СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Целью данной работы является изучение архитектуры гетерогенных вычислительных систем и технологии разработки ускорителей вычислений на базе ПЛИС фирмы Xilinx.

Для достижения данной цели необходимо выполнить следующие задачи:

- изучить основные сведения о платформе Xilinx Alveo U200;
- разработать RTL[1] 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.

[1]Register Transfer Language — язык регистровых передач

1.1 Функциональная схема разрабатываемой аппаратной системы

1.2 Изучение работа шины AXI

В соответствии с 11 вариантом требовалось реализовать функцию в соответствии с (формулой 1.1). Константа 25 в изначальном варианте была записана в обратном порядке байтов (строка 5 листинга 1.1).

$$R[i] = 25 + A[i]/2 \tag{1.1}$$

.

Листинг 1.1 — Изменный код модуля rtl_kernel_wizard_0_example_adder.v

```
1  module rtl_kernel_wizard_0_example_adder #(
2    parameter integer C_AXIS_TDATA_WIDTH = 512, // Data width of both input
         and output data
3    parameter integer C_ADDER_BIT_WIDTH  = 32,
4    parameter integer C_NUM_CLOCKS       = 1,
5    parameter integer USER_CONSTANT      = 32'h98000000   // little endian
         25 const
6  )
7  ...
8  localparam integer LP_NUM_LOOPS = C_AXIS_TDATA_WIDTH/C_ADDER_BIT_WIDTH;
9  localparam         LP_CLOCKING_MODE = C_NUM_CLOCKS == 1 ? "common_clock" :
10 ...
11 // Register s_axis_interface/inputs
12 always @(posedge s_axis_aclk) begin
13   ...
14   d1_constant <= ctrl_constant;
15 end
16 // Adder function
17 always @(posedge s_axis_aclk) begin
18   for (i = 0; i < LP_NUM_LOOPS; i = i + 1) begin
19     d2_tdata[i*C_ADDER_BIT_WIDTH+:C_ADDER_BIT_WIDTH] <=
         d1_tdata[C_ADDER_BIT_WIDTH*i+:C_ADDER_BIT_WIDTH]/2 + USER_CONSTANT;
20   end
21 end
22 ...
23 endmodule
24 `default_nettype wire
```

Далее приведены диаграммы, иллюстрирующие процесс рукопожатия и пакетного чтения.

Чтобы указать завершение пакетного чтения и записи, устройство использует сигнал RLAST. На диаграмме этого нельзя увидеть, так как изначально было указано малое время симуляции.

Ниже на рисунке 1.1 приведена транзакция чтения данных вектора на шине AXI4 MM из DDR памяти.

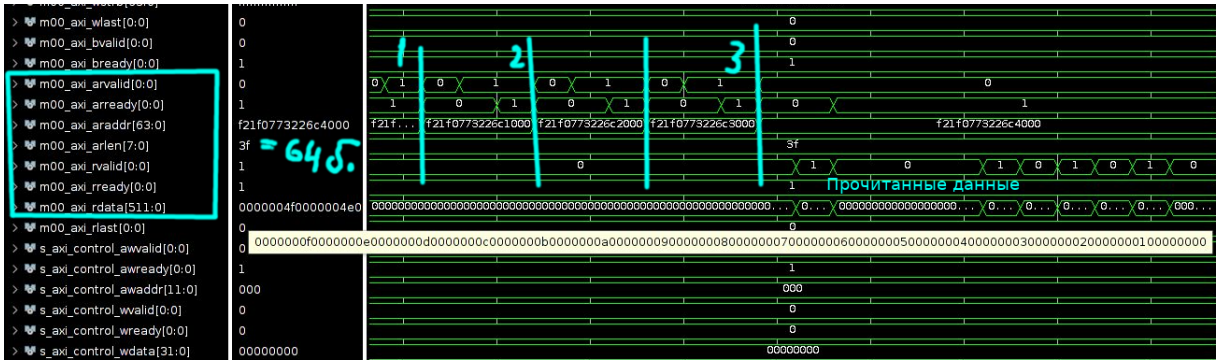

Рисунок 1.1 — Транзакция чтения данных вектора на шине AXI4 MM из DDR памяти.

Ниже на рисунке 1.2 приведена транзакция записи результата инкремента данных на шине AXI4 MM. Видно, что каждое прочитанное значение было инкрементировано.

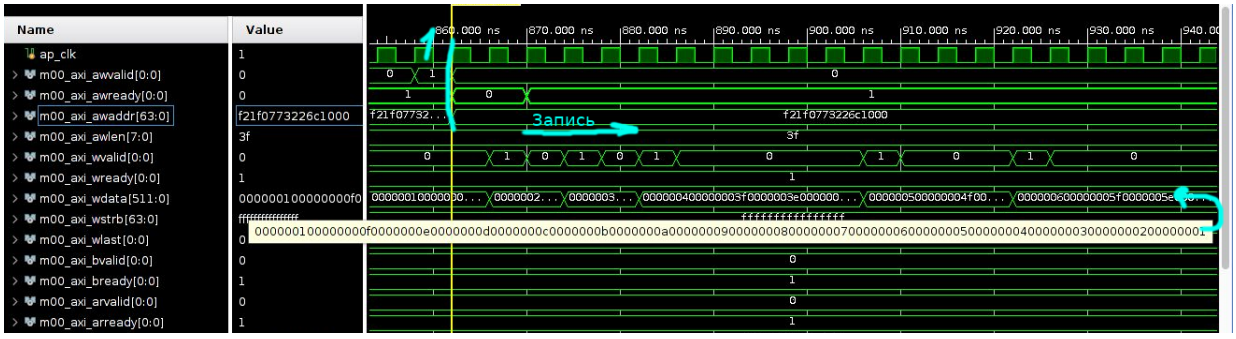

Рисунок 1.2 — Транзакция записи результата инкремента данных на шине AXI4 MM

Ниже на рисунке 1.3 приведен фрагмент кода модуля tl_kernel_wizard_0_example_adder.v (до изменения) с выполнением инкрементирования данных.

```
  dl_constant <= ctrl_constant;
end

// 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] <= dl_tdata[C_ADDER_BIT_WIDTH*i+:C_ADDER_BIT_WIDTH] + dl_constant;
  end
end

// Register inputs to fifo
```

