
Format : JSON
File    : 'mem_topology'

Section: 'IP_LAYOUT'(8) was successfully added.
Format : JSON
File    : 'ip_layout'

Section: 'CONNECTIVITY'(7) was successfully added.
Format : JSON
File    : 'connectivity'
Section: 'CLOCK_FREQ_TOPOLOGY'(11) was successfully added.
Size    : 274 bytes
Format : JSON
File    : '/iu_home/iu7039/workspace/p1/_x/link/int/vinc_xml.rtd'
Section: 'BUILD_METADATA'(14) was successfully added.
Size    : 3025 bytes
Format : JSON
File    : '/iu_home/iu7039/workspace/p1/_x/link/int/vinc_build.rtd'
Section: 'EMBEDDED_METADATA'(2) was successfully added.
Size    : 2754 bytes
Format : RAW
File    : '/iu_home/iu7039/workspace/p1/_x/link/int/vinc.xml'
Section: 'SYSTEM_METADATA'(22) was successfully added.
Size    : 6254 bytes
Format : RAW
File    : '/iu_home/iu7039/workspace/p1/_x/link/int/systemDiagramModelSlrBaseAddress.json'
Section: 'IP_LAYOUT'(8) was successfully appended to.
Format : JSON
File    : 'ip_layout'
Successfully wrote (43453522 bytes) to the output file: /iu_home/iu7039/workspace/p1/vinc.xclbin
Leaving xclbinutil.
INFO: [v++ 60-1441] [00:27:30] Run run_link: Step xclbinutil: Completed
Time (s): cpu = 00:00:00.53 ; elapsed = 00:00:03 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
    physical = 230646 ; free virtual = 258906
INFO: [v++ 60-1443] [00:27:30] Run run_link: Step xclbinutilinfo: Started
INFO: [v++ 60-1453] Command Line: xclbinutil --quiet --force --info /iu_home/iu7039/workspace/p1/vinc.
    xclbin.info --input /iu_home/iu7039/workspace/p1/vinc.xclbin
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link
INFO: [v++ 60-1441] [00:27:34] Run run_link: Step xclbinutilinfo: Completed
Time (s): cpu = 00:00:03 ; elapsed = 00:00:04 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
    physical = 230622 ; free virtual = 258883
INFO: [v++ 60-1443] [00:27:34] Run run_link: Step generate_sc_driver: Started
INFO: [v++ 60-1453] Command Line:
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7039/workspace/p1/_x/link/run_link
INFO: [v++ 60-1441] [00:27:34] Run run_link: Step generate_sc_driver: Completed
Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.06 . Memory (MB): peak = 1553.914 ; gain = 0.000 ; free
    physical = 230618 ; free virtual = 258879
INFO: [v++ 60-244] Generating system estimate report...
INFO: [v++ 60-1092] Generated system estimate report: /iu_home/iu7039/workspace/p1/_x/reports/link/
    system_estimate_vinc.xtxt
INFO: [v++ 60-586] Created /iu_home/iu7039/workspace/p1/vinc.ltx
INFO: [v++ 60-586] Created vinc.xclbin
INFO: [v++ 60-1307] Run completed. Additional information can be found in:
Guidance: /iu_home/iu7039/workspace/p1/_x/reports/link/v++_link_vinc_guidance.html
Timing Report: /iu_home/iu7039/workspace/p1/_x/reports/link/imp/
    impl_1_xilinx_u200_xdma_201830_2_bb_locked_timing_summary_routed.rpt
Vivado Log: /iu_home/iu7039/workspace/p1/_x/logs/link/vivado.log
Steps Log File: /iu_home/iu7039/workspace/p1/_x/logs/link/link.steps.log
INFO: [v++ 60-2343] Use the vitis_analyzer tool to visualize and navigate the relevant reports. Run the
    following command.
vitis_analyzer /iu_home/iu7039/workspace/p1/vinc.xclbin.link_summary
INFO: [v++ 60-791] Total elapsed time: 12h 46m 1s
INFO: [v++ 60-1653] Closing dispatch client.
```

Приложение 3

Листинг 3 – Содержимое xclbin.info-файла

```

==============================================================================
XRT Build Version: 2.8.743 (2020.2)
 Build Date: 2020-11-16 00:19:11
 Hash ID: 77d5484b5c4daa691a7f78235053fb036829b1e9
==============================================================================
 xclbin Information
 ------------------
 Generated by:           v++ (2020.2) on 2020-11-18-05:13:29
 Version:                2.8.743
 Kernels:                rtl_kernel_wizard_0
 Signature:
 Content:                 Bitstream
UUID (xclbin):           f48ab827-3033-4124-9be2-3e2b03b06a1f
 Sections:                DEBUG_IP_LAYOUT, BITSTREAM, MEM_TOPOLOGY, IP_LAYOUT,
CONNECTIVITY, CLOCK_FREQ_TOPOLOGY, BUILD_METADATA,
EMBEDDED_METADATA, SYSTEM_METADATA,
GROUP_CONNECTIVITY, GROUP_TOPOLOGY
==============================================================================
 Hardware Platform (Shell) Information
 -------------------------------------
 Vendor:                  xilinx
 Board:                   u200
 Name:                   xdma
 Version:                 201830.2
 Generated Version:       Vivado 2018.3 (SW Build: 2568420)
 Created:                 Tue Jun 25 06:55:20 2019
FPGA Device:              xcu200
 Board Vendor:            xilinx.com
 Board Name:              xilinx.com:au200:1.0
 Board Part:              xilinx.com:au200:part0:1.0
 Platform VBNV:           xilinx_u200_xdma_201830_2
 Static UUID:             c102e7af-b2b8-4381-992b-9a00cc3863eb
 Feature ROM TimeStamp:   1561465320

