



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»

НАПРАВЛЕНИЕ ПОДГОТОВКИ «09.03.04 Программная инженерия»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4

Название: Разработка ускорителей вычислений на платформе Xilinx Alveo

Дисциплина: Архитектура ЭВМ

Студент ИУ7-54Б С. Д. Параскун

Группа

Подпись, дата

И. О. Фамилия

Преподаватель А. Ю. Попов

Подпись, дата

И. О. Фамилия

Москва, 2021 г.

Цели работы

Основной целью работы является изучение архитектуры гетерогенных вычислительных систем и технологии разработки ускорителей вычислений на базе ПЛИС фирмы Xilinx. В ходе лабораторной работы будут изучены основные сведения о платформе Xilinx Alveo U200, разработано RTL (Register Transfer Language, язык регистровых передач) описание ускорителя вычислений по индивидуальному варианту, выполнена генерацию ядра ускорителя, выполнены синтез и сборка бинарного модуля ускорителя, разработаны и отлажены тестирующее программное обеспечение на серверной хост-платформе, проведены тесты работы ускорителя вычислений.

В данной работе будет выполнен 12 вариант.

1. Функциональная схема

На рис. 1.1 представлена функциональная схема разрабатываемой аппаратной системы. Дальнейшая работа будет выполняться по ней.

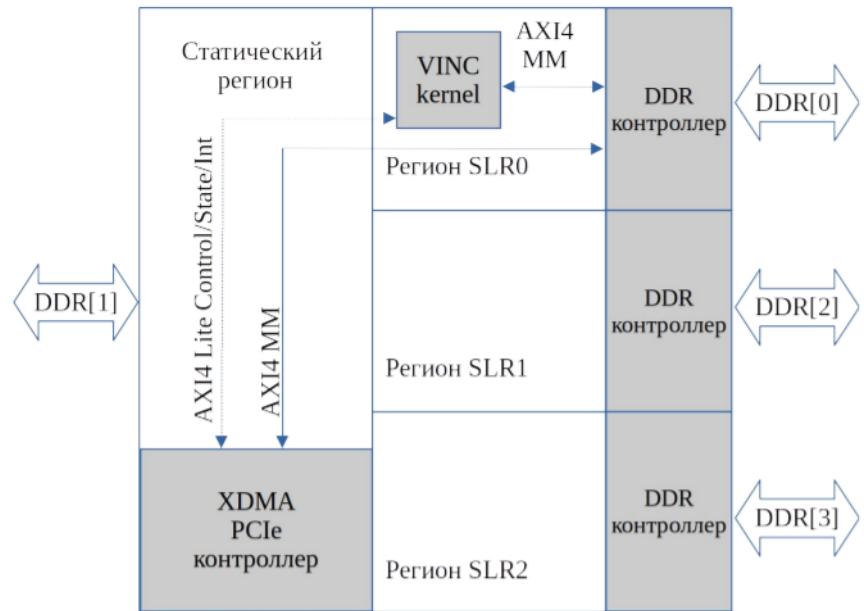


Рисунок 1.1 – Функциональная схема разрабатываемой аппаратной системы

2. Изучение работы шины AXI

На рис. 2.1 - 2.3 представлены транзакции чтения данных вектора на шине AXI4 MM из DDR памяти, записи результата инкремента данных на шине и инкремент данных в модуле `rtl_kernel_wizard_0_example_adder.v` исходной симуляции.