Рисунок 1.3 — Код модуля tl_kernel_wizard_0_example_adder.v с выполнением инкрементирования данных

Теперь изменим модуль rtl_kernel_wizard_0_example_adder.v, чтобы ускоритель выполнял предложенную функцию. Ниже на рисунке 1.4 приведена транзакция чтения данных вектора на шине AXI4 MM из DDR памяти.

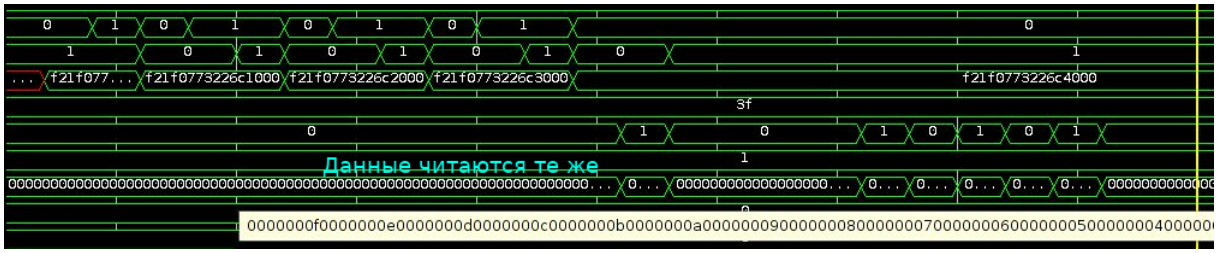

Рисунок 1.4 — Транзакция чтения данных вектора на шине AXI4 MM из DDR памяти.

Ниже на рисунке 1.5 приведена транзакция записи измененных данных.

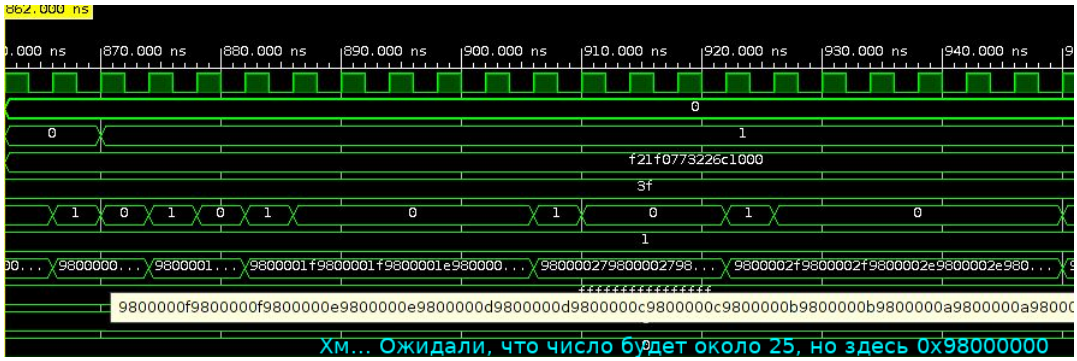

Рисунок 1.5 — Транзакция записи результата выполнения функции

К сожалению, последняя симуляция была выполнена уже после сборки проекта.

1.3 Сборка проекта

Ниже в листинге 1.2 приведено содержимое конфигурационного файла. В соответствии с вариантом требовалось использовать регионы SLR2 и DDR[2].

Листинг 1.2 — Содержимое конфигурационного файла

```
[connectivity]
nk=rtl_kernel_wizard_0:1:vinc0
slr=vinc0:SLR2
sp=vinc0.m00_axi:DDR[2]
sp=vinc0.m00_axi:PLRAM[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
```

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

1.4 Запуск программного обеспечения на хост-системе

Ниже, в листинге 1.3 приведены измененные части кода модифицированного модуля host_example.cpp.

Листинг 1.3 — Модуль host_example.cpp

```
    ...
    // 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; // Пусть присвоим элементам массива их индексы
        h_axi00_ptr0_output[i] = 0;
    }
   ...
    for (cl_uint i = 0; i < number_of_words; i++) {
        if ((h_data[i]/2 + 25) != 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]);
12              check_status = 1;
13          }
14          // printf("i=%d, input=%d, output=%d\n", i,  h_data[i]/2 + 25,
                h_axi00_ptr0_output[i]);
15      }
16      ...
17  } // end of main
```

Автору отчета не удалось, к сожалению, разобраться с графическим интерфейсом программы Xilinx Vitis IDE для настройки конфигурации отладочной сессии, поэтому решено было воспользоваться утилитой xgdb [2]. Ниже на рисунке 1.6 приведены результаты первого запуска.

