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»

Отчёт

по лабораторной работе №4

Название: Методология разработки и верификации ускорителей вычислений на платформе Xilinx Alveo

Дисциплина: Архитектура ЭВМ

Студент ИУ7-54Б _____ Л.Е.Тартыков

(Группа) (Подпись, дата) (И.О. Фамилия)

Преподаватель _____ А.Ю.Попов

(Подпись, дата) (И.О. Фамилия)

Москва, 2021

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 (рисунок 1). В рамках всей системы используется единое 64-х разрядное адресное пространство, в котором формируются адреса на всех AXI4 шинах.

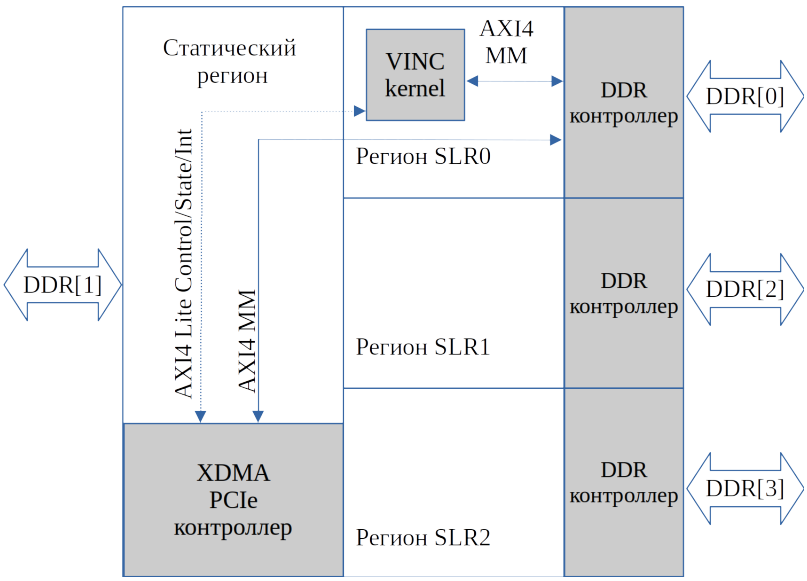

Рисунок 1 – Размещение проекта на ПЛИС xcu200-fsgd2104-2-e карты Alveo U200

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.

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

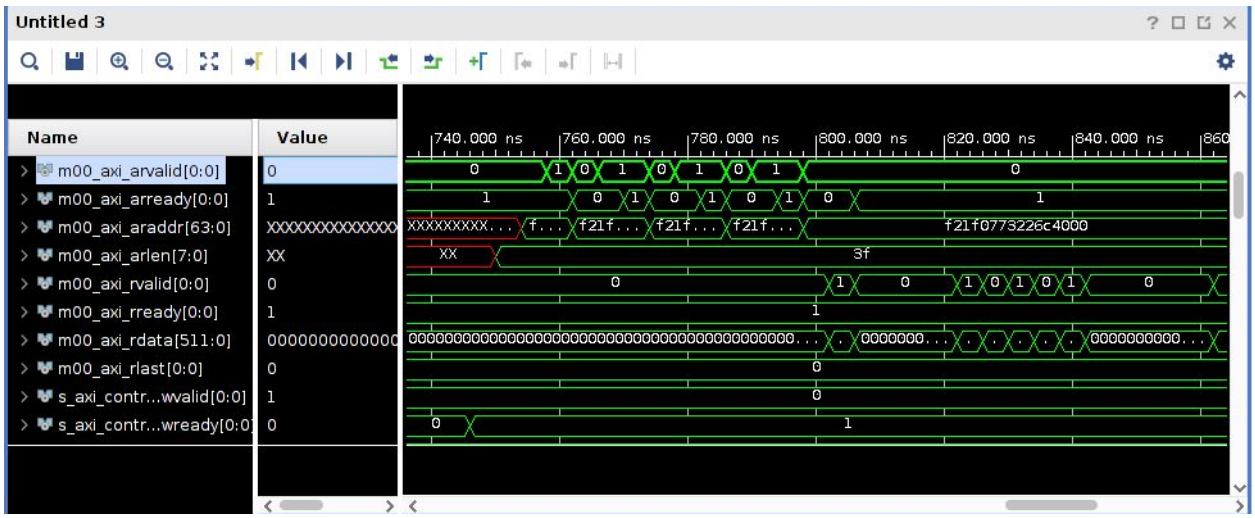

Рисунок 2 – Транзакция чтения данных вектора на шине AXI4 MM из DDR памяти

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

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

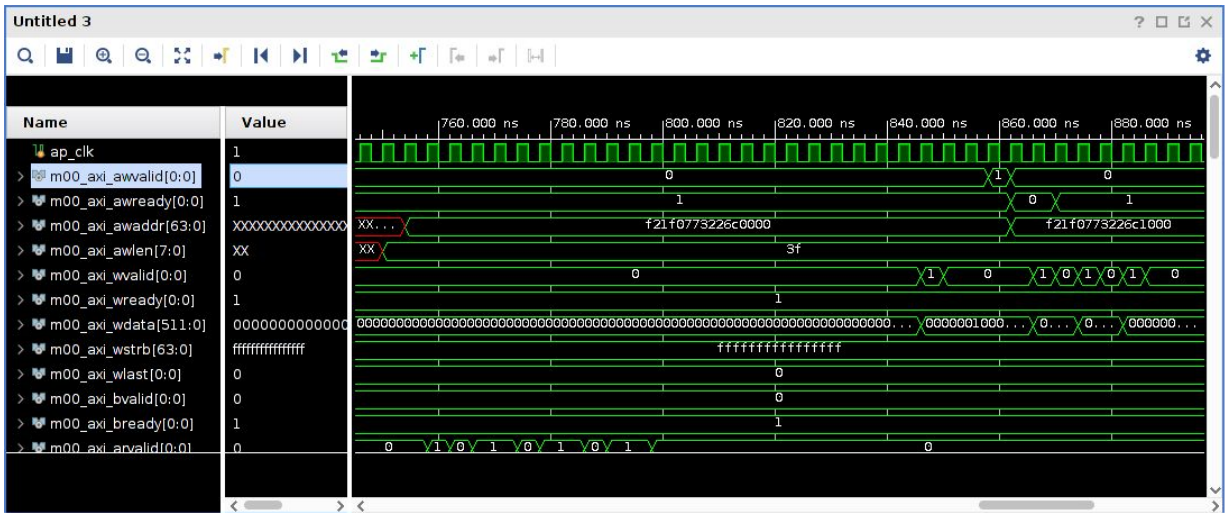

Рисунок 3 – Транзакция записи результата инкремента данных на шине AXI4 MM

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

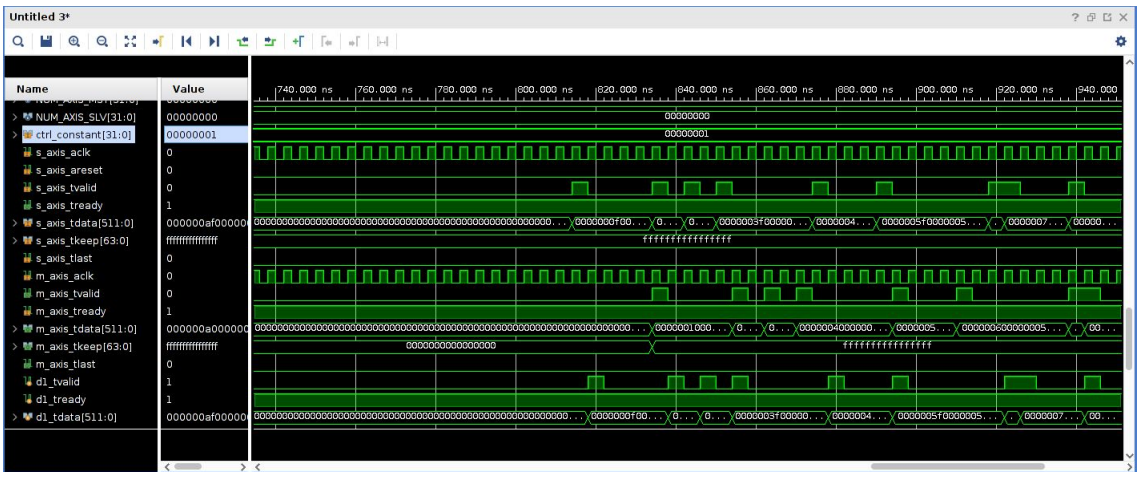

Рисунок 4 – Транзакция записи результата инкремента данных на шине AXI4 MM

Копии экранов моделирования исходного проекта VINC (измененная программа)