 Clocks
------
Name:        DATA_CLK
 Index:       0
 Type:        DATA
 Frequency:  300 MHz

Name:        KERNEL_CLK
 Index:       1
 Type:        KERNEL
 Frequency:  500 MHz

Memory Configuration
--------------------
Name:           bank0
 Index:          0
 Type:          MEM_DDR4
 Base Address:  0x4000000000
 Address Size:  0x400000000
 Bank Used:     Yes

Name:           bank1
 Index:          1
 Type:          MEM_DDR4
 Base Address:  0x5000000000
 Address Size:  0x400000000
 Bank Used:     No

Name:           bank2
 Index:          2
 Type:          MEM_DDR4
 Base Address:  0x6000000000
 Address Size:  0x400000000
 Bank Used:     No

Name:           bank3
 Index:          3
 Type:          MEM_DDR4
 Base Address:  0x7000000000
 Address Size:  0x400000000
```

```
Bank Used:      No

Name:           PLRAM[0]
Index:          4
Type:           MEM_DRAM
Base Address: 0x3000000000
Address Size: 0x20000
Bank Used:      No

Name:           PLRAM[1]
Index:          5
Type:           MEM_DRAM
Base Address: 0x3000200000
Address Size: 0x20000
Bank Used:      No

Name:           PLRAM[2]
Index:          6
Type:           MEM_DRAM
Base Address: 0x3000400000
Address Size: 0x20000
Bank Used:      No
==============================================================================
Kernel: rtl_kernel_wizard_0

Definition
----------
Signature: rtl_kernel_wizard_0 (uint num, int* axi00_ptr0)

Ports
-----
Port:            s_axi_control
Mode:            slave
Range (bytes): 0x1000
Data Width:      32 bits
Port Type:       addressable

Port:            m00_axi
Mode:            master
Range (bytes): 0xFFFFFFFFFFFFFFFF
Data Width:      512 bits
Port Type:       addressable

--------------------------
Instance:          vinc0
Base Address: 0x1c00000

Argument:            num
Register Offset:    0x010
Port:                s_axi_control
Memory:              <not applicable>

Argument:            axi00_ptr0
Register Offset:    0x018
Port:                m00_axi
Memory:              bank0 (MEM_DDR4)
==============================================================================
Generated By
------------
Command:         v++
Version:         2020.2 - 2020-11-18-05:13:29 (SW BUILD: 0)
Command Line:  v++ --config /iu_home/iu7039/workspace/p1/Alveo_lab01.cfg --connectivity.nk rtl_kernel_wizard_0:1:vinc0 --connectivity.slr vinc0:SLR0 --connectivity.sp vinc0.m00_axi:DDR[0] --input_files /iu_home/iu7039/workspace/p1/Alveo_lab01_kernels/src/vitis_rtl_kernel/rtl_kernel_wizard_0/rtl_kernel_wizard_0.xo --link --optimize 0 --output vinc.xclbin --platform xilinx_u200_xdma_201830_2 --report_level 0 --target hw --vivado.prop run.impl_1.STEPS.OPT_DESIGN.ARGS.DIRECTIVE=Explore --vivado.prop run.impl_1.STEPS.PLACE_DESIGN.ARGS.DIRECTIVE=Explore --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.IS_ENABLED=true --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.ARGS.DIRECTIVE=AggressiveExplore --vivado.prop run.impl_1.STEPS.ROUTE_DESIGN.ARGS.DIRECTIVE=Explore
Options:         --config /iu_home/iu7039/workspace/p1/Alveo_lab01.cfg
--connectivity.nk rtl_kernel_wizard_0:1:vinc0
--connectivity.slr vinc0:SLR0
--connectivity.sp vinc0.m00_axi:DDR[0]
--input_files /iu_home/iu7039/workspace/p1/Alveo_lab01_kernels/src/vitis_rtl_kernel/rtl_kernel_wizard_0/rtl_kernel_wizard_0.xo
--link
--optimize 0
--output vinc.xclbin
--platform xilinx_u200_xdma_201830_2
--report_level 0
--target hw
--vivado.prop run.impl_1.STEPS.OPT_DESIGN.ARGS.DIRECTIVE=Explore
```

```
--vivado.prop run.impl_1.STEPS.PLACE_DESIGN.ARGS.DIRECTIVE=Explore
--vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.IS_ENABLED=true
--vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.ARGS.DIRECTIVE=AggressiveExplore
--vivado.prop run.impl_1.STEPS.ROUTE_DESIGN.ARGS.DIRECTIVE=Explore
================================================================================
User Added Key Value Pairs
--------------------------
<empty>
```