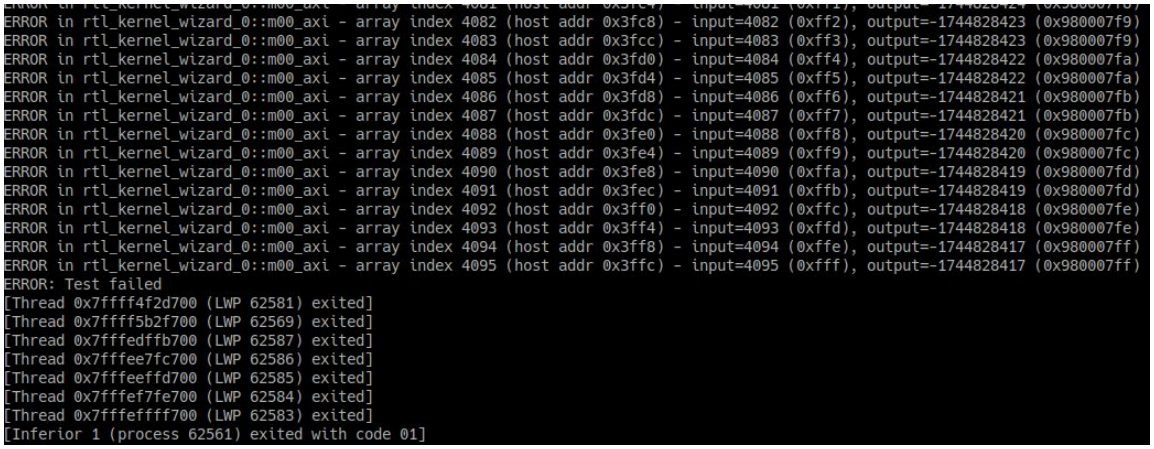

Рисунок 1.6 — Результаты тестирования

Ошибки. Много ошибок. Тестирование показало, что записывать константу 25 в обратном порядке байт не требовалось. Чтобы заного не пересобировать проект, изменил проверочное условие (листинг 1.4). Все содержимое файла приведено в приложении А.

Листинг 1.4 — Содержимое файла host_example.cpp

```
1       ...
2       for (cl_uint i = 0; i < number_of_words; i++) {
3           if ((h_data[i]/2 + 0x98000000) != h_axi00_ptr0_output[i]) {
4               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]);
5               check_status = 1;
```

```
6           }
7           // printf("i=%d, input=%d, output=%d\n", i,  h_data[i]/2 +
                0x98000000, h_axi00_ptr0_output[i]);
8       }
9       ...
10  } // end of main
```

Ниже на рисунке 1.7 приведены результаты тестирования.

```
iu7041@dl580:~/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports$ xgdb --a
rgs rtl_kernel_wizard_0_host_example.exe rtl_kernel_wizard_0_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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_e
x/exports/rtl_kernel_wizard_0_host_example.exe rtl_kernel_wizard_0_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 30718)]
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 rtl_kernel_wizard_0_vinc.xclbin
[New Thread 0x7ffff4f2d700 (LWP 30794)]
[New Thread 0x7fffefff700 (LWP 30797)]
[New Thread 0x7fffef7fe700 (LWP 30798)]
[New Thread 0x7fffeeffd700 (LWP 30800)]
[New Thread 0x7fffee7fc700 (LWP 30801)]
[New Thread 0x7fffedffb700 (LWP 30802)]
INFO: Test completed successfully.
[Thread 0x7ffff4f2d700 (LWP 30794) exited]
[Thread 0x7ffff5b2f700 (LWP 30718) exited]
[Thread 0x7fffedffb700 (LWP 30802) exited]
[Thread 0x7fffee7fc700 (LWP 30801) exited]
[Thread 0x7fffeeffd700 (LWP 30800) exited]
[Thread 0x7fffef7fe700 (LWP 30798) exited]
[Thread 0x7fffefff700 (LWP 30797) exited]
[Inferior 1 (process 30670) exited normally]
(gdb)
```

Рисунок 1.7 — Результаты тестирования

Ответы на контрольные вопросы

1. Преимущества и недостатки XDMA и QDMA платформ

Недостатки использовани XDMA:

• бОльшая латентность и меньшая пропускная способность за счет того, что данные сначало должны быть перемещены в память ускорителя.

Преимущества использования QDMA:

• предоставляет прямое потоковое соединение с низкой задержкой и большой пропускной способностью между хостом и ядрами;

• позволяет передавать поток данных непосредственно в логику FPGA параллельно с их обработкой.

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

1. С помощью вызова clGetPlatformIDs хост получает все платформы.

2. С помощью вызова clGetPlatformInfo хост получает имя платформы и затем выбирает платформу Xilinx.

3. С помощью вызова clGetDeviceIDs хост получает ID устройства.

4. С помощью вызова clGetDeviceInfo хост получает информацию об устройстве.

5. С помощью вызова clCreateContext создается контекст для переменных.

6. С помощью вызова clCreateCommandQueue создается команда для устройтво-ускорителя.

3. Процедура запуска задания на исполнения в ускорительном ядре VINC

1. С помощью вызова load_file_to_memory (см. приложение А) данные, бинарный поток (данные из *.xclbin), копируются из ОЗУ в локальную память ускорителя посредством DMA.

2. По итогу выполнения clCreateProgramWithBinary, clBuildProgram и clCreateKernel создается исполняемый файл (уже в памяти устройства-ускорителя).

3. С помощью clCreateBuffer и clEnqueueWriteBuffer данные, подлежащие обработке, копируются из ОЗУ в локальную память ускорителя посредством DMA (с помощью второй команды осуществляется передача указателей на начало буферов исходных операндов).

4. С помощью двух вызовов clSetKernelArg указываются параметры (в данном случае это d_scalar00 и d_axi00_ptr0).

5. С помощью команды clEnqueueNDRangeKernel запускается исполнение ядра (программы на ускорителе).

6. С помощью команды clEnqueueReadBuffer выполняется чтение готовых данных.

4. Процесс линковки на основании содержимого log-файла

Процесс сборки состоит из шести этапов (фаз).