В соответствии с вариантом 16 необходимо было изменить код проекта. Реализовать функцию по формуле (1):

$$R[i] = A[i] * 4 - 16 \tag{1}$$

На рисунке 5 приведен код измененнной программы.

```
76
77  // Adder function
78  always @(posedge s_axis_aclk) begin
79    for (i = 0; i < LP_NUM_LOOPS; i = i + 1) begin
80      d2_tdata[i*C_ADDER_BIT_WIDTH+:C_ADDER_BIT_WIDTH] <= d1_tdata[C_ADDER_BIT_WIDTH*i+:C_ADDER_BIT_WIDTH] * 4 - 16;
81    end
82  end
```

Рисунок 5 – Измененный код модуля _adder.v

На рисунке 6 приведена транзакция чтения данных вектора на шине AXI4 MM из DDR памяти.

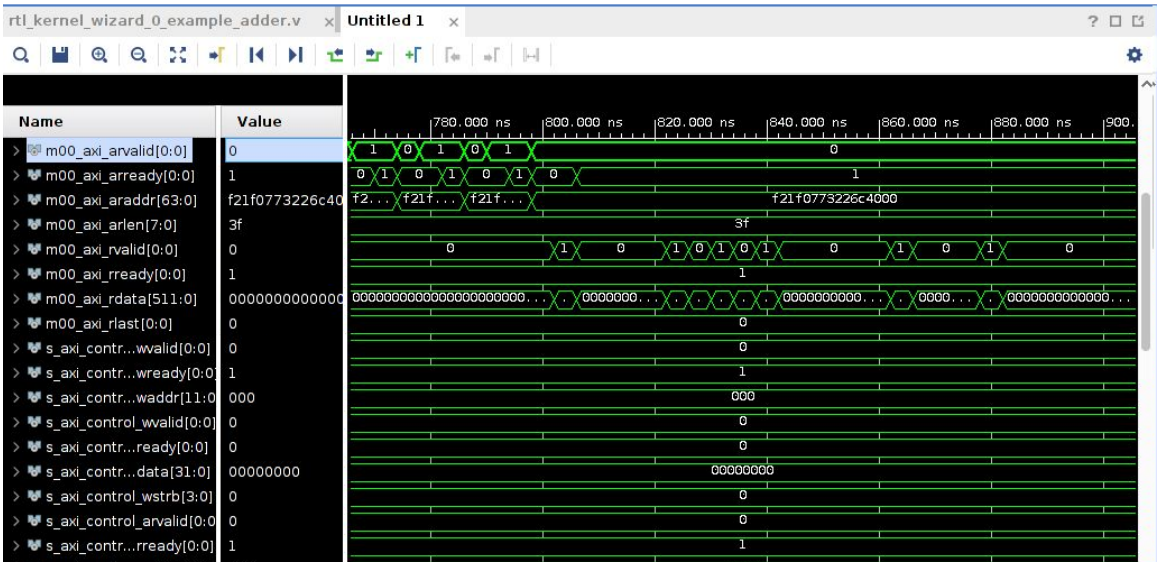

Рисунок 6 – Транзакция чтения данных вектора на шине AXI4 MM из DDR памяти

На рисунке 7 приведена транзакция записи результата инкремента данных на шине AXI4 MM.

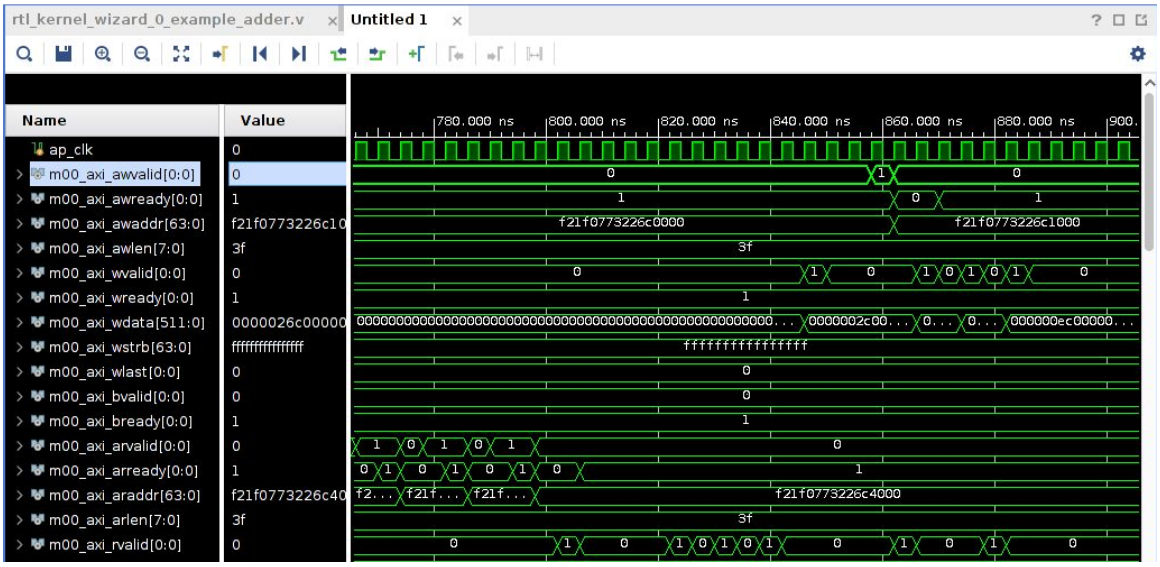

Рисунок 7 – Транзакция записи результата инкремента данных на шине AXI4 MM

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 8 представлен конфигурационный файл для сборки проекта.

Alveo_lab1.cfg | host_example... | Alveo_lab1.cfg | host_example...

```
1 [connectivity]
2 nk=rtl_kernel_wizard_0:1:vinc0
3
4 slr=vinc0:SLR2
5 sp=vinc0.m00_axi:DDR[3]
6
7 [vivado]
8 prop=run.impl_1.STEPSOPT_DESIGN.ARGS.DIRECTIVE=Explore
9 prop=run.impl_1.STEPS.PLACE_DESIGN.ARGS.DIRECTIVE=Explore
10 prop=run.impl_1.STEPS.PHYS_OPT_DESIGN.IS_ENABLED=true
11 proprun.impl_1.STEPS.PHYS_OPT_DESIGN.ARGS.DIRECTIVE=AggressiveExplore
12 prop=run.impl_1.STEPS.ROUE_DESIGN.ARGS.DIRECTIVE=Explore
```

Рисунок 8 – 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 9. Было изменено условие проверки, на проверку, соответствующую моему варианту.

```
315
316     for (cl_uint i = 0; i < number_of_words; i++) {
317         if ((h_data[i] * 4 - 16) != h_axi00_ptr0_output[i]) {
318             printf("ERROR in rtl_kernel_wizard_0::m00_axi - array index %d (host addr 0x%03
319             check_status = 1;
320         }
321         //printf("i=%d, input=%d, output=%d\n", i,  h_axi00_ptr0_input[i] h_axi00_ptr0_outp
322     }
```

Рисунок 9 – Фрагмент кода host_example.cpp

Тесты запускались с помощью команды Run->Run Configurations с настройкой параметров согласно условию лабораторной работы. По результатам на рисунке 10 можно увидеть, что все тесты выполнены успешно.

```
<terminated> (exit value: 0) SystemDebugger_Alveo_lab1_system_Alveo_lab1 [OpenCL] /iu_home/iu7136/workspac
[Console output redirected to file:/iu_home/iu7136/workspace/Alveo_lab1/Hardware/SystemD
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/iu7136/workspace/vinc.xclbin
INFO: Test completed successfully.
```

Рисунок 10 – Выполнение тестов

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.