1. Анализ конфигурационного файла, анализ профиля устройства, поиск необходимых аппаратных компонентов, интерфейсов;

2. FPGA linking synthesized kernels to platform (Block-level synthesis, Top-level synthesis);

3. FPGA logic optimization (минимизация логики (булевой) для оптимизации площади, минимизации задержек);

4. FPGA logic placement (Преобразование булевых уравнений в схему логики ПЛИС. Выбор конкретного места для каждого логического блока в ПЛИС);

5. FPGA routing (разводка [создание соединений между логическими блоками]);

6. FPGA bitstream generation (генерирование файла с программной информацией для отправки его на ПЛИС [*.xclbin файл]);

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Verilog Basics Part-I. [Электронный ресурс]. URL : http://www.asic-world.com/systemverilog/basic1.html (Дата обращения 14.11.2021). Текст электронный

2. Vitis Unified Software Development Platform 2021.1 Documentation . [Электронный ресурс]. URL : https://www.xilinx.com/html_docs/xilinx2021_1/vitis_doc/debuggingapplicationskernels.html#rjl1538574380183 (Дата обращения 14.11.2021). Текст электронный

ПРИЛОЖЕНИЕ А

СОДЕРЖИМОЕ ФАЙЛА HOST_EXAMPLE.CPP

Листинг А.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;
```

```
71      const cl_uint number_of_words = 4096; // 16KB of data
72
73
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; // Здесь задаются входные значения
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
305     if (err != CL_SUCCESS) {
306         printf("ERROR: Failed to read output array! %d\n", err);
307         printf("ERROR: Test failed\n");
308         return EXIT_FAILURE;
309     }
310     clWaitForEvents(1, &readevent);
311     // Check Results
312
313     for (cl_uint i = 0; i < number_of_words; i++) {
314         if ((h_data[i]/2 + 0x98000000) != 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_data[i]/2 + 0x98000000, 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
329
330     free(h_data);
331     clReleaseProgram(program);
332     clReleaseKernel(kernel);
333     clReleaseCommandQueue(commands);
334     clReleaseContext(context);
335
336     if (check_status) {
337         printf("ERROR: Test failed\n");
338         return EXIT_FAILURE;
339     } else {
340         printf("INFO: Test completed successfully.\n");
341         return EXIT_SUCCESS;
342     }
343
344
345 } // end of main
```

ПРИЛОЖЕНИЕ Б

СОДЕРЖИМОЕ LOG-ФАЙЛА

Листинг Б.1 — Содержимое log-файла

```
1  INFO: [v++ 60-1306] Additional information associated with this v++ link can be found at:
2      Reports: /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
           rtl_kernel_wizard_0_ex/exports/_x/reports/link
3      Log files: /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
           rtl_kernel_wizard_0_ex/exports/_x/logs/link
4  INFO: [v++ 60-1548] Creating build summary session with primary output
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0_vinc.xclbin.link_summary, at Sat Nov 13
       20:27:01 2021
5  INFO: [v++ 60-1316] Initiating connection to rulecheck server, at Sat Nov 13 20:27:01 2021
6  INFO: [v++ 60-1315] Creating rulecheck session with output
       '/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/reports/link/v++_link_rtl_kernel_wizard_0_vinc_guidance.html',
       at Sat Nov 13 20:27:04 2021
7  INFO: [v++ 60-895]    Target platform:
       /opt/xilinx/platforms/xilinx_u200_xdma_201830_2/xilinx_u200_xdma_201830_2.xpfm
8  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'
9  INFO: [v++ 74-74] Compiler Version string: 2020.2
10 INFO: [v++ 60-1302] Platform 'xilinx_u200_xdma_201830_2.xpfm' has been explicitly enabled for this
       release.
11 INFO: [v++ 60-629] Linking for hardware target
12 INFO: [v++ 60-423]    Target device: xilinx_u200_xdma_201830_2
13 INFO: [v++ 60-1332] Run 'run_link' status: Not started
14 INFO: [v++ 60-1443] [20:27:29] Run run_link: Step system_link: Started
15 INFO: [v++ 60-1453] Command Line: system_link --xo
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0.xo --config
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/int --temp_dir
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link
16 INFO: [v++ 60-1454] Run Directory:
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/run_link
17 INFO: [SYSTEM_LINK 60-1316] Initiating connection to rulecheck server, at Sat Nov 13 20:27:35 2021
18 INFO: [SYSTEM_LINK 82-70] Extracting xo v3 file
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0.xo
19 INFO: [SYSTEM_LINK 82-53] Creating IP database
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml
20 INFO: [SYSTEM_LINK 82-38] [20:27:36] build_xd_ip_db started:
       /data/Xilinx/Vitis/2020.2/bin/build_xd_ip_db -ip_search 0  -sds-pf
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/xilinx_u200_xdma_201830_2.hpfm -clkid 0 -ip
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/iprepo
       /mycompany_com_kernel_rtl_kernel_wizard_0_1_0,rtl_kernel_wizard_0 -o
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml
21 INFO: [SYSTEM_LINK 82-37] [20:27:50] build_xd_ip_db finished successfully
22 Time (s): cpu = 00:00:15 ; elapsed = 00:00:15 . Memory (MB): peak = 1693.402 ; gain = 0.000 ; free
       physical = 468113 ; free virtual = 483648
23 INFO: [SYSTEM_LINK 82-51] Create system connectivity graph
24 INFO: [SYSTEM_LINK 82-102] Applying explicit connections to the system connectivity graph:
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml
25 INFO: [SYSTEM_LINK 82-38] [20:27:50] cfgen started: /data/Xilinx/Vitis/2020.2/bin/cfgen  -nk
       rtl_kernel_wizard_0:1:vinc0 -slr vinc0:SLR2 -sp vinc0.m00_axi:DDR[2] -sp vinc0.m00_axi:PLRAM[0]
      -dmclkid 0 -r /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml -o
       /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
       rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml
26 INFO: [CFGEN 83-0] Kernel Specs:
27 INFO: [CFGEN 83-0]    kernel: rtl_kernel_wizard_0, num: 1  {vinc0}
28 INFO: [CFGEN 83-0] Port Specs:
```

```
INFO: [CFGEN 83-0]   kernel: vinc0, k_port: m00_axi, sptag: DDR[2]
INFO: [CFGEN 83-0]   kernel: vinc0, k_port: m00_axi, sptag: PLRAM[0]
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[2] for directive
    vinc0.m00_axi:DDR[2]
INFO: [SYSTEM_LINK 82-37] [20:28:03] cfgen finished successfully
Time (s): cpu = 00:00:12 ; elapsed = 00:00:12 . Memory (MB): peak = 1693.402 ; gain = 0.000 ; free
    physical = 468091 ; free virtual = 483626
INFO: [SYSTEM_LINK 82-52] Create top-level block diagram
INFO: [SYSTEM_LINK 82-38] [20:28:03] cf2bd started: /data/Xilinx/Vitis/2020.2/bin/cf2bd --linux
    --trace_buffer 1024 --input_file
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml --ip_db
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/_sysl/.cdb/xd_ip_db.xml --cf_name dr
    --working_dir /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/_sysl/.xsd --temp_dir
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/sys_link --output_dir
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int --target_bd pfm_dynamic.bd
INFO: [CF2BD 82-31] Launching cf2xd: cf2xd -linux -trace-buffer 1024 -i
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/cfgraph/cfgen_cfgraph.xml -r
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/sys_link/_sysl/.xsd
INFO: [CF2BD 82-28] cf_xsd finished successfully
INFO: [SYSTEM_LINK 82-37] [20:28:10] cf2bd finished successfully
Time (s): cpu = 00:00:06 ; elapsed = 00:00:07 . Memory (MB): peak = 1693.402 ; gain = 0.000 ; free
    physical = 468080 ; free virtual = 483620
INFO: [v++ 60-1441] [20:28:10] Run run_link: Step system_link: Completed
Time (s): cpu = 00:00:41 ; elapsed = 00:00:41 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free
    physical = 468113 ; free virtual = 483648
INFO: [v++ 60-1443] [20:28:10] Run run_link: Step cf2sw: Started
INFO: [v++ 60-1453] Command Line: cf2sw -sdsl
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/sdsl.dat -rtd
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/cf2sw.rtd -nofilter
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/cf2sw_full.rtd -xclbin
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/xclbin_orig.xml -o
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/xclbin_orig.1.xml
INFO: [v++ 60-1454] Run Directory:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/run_link
INFO: [v++ 60-1441] [20:28:18] Run run_link: Step cf2sw: Completed
Time (s): cpu = 00:00:07 ; elapsed = 00:00:08 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free
    physical = 468140 ; free virtual = 483676
INFO: [v++ 60-1443] [20:28:18] Run run_link: Step rtd2_system_diagram: Started
INFO: [v++ 60-1453] Command Line: rtd2SystemDiagram
INFO: [v++ 60-1454] Run Directory:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/run_link
INFO: [v++ 60-1441] [20:28:22] Run run_link: Step rtd2_system_diagram: Completed
Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:04 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free
    physical = 467615 ; free virtual = 483150
INFO: [v++ 60-1443] [20:28:22] 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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/.ipcache --output_dir
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int --log_dir
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/logs/link --report_dir
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/reports/link --config
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/vplConfig.ini -k
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/kernel_info.dat --webtalk_flag Vitis --temp_dir
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
```