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]).

Приложение А

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


```

```
74      cl_platform_id platform_id;          // platform id
75      cl_device_id device_id;              // compute device id
76      cl_context context;                  // compute context
77      cl_command_queue commands;           // compute command queue
78      cl_program program;                  // compute programs
79      cl_kernel kernel;                    // compute kernel
80
81      cl_uint* h_data;                                  // host memory for input vector
82      char cl_platform_vendor[1001];
83      char target_device_name[1001] = TARGET_DEVICE;
84
85      cl_uint* h_axi00_ptr0_output = (cl_uint*)aligned_alloc(MEM_ALIGNMENT,MAX_LENGTH * sizeof(cl_uint*))
            ; // host memory for output vector
86      cl_mem d_axi00_ptr0;                            // device memory used for a vector
87
88      if (argc != 2) {
89          printf("Usage: %s xclbin\n", argv[0]);
90          return EXIT_FAILURE;
91      }
92
93      // Fill our data sets with pattern
94      h_data = (cl_uint*)aligned_alloc(MEM_ALIGNMENT,MAX_LENGTH * sizeof(cl_uint*));
95      for(cl_uint i = 0; i < MAX_LENGTH; i++) {
96          h_data[i]  = i;
97          h_axi00_ptr0_output[i] = 0;
98
99      }
100
101     // Get all platforms and then select Xilinx platform
102     cl_platform_id platforms[16];        // platform id
103     cl_uint platform_count;
104     cl_uint platform_found = 0;
105     err = clGetPlatformIDs(16, platforms, &platform_count);
106     if (err != CL_SUCCESS) {
107         printf("ERROR: Failed to find an OpenCL platform!\n");
108         printf("ERROR: Test failed\n");
109         return EXIT_FAILURE;
110     }
111     printf("INFO: Found %d platforms\n", platform_count);
112
113     // Find Xilinx Plaftorm
114     for (cl_uint iplat=0; iplat<platform_count; iplat++) {
115         err = clGetPlatformInfo(platforms[iplat], CL_PLATFORM_VENDOR, 1000, (void *)cl_platform_vendor,
                NULL);
116         if (err != CL_SUCCESS) {
117             printf("ERROR: clGetPlatformInfo(CL_PLATFORM_VENDOR) failed!\n");
118             printf("ERROR: Test failed\n");
119             return EXIT_FAILURE;
120         }
121         if (strcmp(cl_platform_vendor, "Xilinx") == 0) {
122             printf("INFO: Selected platform %d from %s\n", iplat, cl_platform_vendor);
123             platform_id = platforms[iplat];
124             platform_found = 1;
125         }
126     }
127     if (!platform_found) {
128         printf("ERROR: Platform Xilinx not found. Exit.\n");
129         return EXIT_FAILURE;
130     }
131
132     // Get Accelerator compute device
133     cl_uint num_devices;
134     cl_uint device_found = 0;
135     cl_device_id devices[16];  // compute device id
136     char cl_device_name[1001];
137     err = clGetDeviceIDs(platform_id, CL_DEVICE_TYPE_ACCELERATOR, 16, devices, &num_devices);
138     printf("INFO: Found %d devices\n", num_devices);
139     if (err != CL_SUCCESS) {
140         printf("ERROR: Failed to create a device group!\n");
141         printf("ERROR: Test failed\n");
142         return -1;
143     }
144
145     //iterate all devices to select the target device.
146     for (cl_uint i=0; i<num_devices; i++) {
147         err = clGetDeviceInfo(devices[i], CL_DEVICE_NAME, 1024, cl_device_name, 0);
148         if (err != CL_SUCCESS) {
149             printf("ERROR: Failed to get device name for device %d!\n", i);
150             printf("ERROR: Test failed\n");
151             return EXIT_FAILURE;
152         }
153         printf("CL_DEVICE_NAME %s\n", cl_device_name);
```

```
154         if(strcmp(cl_device_name, target_device_name) == 0) {
155             device_id = devices[i];
156             device_found = 1;
157             printf("Selected %s as the target device\n", cl_device_name);
158         }
159     }
160 
161     if (!device_found) {
162         printf("ERROR: Target device %s not found. Exit.\n", target_device_name);
163         return EXIT_FAILURE;
164     }
165 
166     // Create a compute context
167     //
168     context = clCreateContext(0, 1, &device_id, NULL, NULL, &err);
169     if (!context) {
170         printf("ERROR: Failed to create a compute context!\n");
171         printf("ERROR: Test failed\n");
172         return EXIT_FAILURE;
173     }
174 
175     // Create a command commands
176     commands = clCreateCommandQueue(context, device_id, CL_QUEUE_PROFILING_ENABLE |
            CL_QUEUE_OUT_OF_ORDER_EXEC_MODE_ENABLE, &err);
177     if (!commands) {
178         printf("ERROR: Failed to create a command commands!\n");
179         printf("ERROR: code %i\n",err);
180         printf("ERROR: Test failed\n");
181         return EXIT_FAILURE;
182     }
183 
184     cl_int status;
185 
186     // Create Program Objects
187     // Load binary from disk
188     unsigned char *kernelbinary;
189     char *xclbin = argv[1];
190 
191     //------------------------------------
192     // xclbin
193     //------------------------------------
194     printf("INFO: loading xclbin %s\n", xclbin);
195     cl_uint n_i0 = load_file_to_memory(xclbin, (char **) &kernelbinary);
196     if (n_i0 < 0) {
197         printf("ERROR: failed to load kernel from xclbin: %s\n", xclbin);
198         printf("ERROR: Test failed\n");
199         return EXIT_FAILURE;
200     }
201 
202     size_t n0 = n_i0;
203 
204     // Create the compute program from offline
205     program = clCreateProgramWithBinary(context, 1, &device_id, &n0,
206     (const unsigned char **) &kernelbinary, &status, &err);
207     free(kernelbinary);
208 
209     if ((!program) || (err!=CL_SUCCESS)) {
210         printf("ERROR: Failed to create compute program from binary %d!\n", err);
211         printf("ERROR: Test failed\n");
212         return EXIT_FAILURE;
213     }
214 
215 
216     // Build the program executable
217     //
218     err = clBuildProgram(program, 0, NULL, NULL, NULL, NULL);
219     if (err != CL_SUCCESS) {
220         size_t len;
221         char buffer[2048];
222 
223         printf("ERROR: Failed to build program executable!\n");
224         clGetProgramBuildInfo(program, device_id, CL_PROGRAM_BUILD_LOG, sizeof(buffer), buffer, &len);
225         printf("%s\n", buffer);
226         printf("ERROR: Test failed\n");
227         return EXIT_FAILURE;
228     }
229 
230     // Create the compute kernel in the program we wish to run
231     //
232     kernel = clCreateKernel(program, "rtl_kernel_wizard_0", &err);
233     if (!kernel || err != CL_SUCCESS) {
234         printf("ERROR: Failed to create compute kernel!\n");
```