```
      rtl_kernel_wizard_0_ex/exports/_x/link --no-info --iprepo
      /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
      rtl_kernel_wizard_0_ex/exports/_x/link/int/xo/ip_repo/mycompany_com_kernel_rtl_kernel_wizard_0_1_0
      --messageDb /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
      rtl_kernel_wizard_0_ex/exports/_x/link/run_link/vpl.pb
      /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
      rtl_kernel_wizard_0_ex/exports/_x/link/int/dr.bd.tcl
58 INFO: [v++ 60-1454] Run Directory:
      /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
      rtl_kernel_wizard_0_ex/exports/_x/link/run_link
59
60 ****** vpl v2020.2 (64-bit)
61   **** SW Build (by xbuild) on 2020-11-18-05:13:29
62     ** Copyright 1986-2020 Xilinx, Inc. All Rights Reserved.
63
64 INFO: [VPL 60-839] Read in kernel information from file
      '/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
      rtl_kernel_wizard_0_ex/exports/_x/link/int/kernel_info.dat'.
65 INFO: [VPL 74-74] Compiler Version string: 2020.2
66 INFO: [VPL 60-423]   Target device: xilinx_u200_xdma_201830_2
67 INFO: [VPL 60-1032] Extracting hardware platform to
      /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
      rtl_kernel_wizard_0_ex/exports/_x/link/vivado/vpl/.local/hw_platform
68 WARNING: /data/Xilinx/Vitis/2020.2/tps/lnx64/jre9.0.4 does not exist.
69 [20:31:37] Run vpl: Step create_project: Started
70 Creating Vivado project.
71 [20:31:50] Run vpl: Step create_project: Completed
72 [20:31:50] Run vpl: Step create_bd: Started
73 [20:33:26] Run vpl: Step create_bd: RUNNING...
74 [20:35:03] Run vpl: Step create_bd: RUNNING...
75 [20:36:32] Run vpl: Step create_bd: RUNNING...
76 [20:37:05] Run vpl: Step create_bd: Completed
77 [20:37:05] Run vpl: Step update_bd: Started
78 [20:37:07] Run vpl: Step update_bd: Completed
79 [20:37:07] Run vpl: Step generate_target: Started
80 [20:38:33] Run vpl: Step generate_target: RUNNING...
81 [20:39:59] Run vpl: Step generate_target: RUNNING...
82 [20:41:25] Run vpl: Step generate_target: RUNNING...
83 [20:42:53] Run vpl: Step generate_target: RUNNING...
84 [20:43:37] Run vpl: Step generate_target: Completed
85 [20:43:37] Run vpl: Step config_hw_runs: Started
86 [20:43:52] Run vpl: Step config_hw_runs: Completed
87 [20:43:52] Run vpl: Step synth: Started
88 [20:45:07] Block-level synthesis in progress, 0 of 13 jobs complete, 8 jobs running.
89 [20:45:41] Block-level synthesis in progress, 0 of 13 jobs complete, 8 jobs running.
90 [20:46:18] Block-level synthesis in progress, 0 of 13 jobs complete, 8 jobs running.
91 [20:46:53] Block-level synthesis in progress, 0 of 13 jobs complete, 8 jobs running.
92 [20:47:27] Block-level synthesis in progress, 0 of 13 jobs complete, 8 jobs running.
93 [20:48:02] Block-level synthesis in progress, 0 of 13 jobs complete, 8 jobs running.
94 [20:48:37] Block-level synthesis in progress, 5 of 13 jobs complete, 3 jobs running.
95 [20:49:14] Block-level synthesis in progress, 5 of 13 jobs complete, 5 jobs running.
96 [20:49:49] Block-level synthesis in progress, 5 of 13 jobs complete, 7 jobs running.
97 [20:50:25] Block-level synthesis in progress, 6 of 13 jobs complete, 6 jobs running.
98 [20:51:01] Block-level synthesis in progress, 6 of 13 jobs complete, 6 jobs running.
99 [20:51:36] Block-level synthesis in progress, 6 of 13 jobs complete, 6 jobs running.
100 [20:52:11] Block-level synthesis in progress, 6 of 13 jobs complete, 6 jobs running.
101 [20:52:49] Block-level synthesis in progress, 7 of 13 jobs complete, 5 jobs running.
102 [20:53:23] Block-level synthesis in progress, 9 of 13 jobs complete, 3 jobs running.
103 [20:53:58] Block-level synthesis in progress, 9 of 13 jobs complete, 3 jobs running.
104 [20:54:33] Block-level synthesis in progress, 9 of 13 jobs complete, 3 jobs running.
105 [20:55:09] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
106 [20:55:44] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
107 [20:56:20] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
108 [20:56:55] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
109 [20:57:31] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
110 [20:58:06] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
111 [20:58:43] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
112 [20:59:18] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
113 [20:59:54] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
114 [21:00:29] Block-level synthesis in progress, 10 of 13 jobs complete, 2 jobs running.
115 [21:01:05] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
116 [21:01:40] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
117 [21:02:15] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
118 [21:02:49] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
119 [21:03:25] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
120 [21:04:00] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
121 [21:04:37] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
122 [21:05:12] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
123 [21:05:48] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
```

```
[21:06:22] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:06:59] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:07:34] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:08:12] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:08:47] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:09:22] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:09:56] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:10:32] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:11:08] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:11:43] Block-level synthesis in progress, 11 of 13 jobs complete, 1 job running.
[21:12:17] Block-level synthesis in progress, 12 of 13 jobs complete, 0 jobs running.
[21:12:53] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:13:28] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:14:02] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:14:37] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:15:11] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:15:46] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:16:21] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:16:55] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:17:30] Block-level synthesis in progress, 12 of 13 jobs complete, 1 job running.
[21:18:05] Block-level synthesis in progress, 13 of 13 jobs complete, 0 jobs running.
[21:18:40] Block-level synthesis in progress, 13 of 13 jobs complete, 0 jobs running.
[21:19:15] Top-level synthesis in progress.
[21:19:50] Top-level synthesis in progress.
[21:20:25] Top-level synthesis in progress.
[21:21:01] Top-level synthesis in progress.
[21:21:36] Top-level synthesis in progress.
[21:22:12] Top-level synthesis in progress.
[21:22:47] Top-level synthesis in progress.
[21:23:23] Top-level synthesis in progress.
[21:23:59] Run vpl: Step synth: Completed
[21:23:59] Run vpl: Step impl: Started
[21:50:14] Finished 2nd of 6 tasks (FPGA linking synthesized kernels to platform). Elapsed time: 01h
    21m 46s

[21:50:14] Starting logic optimization..
[21:53:09] Phase 1 Generate And Synthesize MIG Cores
[22:10:02] Phase 2 Generate And Synthesize Debug Cores
[22:21:59] Phase 3 Retarget
[22:22:35] Phase 4 Constant propagation
[22:23:51] Phase 5 Sweep
[22:25:38] Phase 6 BUFG optimization
[22:26:15] Phase 7 Shift Register Optimization
[22:26:52] Phase 8 Post Processing Netlist
[22:33:28] Finished 3rd of 6 tasks (FPGA logic optimization). Elapsed time: 00h 43m 13s