```
235         printf("ERROR: Test failed\n");
236         return EXIT_FAILURE;
237     }
238
239     // Create structs to define memory bank mapping
240     cl_mem_ext_ptr_t mem_ext;
241     mem_ext.obj = NULL;
242     mem_ext.param = kernel;
243
244
245     mem_ext.flags = 1;
246     d_axi00_ptr0 = clCreateBuffer(context, CL_MEM_READ_WRITE | CL_MEM_EXT_PTR_XILINX, sizeof(cl_uint)
            * number_of_words, &mem_ext, &err);
247     if (err != CL_SUCCESS) {
248         std::cout << "Return code for clCreateBuffer flags=" << mem_ext.flags << ": " << err << std::
                endl;
249     }
250
251
252     if (!(d_axi00_ptr0)) {
253         printf("ERROR: Failed to allocate device memory!\n");
254         printf("ERROR: Test failed\n");
255         return EXIT_FAILURE;
256     }
257
258
259     err = clEnqueueWriteBuffer(commands, d_axi00_ptr0, CL_TRUE, 0, sizeof(cl_uint) * number_of_words,
            h_data, 0, NULL, NULL);
260     if (err != CL_SUCCESS) {
261         printf("ERROR: Failed to write to source array h_data: d_axi00_ptr0: %d!\n", err);
262         printf("ERROR: Test failed\n");
263         return EXIT_FAILURE;
264     }
265
266
267     // Set the arguments to our compute kernel
268     // cl_uint vector_length = MAX_LENGTH;
269     err = 0;
270     cl_uint d_scalar00 = 0;
271     err |= clSetKernelArg(kernel, 0, sizeof(cl_uint), &d_scalar00); // Not used in example RTL logic.
272     err |= clSetKernelArg(kernel, 1, sizeof(cl_mem), &d_axi00_ptr0);
273
274     if (err != CL_SUCCESS) {
275         printf("ERROR: Failed to set kernel arguments! %d\n", err);
276         printf("ERROR: Test failed\n");
277         return EXIT_FAILURE;
278     }
279
280     size_t global[1];
281     size_t local[1];
282     // Execute the kernel over the entire range of our 1d input data set
283     // using the maximum number of work group items for this device
284
285     global[0] = 1;
286     local[0] = 1;
287     err = clEnqueueNDRangeKernel(commands, kernel, 1, NULL, (size_t*)&global, (size_t*)&local, 0, NULL,
            NULL);
288     if (err) {
289         printf("ERROR: Failed to execute kernel! %d\n", err);
290         printf("ERROR: Test failed\n");
291         return EXIT_FAILURE;
292     }
293
294     clFinish(commands);
295
296
297     // Read back the results from the device to verify the output
298     //
299     cl_event readevent;
300
301     err = 0;
302     err |= clEnqueueReadBuffer( commands, d_axi00_ptr0, CL_TRUE, 0, sizeof(cl_uint) * number_of_words,
            h_axi00_ptr0_output, 0, NULL, &readevent );
303
304     if (err != CL_SUCCESS) {
305         printf("ERROR: Failed to read output array! %d\n", err);
306         printf("ERROR: Test failed\n");
307         return EXIT_FAILURE;
308     }
309     clWaitForEvents(1, &readevent);
310
311     // Check Results
```

```
312
313     for (cl_uint i = 0; i < number_of_words; i++) {
314         if ((h_data[i] * 4 - 16) != h_axi00_ptr0_output[i]) {
315             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]);
316             check_status = 1;
317         }
318         //printf("i=%d, input=%d, output=%d\n", i,  h_axi00_ptr0_input[i] h_axi00_ptr0_output[i]);
319     }
320
321
322     //------------------------------------------
323     // Shutdown and cleanup
324     //------------------------------------------
325     clReleaseMemObject(d_axi00_ptr0);
326     free(h_axi00_ptr0_output);
327
328     free(h_data);
329     clReleaseProgram(program);
330     clReleaseKernel(kernel);
331     clReleaseCommandQueue(commands);
332     clReleaseContext(context);
333
334     if (check_status) {
335         printf("ERROR: Test failed\n");
336         return EXIT_FAILURE;
337     } else {
338         printf("INFO: Test completed successfully.\n");
339         return EXIT_SUCCESS;
340     }
341
342
343 } // end of main
```

Приложение Б

Листинг 2 – Содержимое файла v++*.log

```
INFO: [v++ 60-1306] Additional information associated with this v++ link can be found at:
Reports: /iu_home/iu7136/_x/reports/link
Log files: /iu_home/iu7136/_x/logs/link
INFO: [v++ 60-1548] Creating build summary session with primary output /iu_home/iu7136/workspace/vinc.xclbin.link_summary, at Sat Dec 25 22:16:51 2021
INFO: [v++ 60-1316] Initiating connection to rulecheck server, at Sat Dec 25 22:16:51 2021
INFO: [v++ 60-1315] Creating rulecheck session with output '/iu_home/iu7136/_x/reports/link/v++_link_vinc_guidance.html', at Sat Dec 25 22:17:08 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] [22:17:52] Run run_link: Step system_link: Started
INFO: [v++ 60-1453] Command Line: system_link --xo /iu_home/iu7136/workspace/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0.xo --config /iu_home/iu7136/_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/iu7136/_x/link/int --temp_dir /iu_home/iu7136/_x/link/sys_link
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_x/link/run_link
INFO: [SYSTEM_LINK 60-1316] Initiating connection to rulecheck server, at Sat Dec 25 22:18:03 2021
INFO: [SYSTEM_LINK 82-70] Extracting xo v3 file /iu_home/iu7136/workspace/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0.xo
INFO: [SYSTEM_LINK 82-53] Creating IP database /iu_home/iu7136/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml
INFO: [SYSTEM_LINK 82-38] [22:18:05] build_xd_ip_db started: /data/Xilinx/Vitis/2020.2/bin/build_xd_ip_db -ip_search 0  -sds-pf /iu_home/iu7136/_x/link/sys_link/xilinx_u200_xdma_201830_2.hpfm -clkid 0 -ip /iu_home/iu7136/_x/link/sys_link/iprepo/mycompany_com_kernel_rtl_kernel_wizard_0_1_0,rtl_kernel_wizard_0 -o /iu_home/iu7136/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml
INFO: [SYSTEM_LINK 82-37] [22:18:31] build_xd_ip_db finished successfully
Time (s): cpu = 00:00:27 ; elapsed = 00:00:26 . Memory (MB): peak = 1557.895 ; gain = 0.000 ; free physical = 207946 ; free virtual = 372608
INFO: [SYSTEM_LINK 82-51] Create system connectivity graph
INFO: [SYSTEM_LINK 82-102] Applying explicit connections to the system connectivity graph: /iu_home/iu7136/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml
INFO: [SYSTEM_LINK 82-38] [22:18:31] cfgen started: /data/Xilinx/Vitis/2020.2/bin/cfgen  -nk rtl_kernel_wizard_0:1:vinc0 -slr vinc0:SLR2 -sp vinc0.m00_axi:DDR[3] -dmclkid 0 -r /iu_home/iu7136/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml -o /iu_home/iu7136/_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[3]
INFO: [CFGEN 83-0] SLR Specs:
INFO: [CFGEN 83-0]   instance: vinc0, SLR: SLR2
INFO: [CFGEN 83-2228] Creating mapping for argument vinc0.axi00_ptr0 to DDR[3] for directive vinc0.m00_axi:DDR[3]
INFO: [SYSTEM_LINK 82-37] [22:18:53] cfgen finished successfully
Time (s): cpu = 00:00:22 ; elapsed = 00:00:22 . Memory (MB): peak = 1557.895 ; gain = 0.000 ; free physical = 207787 ; free virtual = 372450
INFO: [SYSTEM_LINK 82-52] Create top-level block diagram
INFO: [SYSTEM_LINK 82-38] [22:18:53] cf2bd started: /data/Xilinx/Vitis/2020.2/bin/cf2bd  --linux --trace_buffer 1024 --input_file /iu_home/iu7136/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml --ip_db /iu_home/iu7136/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml --cf_name dr --working_dir /iu_home/iu7136/_x/link/sys_link/_sysl/.xsd --temp_dir /iu_home/iu7136/_x/link/sys_link --output_dir /iu_home/iu7136/_x/link/int --target_bd pfm_dynamic.bd
INFO: [CF2BD 82-31] Launching cf2xd: cf2xd -linux -trace-buffer 1024 -i /iu_home/iu7136/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml -r /iu_home/iu7136/_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/iu7136/_x/link/sys_link/_sysl/.xsd
INFO: [CF2BD 82-28] cf_xsd finished successfully
INFO: [SYSTEM_LINK 82-37] [22:19:06] cf2bd finished successfully
Time (s): cpu = 00:00:10 ; elapsed = 00:00:13 . Memory (MB): peak = 1557.895 ; gain = 0.000 ; free physical = 207580 ; free virtual = 372247
INFO: [v++ 60-1441] [22:19:06] Run run_link: Step system_link: Completed
```

```
Time (s): cpu = 00:01:13 ; elapsed = 00:01:14 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 207612 ; free virtual = 372276
INFO: [v++ 60-1443] [22:19:06] Run run_link: Step cf2sw: Started
INFO: [v++ 60-1453] Command Line: cf2sw -sdsl /iu_home/iu7136/_x/link/int/sdsl.dat -rtd /iu_home/iu7136/_x/link/int/cf2sw.rtd -nofilter /iu_home/iu7136/_x/link/int/cf2sw_full.rtd -xclbin /iu_home/iu7136/_x/link/int/xclbin_orig.xml -o /iu_home/iu7136/_x/link/int/xclbin_orig.1.xml
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_x/link/run_link
INFO: [v++ 60-1441] [22:19:21] Run run_link: Step cf2sw: Completed
Time (s): cpu = 00:00:14 ; elapsed = 00:00:15 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 207583 ; free virtual = 372246
INFO: [v++ 60-1443] [22:19:21] Run run_link: Step rtd2_system_diagram: Started
INFO: [v++ 60-1453] Command Line: rtd2SystemDiagram
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_x/link/run_link
INFO: [v++ 60-1441] [22:19:31] Run run_link: Step rtd2_system_diagram: Completed
Time (s): cpu = 00:00:00.02 ; elapsed = 00:00:09 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 207034 ; free virtual = 371697
INFO: [v++ 60-1443] [22:19:31] 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/iu7136/.ipcache --output_dir /iu_home/iu7136/_x/link/int --log_dir /iu_home/iu7136/_x/logs/link --report_dir /iu_home/iu7136/_x/reports/link --config /iu_home/iu7136/_x/link/int/vplConfig.ini -k /iu_home/iu7136/_x/link/int/kernel_info.dat --webtalk_flag Vitis --temp_dir /iu_home/iu7136/_x/link --no-info --iprepo /iu_home/iu7136/_x/link/int/xo/ip_repo/mycompany_com_kernel_rtl_kernel_wizard_0_1_0 --messageDb /iu_home/iu7136/_x/link/run_link/vpl.pb /iu_home/iu7136/_x/link/int/dr.bd.tcl
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_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/iu7136/_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/iu7136/_x/link/vivado/vpl/.local/hw_platform
WARNING: /data/Xilinx/Vitis/2020.2/tps/lnx64/jre9.0.4 does not exist.
[22:24:19] Run vpl: Step create_project: Started
Creating Vivado project.
[22:24:39] Run vpl: Step create_project: RUNNING...
[22:24:48] Run vpl: Step create_project: Completed
[22:24:48] Run vpl: Step create_bd: Started
[22:26:30] Run vpl: Step create_bd: RUNNING...
[22:28:08] Run vpl: Step create_bd: RUNNING...
[22:29:42] Run vpl: Step create_bd: RUNNING...
[22:31:31] Run vpl: Step create_bd: RUNNING...
[22:32:57] Run vpl: Step create_bd: RUNNING...
[22:33:47] Run vpl: Step create_bd: Completed
[22:33:47] Run vpl: Step update_bd: Started
[22:33:50] Run vpl: Step update_bd: Completed
[22:33:50] Run vpl: Step generate_target: Started
[22:35:29] Run vpl: Step generate_target: RUNNING...
[22:37:10] Run vpl: Step generate_target: RUNNING...
[22:38:42] Run vpl: Step generate_target: RUNNING...
[22:40:20] Run vpl: Step generate_target: RUNNING...
[22:41:54] Run vpl: Step generate_target: RUNNING...
[22:43:36] Run vpl: Step generate_target: RUNNING...
[22:44:17] Run vpl: Step generate_target: Completed
[22:44:17] Run vpl: Step config_hw_runs: Started
[22:45:33] Run vpl: Step config_hw_runs: Completed
[22:45:33] Run vpl: Step synth: Started
[22:47:42] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:48:20] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:48:58] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:49:43] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:50:29] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:51:10] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:51:54] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:52:34] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:53:16] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:53:59] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:54:41] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
[22:55:22] Block-level synthesis in progress, 5 of 66 jobs complete, 3 jobs running.
[22:56:01] Block-level synthesis in progress, 6 of 66 jobs complete, 2 jobs running.
[22:56:43] Block-level synthesis in progress, 6 of 66 jobs complete, 7 jobs running.
[22:57:23] Block-level synthesis in progress, 6 of 66 jobs complete, 8 jobs running.
[22:58:04] Block-level synthesis in progress, 8 of 66 jobs complete, 6 jobs running.
[22:58:47] Block-level synthesis in progress, 9 of 66 jobs complete, 7 jobs running.
[22:59:29] Block-level synthesis in progress, 10 of 66 jobs complete, 6 jobs running.
[23:00:09] Block-level synthesis in progress, 10 of 66 jobs complete, 7 jobs running.
[23:00:50] Block-level synthesis in progress, 10 of 66 jobs complete, 8 jobs running.
[23:01:30] Block-level synthesis in progress, 10 of 66 jobs complete, 8 jobs running.
[23:02:11] Block-level synthesis in progress, 10 of 66 jobs complete, 8 jobs running.
```