[22:33:28] Starting logic placement..
[22:35:13] Phase 1 Placer Initialization
[22:35:13] Phase 1.1 Placer Initialization Netlist Sorting
[22:40:04] Phase 1.2 IO Placement/ Clock Placement/ Build Placer Device
[22:43:35] Phase 1.3 Build Placer Netlist Model
[22:48:52] Phase 1.4 Constrain Clocks/Macros
[22:49:28] Phase 2 Global Placement
[22:49:28] Phase 2.1 Floorplanning
[22:51:14] Phase 2.1.1 Partition Driven Placement
[22:51:14] Phase 2.1.1.1 PBP: Partition Driven Placement
[22:52:26] Phase 2.1.1.2 PBP: Clock Region Placement
[22:54:11] Phase 2.1.1.3 PBP: Compute Congestion
[22:54:11] Phase 2.1.1.4 PBP: UpdateTiming
[22:55:21] Phase 2.1.1.5 PBP: Add part constraints
[22:55:57] Phase 2.2 Update Timing before SLR Path Opt
[22:55:57] Phase 2.3 Global Placement Core
[23:12:35] Phase 2.3.1 Physical Synthesis In Placer
[23:17:54] Phase 3 Detail Placement
[23:17:54] Phase 3.1 Commit Multi Column Macros
[23:18:29] Phase 3.2 Commit Most Macros & LUTRAMs
[23:20:52] Phase 3.3 Small Shape DP
[23:20:52] Phase 3.3.1 Small Shape Clustering
[23:22:03] Phase 3.3.2 Flow Legalize Slice Clusters
[23:22:03] Phase 3.3.3 Slice Area Swap
[23:25:03] Phase 3.4 Place Remaining
[23:25:03] Phase 3.5 Re-assign LUT pins
[23:25:38] Phase 3.6 Pipeline Register Optimization
[23:25:38] Phase 3.7 Fast Optimization
[23:27:59] Phase 4 Post Placement Optimization and Clean-Up
[23:27:59] Phase 4.1 Post Commit Optimization
[23:32:06] Phase 4.1.1 Post Placement Optimization
[23:32:42] Phase 4.1.1.1 BUFG Insertion
[23:32:42] Phase 1 Physical Synthesis Initialization
```

```
[23:33:52] Phase 4.1.1.2 BUFG Replication
[23:35:03] Phase 4.1.1.3 Replication
[23:38:01] Phase 4.2 Post Placement Cleanup
[23:38:36] Phase 4.3 Placer Reporting
[23:38:36] Phase 4.3.1 Print Estimated Congestion
[23:39:11] Phase 4.4 Final Placement Cleanup
[00:07:01] Finished 4th of 6 tasks (FPGA logic placement). Elapsed time: 01h 33m 33s

[00:07:01] Starting logic routing..
[00:09:22] Phase 1 Build RT Design
[00:14:04] Phase 2 Router Initialization
[00:14:04] Phase 2.1 Fix Topology Constraints
[00:14:39] Phase 2.2 Pre Route Cleanup
[00:14:39] Phase 2.3 Global Clock Net Routing
[00:15:50] Phase 2.4 Update Timing
[00:21:43] Phase 2.5 Update Timing for Bus Skew
[00:21:43] Phase 2.5.1 Update Timing
[00:24:40] Phase 3 Initial Routing
[00:24:40] Phase 3.1 Global Routing
[00:27:01] Phase 4 Rip-up And Reroute
[00:27:01] Phase 4.1 Global Iteration 0
[00:38:14] Phase 4.2 Global Iteration 1
[00:44:11] Phase 4.3 Global Iteration 2
[00:46:34] Phase 5 Delay and Skew Optimization
[00:46:34] Phase 5.1 Delay CleanUp
[00:46:34] Phase 5.1.1 Update Timing
[00:50:07] Phase 5.2 Clock Skew Optimization
[00:50:07] Phase 6 Post Hold Fix
[00:50:07] Phase 6.1 Hold Fix Iter
[00:50:07] Phase 6.1.1 Update Timing
[00:53:05] Phase 7 Route finalize
[00:53:05] Phase 8 Verifying routed nets
[00:53:40] Phase 9 Depositing Routes
[00:55:26] Phase 10 Route finalize
[00:55:26] Phase 11 Post Router Timing
[00:58:21] Finished 5th of 6 tasks (FPGA routing). Elapsed time: 00h 51m 19s

[00:58:21] Starting bitstream generation..
[01:55:05] Creating bitmap...
[02:20:29] Writing bitstream ./pfm_top_i_dynamic_region_my_rm_partial.bit...
[02:20:29] Finished 6th of 6 tasks (FPGA bitstream generation). Elapsed time: 01h 22m 08s
[02:21:52] Run vpl: Step impl: Completed
[02:21:56] Run vpl: FINISHED. Run Status: impl Complete!
INFO: [v++ 60-1441] [02:22:18] Run run_link: Step vpl: Completed
Time (s): cpu = 00:08:48 ; elapsed = 05:53:57 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free
    physical = 461579 ; free virtual = 486671
INFO: [v++ 60-1443] [02:22:18] Run run_link: Step rtdgen: Started
INFO: [v++ 60-1453] Command Line: rtdgen
INFO: [v++ 60-1454] Run Directory:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/address_map.xml -sdsl
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/sdsl.dat -xclbin
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/xclbin_orig.xml -rtd
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc.rtd -o
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc.xml
INFO: [v++ 60-1652] Cf2sw returned exit code: 0
INFO: [v++ 60-2311] HPISystemDiagram::writeSystemDiagramAfterRunningVivado, rtdInputFilePath:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc.rtd
INFO: [v++ 60-2312] HPISystemDiagram::writeSystemDiagramAfterRunningVivado,
    systemDiagramOutputFilePath:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/int/systemDiagramModelSlrBaseAddress.json
INFO: [v++ 60-1618] Launching
INFO: [v++ 60-1441] [02:22:26] Run run_link: Step rtdgen: Completed
```

```
Time (s): cpu = 00:00:07 ; elapsed = 00:00:07 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 461569 ; free virtual = 486660
INFO: [v++ 60-1443] [02:22:26] Run run_link: Step xclbinutil: Started
INFO: [v++ 60-1453] Command Line: xclbinutil --add-section DEBUG_IP_LAYOUT:JSON:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/debug_ip_layout.rtd --add-section BITSTREAM:RAW:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/partial.bit --force --target hw --key-value SYS:dfx_enable:true --add-section :JSON:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc.rtd --append-section :JSON:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/appendSection.rtd --add-section CLOCK_FREQ_TOPOLOGY:JSON:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc_xml.rtd --add-section BUILD_METADATA:JSON:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc_build.rtd --add-section EMBEDDED_METADATA:RAW:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc.xml --add-section SYSTEM_METADATA:RAW:/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/systemDiagramModelSlrBaseAddress.json --output /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0_vinc.xclbin
INFO: [v++ 60-1454] Run Directory: /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/debug_ip_layout.rtd'

Section: 'BITSTREAM'(0) was successfully added.
Size   : 42135954 bytes
Format : RAW
File   : '/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc_xml.rtd'

Section: 'BUILD_METADATA'(14) was successfully added.
Size   : 2966 bytes
Format : JSON
File   : '/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc_build.rtd'

Section: 'EMBEDDED_METADATA'(2) was successfully added.
Size   : 2779 bytes
Format : RAW
File   : '/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/rtl_kernel_wizard_0_vinc.xml'

Section: 'SYSTEM_METADATA'(22) was successfully added.
Size   : 5666 bytes
Format : RAW
File   : '/iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/rtl_kernel_wizard_0_ex/exports/_x/link/int/systemDiagramModelSlrBaseAddress.json'

```

```
Section: 'IP_LAYOUT'(8) was successfully appended to.
Format : JSON
File   : 'ip_layout'
Successfully wrote (42158078 bytes) to the output file:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0_vinc.xclbin
Leaving xclbinutil.
INFO: [v++ 60-1441] [02:22:28] Run run_link: Step xclbinutil: Completed
Time (s): cpu = 00:00:00.29 ; elapsed = 00:00:02 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free
    physical = 461529 ; free virtual = 486661
INFO: [v++ 60-1443] [02:22:28] Run run_link: Step xclbinutilinfo: Started
INFO: [v++ 60-1453] Command Line: xclbinutil --quiet --force --info
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0_vinc.xclbin.info --input
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0_vinc.xclbin
INFO: [v++ 60-1454] Run Directory:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/run_link
INFO: [v++ 60-1441] [02:22:29] Run run_link: Step xclbinutilinfo: Completed
Time (s): cpu = 00:00:01 ; elapsed = 00:00:02 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free
    physical = 461524 ; free virtual = 486655
INFO: [v++ 60-1443] [02:22:29] Run run_link: Step generate_sc_driver: Started
INFO: [v++ 60-1453] Command Line:
INFO: [v++ 60-1454] Run Directory:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/link/run_link
INFO: [v++ 60-1441] [02:22:29] Run run_link: Step generate_sc_driver: Completed
Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:00.04 . Memory (MB): peak = 1585.129 ; gain = 0.000 ;
    free physical = 461523 ; free virtual = 486655
INFO: [v++ 60-244] Generating system estimate report...
INFO: [v++ 60-1092] Generated system estimate report:
    /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/_x/reports/link/system_estimate_rtl_kernel_wizard_0_vinc.xtxt
INFO: [v++ 60-586] Created /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
    rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0_vinc.ltx
INFO: [v++ 60-586] Created rtl_kernel_wizard_0_vinc.xclbin
INFO: [v++ 60-1307] Run completed. Additional information can be found in:
    Guidance: /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
        rtl_kernel_wizard_0_ex/exports/_x/reports/link/v++_link_rtl_kernel_wizard_0_vinc_guidance.html
    Timing Report: /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
        rtl_kernel_wizard_0_ex/exports/_x/reports/link/imp/ impl_1
        _xilinx_u200_xdma_201830_2_bb_locked_timing_summary_routed.rpt
    Vivado Log: /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
        rtl_kernel_wizard_0_ex/exports/_x/logs/link/vivado.log
    Steps Log File: /iu_home/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
        rtl_kernel_wizard_0_ex/exports/_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/iu7041/workspace/iu7_52b_11/Alveo_lab1_kernels/vivado_rtl_kernel/
        rtl_kernel_wizard_0_ex/exports/rtl_kernel_wizard_0_vinc.xclbin.link_summary
INFO: [v++ 60-791] Total elapsed time: 5h 55m 47s
INFO: [v++ 60-1653] Closing dispatch client.
```

ПРИЛОЖЕНИЕ В

СОДЕРЖИМОЕ XCLBIN.INFO-ФАЙЛА

Листинг В.1 — 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):           2b3110c6-decb-4dbb-b638-a8e10a7fb0e4
   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:     Yes

   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 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:             bank2 (MEM_DDR4)
==============================================================================
Generated By
------------
    Command:        v++
    Version:        2020.2 - 2020-11-18-05:13:29 (SW BUILD: 0)
    Command Line:   v++ --config rtl_kernel_wizard_0.cfg --connectivity.nk rtl_kernel_wizard_0:1:vinc0
        --connectivity.slr vinc0:SLR2 --connectivity.sp vinc0.m00_axi:DDR[2] --connectivity.sp
        vinc0.m00_axi:PLRAM[0] --input_files rtl_kernel_wizard_0.xo --link --optimize 0 --output
        rtl_kernel_wizard_0_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 rtl_kernel_wizard_0.cfg
                    --connectivity.nk rtl_kernel_wizard_0:1:vinc0
                    --connectivity.slr vinc0:SLR2
                    --connectivity.sp vinc0.m00_axi:DDR[2]
                    --connectivity.sp vinc0.m00_axi:PLRAM[0]
```

```
                    --input_files rtl_kernel_wizard_0.xo
                    --link
                    --optimize 0
                    --output rtl_kernel_wizard_0_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>
================================================================================
```