```
[23:02:52] Block-level synthesis in progress, 10 of 66 jobs complete, 8 jobs running.
[23:03:35] Block-level synthesis in progress, 10 of 66 jobs complete, 8 jobs running.
[23:04:16] Block-level synthesis in progress, 10 of 66 jobs complete, 8 jobs running.
[23:04:57] Block-level synthesis in progress, 13 of 66 jobs complete, 5 jobs running.
[23:05:37] Block-level synthesis in progress, 14 of 66 jobs complete, 4 jobs running.
[23:06:18] Block-level synthesis in progress, 14 of 66 jobs complete, 7 jobs running.
[23:06:57] Block-level synthesis in progress, 15 of 66 jobs complete, 7 jobs running.
[23:07:35] Block-level synthesis in progress, 17 of 66 jobs complete, 5 jobs running.
[23:08:15] Block-level synthesis in progress, 18 of 66 jobs complete, 5 jobs running.
[23:08:53] Block-level synthesis in progress, 18 of 66 jobs complete, 6 jobs running.
[23:09:32] Block-level synthesis in progress, 19 of 66 jobs complete, 7 jobs running.
[23:10:10] Block-level synthesis in progress, 19 of 66 jobs complete, 7 jobs running.
[23:10:54] Block-level synthesis in progress, 19 of 66 jobs complete, 8 jobs running.
[23:11:32] Block-level synthesis in progress, 20 of 66 jobs complete, 7 jobs running.
[23:12:17] Block-level synthesis in progress, 20 of 66 jobs complete, 7 jobs running.
[23:12:57] Block-level synthesis in progress, 20 of 66 jobs complete, 8 jobs running.
[23:13:40] Block-level synthesis in progress, 20 of 66 jobs complete, 8 jobs running.
[23:14:19] Block-level synthesis in progress, 21 of 66 jobs complete, 7 jobs running.
[23:15:07] Block-level synthesis in progress, 22 of 66 jobs complete, 6 jobs running.
[23:15:47] Block-level synthesis in progress, 23 of 66 jobs complete, 5 jobs running.
[23:16:31] Block-level synthesis in progress, 24 of 66 jobs complete, 6 jobs running.
[23:17:09] Block-level synthesis in progress, 26 of 66 jobs complete, 6 jobs running.
[23:17:52] Block-level synthesis in progress, 28 of 66 jobs complete, 4 jobs running.
[23:18:29] Block-level synthesis in progress, 28 of 66 jobs complete, 6 jobs running.
[23:19:10] Block-level synthesis in progress, 28 of 66 jobs complete, 8 jobs running.
[23:19:48] Block-level synthesis in progress, 31 of 66 jobs complete, 5 jobs running.
[23:20:36] Block-level synthesis in progress, 31 of 66 jobs complete, 6 jobs running.
[23:21:15] Block-level synthesis in progress, 31 of 66 jobs complete, 8 jobs running.
[23:21:58] Block-level synthesis in progress, 34 of 66 jobs complete, 5 jobs running.
[23:22:39] Block-level synthesis in progress, 34 of 66 jobs complete, 5 jobs running.
[23:23:31] Block-level synthesis in progress, 34 of 66 jobs complete, 8 jobs running.
[23:24:11] Block-level synthesis in progress, 34 of 66 jobs complete, 8 jobs running.
[23:24:57] Block-level synthesis in progress, 35 of 66 jobs complete, 7 jobs running.
[23:25:38] Block-level synthesis in progress, 35 of 66 jobs complete, 7 jobs running.
[23:26:20] Block-level synthesis in progress, 37 of 66 jobs complete, 6 jobs running.
[23:26:57] Block-level synthesis in progress, 37 of 66 jobs complete, 6 jobs running.
[23:27:37] Block-level synthesis in progress, 37 of 66 jobs complete, 8 jobs running.
[23:28:15] Block-level synthesis in progress, 37 of 66 jobs complete, 8 jobs running.
[23:28:58] Block-level synthesis in progress, 38 of 66 jobs complete, 7 jobs running.
[23:29:36] Block-level synthesis in progress, 38 of 66 jobs complete, 7 jobs running.
[23:30:19] Block-level synthesis in progress, 38 of 66 jobs complete, 8 jobs running.
[23:30:56] Block-level synthesis in progress, 40 of 66 jobs complete, 6 jobs running.
[23:31:36] Block-level synthesis in progress, 40 of 66 jobs complete, 6 jobs running.
[23:32:15] Block-level synthesis in progress, 40 of 66 jobs complete, 8 jobs running.
[23:32:56] Block-level synthesis in progress, 40 of 66 jobs complete, 8 jobs running.
[23:33:34] Block-level synthesis in progress, 40 of 66 jobs complete, 8 jobs running.
[23:34:18] Block-level synthesis in progress, 41 of 66 jobs complete, 7 jobs running.
[23:34:56] Block-level synthesis in progress, 43 of 66 jobs complete, 5 jobs running.
[23:35:39] Block-level synthesis in progress, 43 of 66 jobs complete, 8 jobs running.
[23:36:16] Block-level synthesis in progress, 43 of 66 jobs complete, 8 jobs running.
[23:36:58] Block-level synthesis in progress, 44 of 66 jobs complete, 7 jobs running.
[23:37:36] Block-level synthesis in progress, 44 of 66 jobs complete, 7 jobs running.
[23:38:21] Block-level synthesis in progress, 45 of 66 jobs complete, 7 jobs running.
[23:38:59] Block-level synthesis in progress, 45 of 66 jobs complete, 7 jobs running.
[23:39:43] Block-level synthesis in progress, 46 of 66 jobs complete, 7 jobs running.
[23:40:22] Block-level synthesis in progress, 46 of 66 jobs complete, 7 jobs running.
[23:41:06] Block-level synthesis in progress, 46 of 66 jobs complete, 8 jobs running.
[23:41:50] Block-level synthesis in progress, 46 of 66 jobs complete, 8 jobs running.
[23:42:36] Block-level synthesis in progress, 46 of 66 jobs complete, 8 jobs running.
[23:43:19] Block-level synthesis in progress, 46 of 66 jobs complete, 8 jobs running.
[23:44:09] Block-level synthesis in progress, 48 of 66 jobs complete, 6 jobs running.
[23:44:50] Block-level synthesis in progress, 48 of 66 jobs complete, 6 jobs running.
[23:45:38] Block-level synthesis in progress, 49 of 66 jobs complete, 7 jobs running.
[23:46:17] Block-level synthesis in progress, 49 of 66 jobs complete, 7 jobs running.
[23:47:01] Block-level synthesis in progress, 50 of 66 jobs complete, 7 jobs running.
[23:47:41] Block-level synthesis in progress, 52 of 66 jobs complete, 5 jobs running.
[23:48:23] Block-level synthesis in progress, 53 of 66 jobs complete, 5 jobs running.
[23:49:02] Block-level synthesis in progress, 54 of 66 jobs complete, 6 jobs running.
[23:49:46] Block-level synthesis in progress, 55 of 66 jobs complete, 6 jobs running.
[23:50:24] Block-level synthesis in progress, 56 of 66 jobs complete, 6 jobs running.
[23:51:07] Block-level synthesis in progress, 56 of 66 jobs complete, 7 jobs running.
[23:51:51] Block-level synthesis in progress, 56 of 66 jobs complete, 7 jobs running.
[23:52:32] Block-level synthesis in progress, 56 of 66 jobs complete, 7 jobs running.
[23:53:11] Block-level synthesis in progress, 56 of 66 jobs complete, 7 jobs running.
[23:53:52] Block-level synthesis in progress, 57 of 66 jobs complete, 6 jobs running.
[23:54:34] Block-level synthesis in progress, 57 of 66 jobs complete, 6 jobs running.
[23:55:17] Block-level synthesis in progress, 57 of 66 jobs complete, 6 jobs running.
[23:55:57] Block-level synthesis in progress, 57 of 66 jobs complete, 6 jobs running.
[23:56:41] Block-level synthesis in progress, 57 of 66 jobs complete, 6 jobs running.
[23:57:22] Block-level synthesis in progress, 58 of 66 jobs complete, 5 jobs running.
[23:58:04] Block-level synthesis in progress, 60 of 66 jobs complete, 3 jobs running.
[23:58:43] Block-level synthesis in progress, 60 of 66 jobs complete, 3 jobs running.
```

```
196 [23:59:24] Block-level synthesis in progress, 60 of 66 jobs complete, 5 jobs running.
197 [00:00:03] Block-level synthesis in progress, 61 of 66 jobs complete, 4 jobs running.
198 [00:00:47] Block-level synthesis in progress, 62 of 66 jobs complete, 3 jobs running.
199 [00:01:27] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
200 [00:02:11] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
201 [00:02:50] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
202 [00:03:34] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
203 [00:04:13] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
204 [00:04:59] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
205 [00:05:39] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
206 [00:06:32] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
207 [00:07:12] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
208 [00:07:56] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
209 [00:08:37] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
210 [00:09:25] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
211 [00:10:06] Block-level synthesis in progress, 63 of 66 jobs complete, 2 jobs running.
212 [00:11:02] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
213 [00:11:41] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
214 [00:12:24] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
215 [00:13:05] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
216 [00:13:54] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
217 [00:14:33] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
218 [00:15:21] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
219 [00:16:05] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
220 [00:16:51] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
221 [00:17:40] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
222 [00:18:29] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
223 [00:19:07] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
224 [00:19:53] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
225 [00:20:33] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
226 [00:21:21] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
227 [00:22:09] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
228 [00:23:08] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
229 [00:23:49] Block-level synthesis in progress, 64 of 66 jobs complete, 1 job running.
230 [00:24:37] Block-level synthesis in progress, 65 of 66 jobs complete, 0 jobs running.
231 [00:25:19] Block-level synthesis in progress, 65 of 66 jobs complete, 0 jobs running.
232 [00:26:02] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
233 [00:26:43] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
234 [00:27:27] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
235 [00:28:08] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
236 [00:28:51] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
237 [00:29:29] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
238 [00:30:13] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
239 [00:30:52] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
240 [00:31:37] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
241 [00:32:16] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
242 [00:33:02] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
243 [00:33:42] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
244 [00:34:25] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
245 [00:35:05] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
246 [00:35:50] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
247 [00:36:33] Block-level synthesis in progress, 66 of 66 jobs complete, 0 jobs running.
248 [00:37:18] Block-level synthesis in progress, 66 of 66 jobs complete, 0 jobs running.
249 [00:37:59] Top-level synthesis in progress.
250 [00:38:41] Top-level synthesis in progress.
251 [00:39:21] Top-level synthesis in progress.
252 [00:40:06] Top-level synthesis in progress.
253 [00:40:44] Top-level synthesis in progress.
254 [00:41:27] Top-level synthesis in progress.
255 [00:42:04] Top-level synthesis in progress.
256 [00:42:48] Top-level synthesis in progress.
257 [00:43:29] Top-level synthesis in progress.
258 [00:44:15] Top-level synthesis in progress.
259 [00:44:55] Top-level synthesis in progress.
260 [00:45:39] Top-level synthesis in progress.
261 [00:46:18] Top-level synthesis in progress.
262 [00:47:03] Top-level synthesis in progress.
263 [00:47:45] Top-level synthesis in progress.
264 [00:48:47] Run vpl: Step synth: Completed
265 [00:48:47] Run vpl: Step impl: Started
266 [01:39:11] Finished 2nd of 6 tasks (FPGA linking synthesized kernels to platform). Elapsed time: 03h 19
        m 31s
267
268 [01:39:11] Starting logic optimization..
269 [01:44:43] Phase 1 Generate And Synthesize MIG Cores
270 [02:18:03] Phase 2 Generate And Synthesize Debug Cores
271 [02:41:11] Phase 3 Retarget
272 [02:43:10] Phase 4 Constant propagation
273 [02:45:09] Phase 5 Sweep
274 [02:49:45] Phase 6 BUFG optimization
275 [02:51:44] Phase 7 Shift Register Optimization
276 [02:52:27] Phase 8 Post Processing Netlist
```

```
[03:08:04] Finished 3rd of 6 tasks (FPGA logic optimization). Elapsed time: 01h 28m 53s

[03:08:04] Starting logic placement..
[03:13:00] Phase 1 Placer Initialization
[03:13:00] Phase 1.1 Placer Initialization Netlist Sorting
[03:27:17] Phase 1.2 IO Placement/ Clock Placement/ Build Placer Device
[03:37:14] Phase 1.3 Build Placer Netlist Model
[03:50:15] Phase 1.4 Constrain Clocks/Macros
[03:51:42] Phase 2 Global Placement
[03:51:42] Phase 2.1 Floorplanning
[03:55:05] Phase 2.1.1 Partition Driven Placement
[03:55:05] Phase 2.1.1.1 PBP: Partition Driven Placement
[03:57:10] Phase 2.1.1.2 PBP: Clock Region Placement
[04:02:07] Phase 2.1.1.3 PBP: Compute Congestion
[04:02:07] Phase 2.1.1.4 PBP: UpdateTiming
[04:04:45] Phase 2.1.1.5 PBP: Add part constraints
[04:05:28] Phase 2.2 Update Timing before SLR Path Opt
[04:05:28] Phase 2.3 Global Placement Core
[04:36:36] Phase 2.3.1 Physical Synthesis In Placer
[04:49:43] Phase 3 Detail Placement
[04:49:43] Phase 3.1 Commit Multi Column Macros
[04:49:43] Phase 3.2 Commit Most Macros & LUTRAMs
[04:55:15] Phase 3.3 Small Shape DP
[04:55:15] Phase 3.3.1 Small Shape Clustering
[04:57:18] Phase 3.3.2 Flow Legalize Slice Clusters
[04:58:17] Phase 3.3.3 Slice Area Swap
[05:02:32] Phase 3.4 Place Remaining
[05:03:09] Phase 3.5 Re-assign LUT pins
[05:05:12] Phase 3.6 Pipeline Register Optimization
[05:05:12] Phase 3.7 Fast Optimization
[05:09:09] Phase 4 Post Placement Optimization and Clean-Up
[05:09:09] Phase 4.1 Post Commit Optimization
[05:18:29] Phase 4.1.1 Post Placement Optimization
[05:19:07] Phase 4.1.1.1 BUFG Insertion
[05:19:07] Phase 1 Physical Synthesis Initialization
[05:21:49] Phase 4.1.1.2 BUFG Replication
[05:25:17] Phase 4.1.1.3 Replication
[05:31:16] Phase 4.2 Post Placement Cleanup
[05:31:59] Phase 4.3 Placer Reporting
[05:31:59] Phase 4.3.1 Print Estimated Congestion
[05:34:01] Phase 4.4 Final Placement Cleanup
[06:40:35] Finished 4th of 6 tasks (FPGA logic placement). Elapsed time: 03h 32m 30s

[06:40:35] Starting logic routing..
[06:45:59] Phase 1 Build RT Design
[06:56:42] Phase 2 Router Initialization
[06:56:42] Phase 2.1 Fix Topology Constraints
[06:57:26] Phase 2.2 Pre Route Cleanup
[06:58:03] Phase 2.3 Global Clock Net Routing
[07:00:51] Phase 2.4 Update Timing
[07:13:21] Phase 2.5 Update Timing for Bus Skew
[07:13:21] Phase 2.5.1 Update Timing
[07:18:50] Phase 3 Initial Routing
[07:18:50] Phase 3.1 Global Routing
[07:23:38] Phase 4 Rip-up And Reroute
[07:23:38] Phase 4.1 Global Iteration 0
[07:46:32] Phase 4.2 Global Iteration 1
[07:52:29] Phase 4.3 Global Iteration 2
[07:56:35] Phase 5 Delay and Skew Optimization
[07:56:35] Phase 5.1 Delay CleanUp
[07:56:35] Phase 5.1.1 Update Timing
[08:03:27] Phase 5.2 Clock Skew Optimization
[08:04:05] Phase 6 Post Hold Fix
[08:04:05] Phase 6.1 Hold Fix Iter
[08:04:05] Phase 6.1.1 Update Timing
[08:09:27] Phase 7 Route finalize
[08:10:09] Phase 8 Verifying routed nets
[08:10:47] Phase 9 Depositing Routes
[08:14:50] Phase 10 Route finalize
[08:14:50] Phase 11 Post Router Timing
[08:22:13] Finished 5th of 6 tasks (FPGA routing). Elapsed time: 01h 41m 37s

[08:22:13] Starting bitstream generation..
[10:12:54] Creating bitmap...
[11:00:45] Writing bitstream ./pfm_top_i_dynamic_region_my_rm_partial.bit...
[11:01:30] Finished 6th of 6 tasks (FPGA bitstream generation). Elapsed time: 02h 39m 17s
[11:04:59] Run vpl: Step impl: Completed
[11:05:14] Run vpl: FINISHED. Run Status: impl Complete!
INFO: [v++ 60-1441] [11:05:42] Run run_link: Step vpl: Completed
Time (s): cpu = 00:50:00 ; elapsed = 12:46:11 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free
     physical = 120138 ; free virtual = 311996
INFO: [v++ 60-1443] [11:05:42] Run run_link: Step rtdgen: Started
```

```
INFO: [v++ 60-1453] Command Line: rtdgen
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_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/iu7136/_x/link/int/address_map.xml -sdsl /iu_home/iu7136/_x/link/int/sdsl.dat -xclbin /iu_home/iu7136/_x/link/int/xclbin_orig.xml -rtd /iu_home/iu7136/_x/link/int/vinc.rtd -o /iu_home/iu7136/_x/link/int/vinc.xml
INFO: [v++ 60-1652] Cf2sw returned exit code: 0
INFO: [v++ 60-2311] HPISystemDiagram::writeSystemDiagramAfterRunningVivado, rtdInputFilePath: /iu_home/iu7136/_x/link/int/vinc.rtd
INFO: [v++ 60-2312] HPISystemDiagram::writeSystemDiagramAfterRunningVivado, systemDiagramOutputFilePath: /iu_home/iu7136/_x/link/int/systemDiagramModelSlrBaseAddress.json
INFO: [v++ 60-1618] Launching
INFO: [v++ 60-1441] [11:05:56] Run run_link: Step rtdgen: Completed
Time (s): cpu = 00:00:13 ; elapsed = 00:00:14 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 120125 ; free virtual = 311985
INFO: [v++ 60-1443] [11:05:56] Run run_link: Step xclbinutil: Started
INFO: [v++ 60-1453] Command Line: xclbinutil --add-section DEBUG_IP_LAYOUT:JSON:/iu_home/iu7136/_x/link/int/debug_ip_layout.rtd --add-section BITSTREAM:RAW:/iu_home/iu7136/_x/link/int/partial.bit --force --target hw --key-value SYS:dfx_enable:true --add-section :JSON:/iu_home/iu7136/_x/link/int/vinc.rtd --append-section :JSON:/iu_home/iu7136/_x/link/int/appendSection.rtd --add-section CLOCK_FREQ_TOPOLOGY:JSON:/iu_home/iu7136/_x/link/int/vinc_xml.rtd --add-section BUILD_METADATA:JSON:/iu_home/iu7136/_x/link/int/vinc_build.rtd --add-section EMBEDDED_METADATA:RAW:/iu_home/iu7136/_x/link/int/vinc.xml --add-section SYSTEM_METADATA:RAW:/iu_home/iu7136/_x/link/int/systemDiagramModelSlrBaseAddress.json --output /iu_home/iu7136/workspace/vinc.xclbin
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_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/iu7136/_x/link/int/debug_ip_layout.rtd'

Section: 'BITSTREAM'(0) was successfully added.
Size   : 42216130 bytes
Format : RAW
File   : '/iu_home/iu7136/_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/iu7136/_x/link/int/vinc_xml.rtd'

Section: 'BUILD_METADATA'(14) was successfully added.
Size   : 3070 bytes
Format : JSON
File   : '/iu_home/iu7136/_x/link/int/vinc_build.rtd'

Section: 'EMBEDDED_METADATA'(2) was successfully added.
Size   : 2759 bytes
Format : RAW
File   : '/iu_home/iu7136/_x/link/int/vinc.xml'

Section: 'SYSTEM_METADATA'(22) was successfully added.
Size   : 5774 bytes
Format : RAW
File   : '/iu_home/iu7136/_x/link/int/systemDiagramModelSlrBaseAddress.json'

Section: 'IP_LAYOUT'(8) was successfully appended to.
Format : JSON
File   : 'ip_layout'
Successfully wrote (42238519 bytes) to the output file: /iu_home/iu7136/workspace/vinc.xclbin
```

```
Leaving xclbinutil.
INFO: [v++ 60-1441] [11:05:58] Run run_link: Step xclbinutil: Completed
Time (s): cpu = 00:00:00.47 ; elapsed = 00:00:02 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 120036 ; free virtual = 311938
INFO: [v++ 60-1443] [11:05:58] Run run_link: Step xclbinutilinfo: Started
INFO: [v++ 60-1453] Command Line: xclbinutil --quiet --force --info /iu_home/iu7136/workspace/vinc.xclbin.info --input /iu_home/iu7136/workspace/vinc.xclbin
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_x/link/run_link
INFO: [v++ 60-1441] [11:06:01] Run run_link: Step xclbinutilinfo: Completed
Time (s): cpu = 00:00:02 ; elapsed = 00:00:03 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 120075 ; free virtual = 311977
INFO: [v++ 60-1443] [11:06:01] Run run_link: Step generate_sc_driver: Started
INFO: [v++ 60-1453] Command Line:
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7136/_x/link/run_link
INFO: [v++ 60-1441] [11:06:01] Run run_link: Step generate_sc_driver: Completed
Time (s): cpu = 00:00:00 ; elapsed = 00:00:00.05 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 120056 ; free virtual = 311958
INFO: [v++ 60-244] Generating system estimate report...
INFO: [v++ 60-1092] Generated system estimate report: /iu_home/iu7136/_x/reports/link/system_estimate_vinc.xtxt
INFO: [v++ 60-586] Created /iu_home/iu7136/workspace/vinc.ltx
INFO: [v++ 60-586] Created /iu_home/iu7136/workspace/vinc.xclbin
INFO: [v++ 60-1307] Run completed. Additional information can be found in:
Guidance: /iu_home/iu7136/_x/reports/link/v++_link_vinc_guidance.html
Timing Report: /iu_home/iu7136/_x/reports/link/imp/impl_1_xilinx_u200_xdma_201830_2_bb_locked_timing_summary_routed.rpt
Vivado Log: /iu_home/iu7136/_x/logs/link/vivado.log
Steps Log File: /iu_home/iu7136/_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/iu7136/workspace/vinc.xclbin.link_summary
INFO: [v++ 60-791] Total elapsed time: 12h 49m 34s
```

Приложение В

Листинг 3 – Содержимое файла vinc_xclbin_info

```
==========================================================
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):           b16544ed-74b8-4bc5-822a-5481923f7bf1
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:      No

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:      Yes

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 scalar00, 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:  0x1e00000

Argument:           scalar00
Register Offset:    0x010
Port:               s_axi_control
Memory:             <not applicable>

Argument:           axi00_ptr0
Register Offset:    0x018
Port:               m00_axi
Memory:             bank3 (MEM_DDR4)
================================================================
Generated By
------------
Command:        v++
Version:        2020.2 - 2020-11-18-05:13:29 (SW BUILD: 0)
Command Line:   v++ --config /iu_home/iu7136/workspace/Alveo_lab1/Alveo_lab1.cfg --connectivity.nk
     rtl_kernel_wizard_0:1:vinc0 --connectivity.slr vinc0:SLR2 --connectivity.sp vinc0.m00_axi:DDR[3]
     --input_files /iu_home/iu7136/workspace/Alveo_lab1_kernels/vivado_rtl_kernel/
     rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0.xo --link --optimize 0 --output /iu_home/iu7136
     /workspace/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/iu7136/workspace/Alveo_lab1/Alveo_lab1.cfg
--connectivity.nk rtl_kernel_wizard_0:1:vinc0
--connectivity.slr vinc0:SLR2
--connectivity.sp vinc0.m00_axi:DDR[3]
--input_files /iu_home/iu7136/workspace/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/
     exports/rtl_kernel_wizard_0.xo
--link
--optimize 0
--output /iu_home/iu7136/workspace/vinc.xclbin
--platform xilinx_u200_xdma_201830_2
--report_level 0
```

```
147 --target hw
148 --vivado.prop run.impl_1.STEPS.OPT_DESIGN.ARGS.DIRECTIVE=Explore
149 --vivado.prop run.impl_1.STEPS.PLACE_DESIGN.ARGS.DIRECTIVE=Explore
150 --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.IS_ENABLED=true
151 --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.ARGS.DIRECTIVE=AggressiveExplore
152 --vivado.prop run.impl_1.STEPS.ROUTE_DESIGN.ARGS.DIRECTIVE=Explore
153 ------------------------------------------------------------
154 User Added Key Value Pairs
155 ------------------------------
156 <empty>
157 ------------------------------------------------------------
```