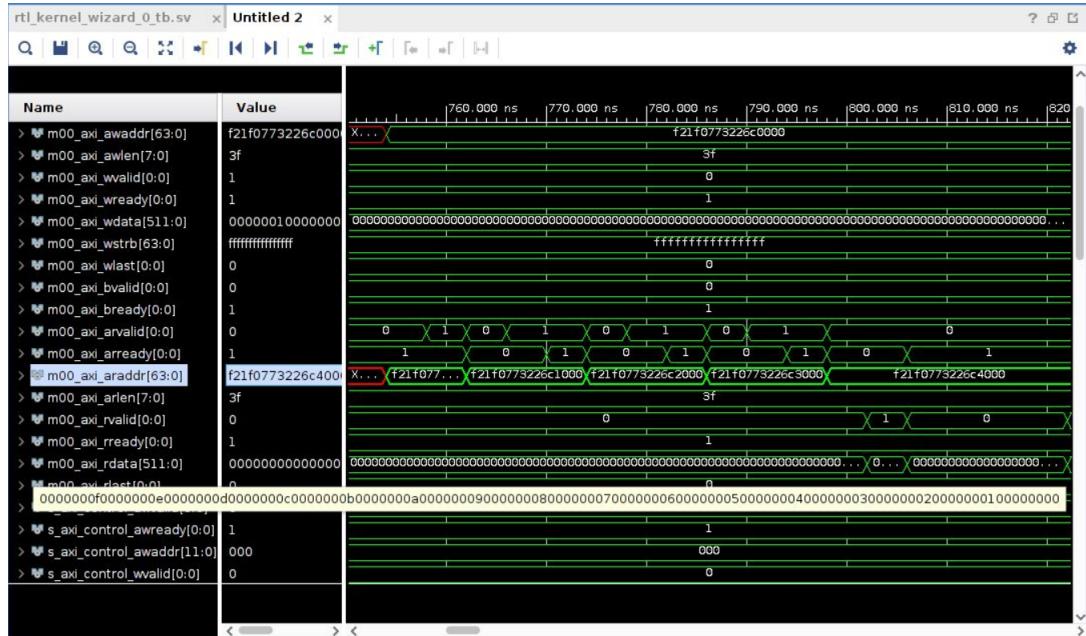


Рисунок 2.1 – Транзакции чтения данных вектора

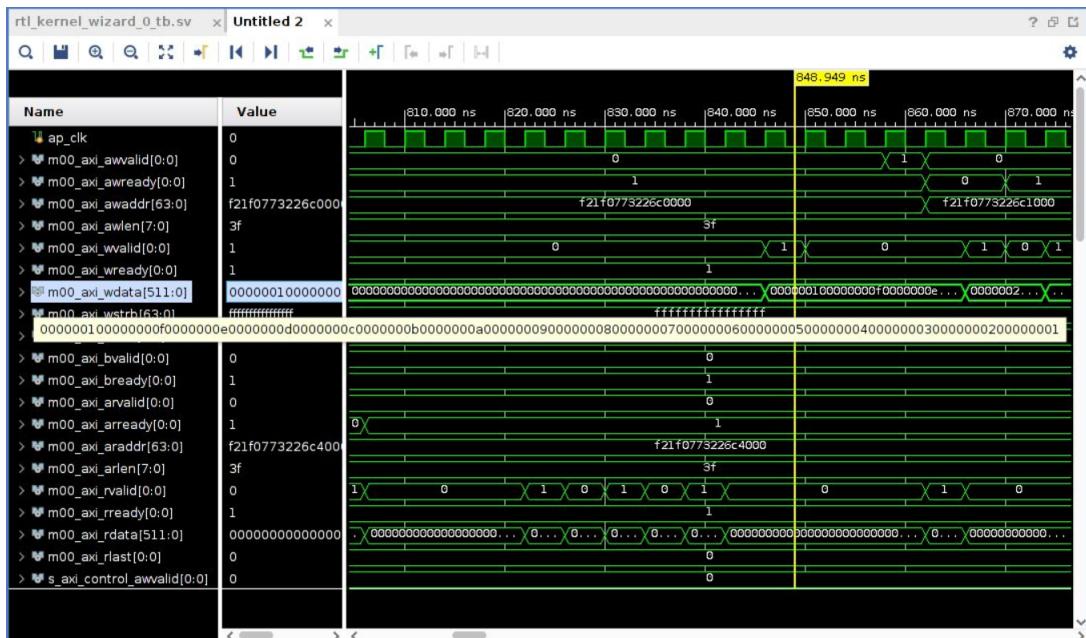


Рисунок 2.2 – Транзакции записи результата инкремента данных

Исходя из данных диаграмм, можно заметить, что считанные данные были инкрементированы корректно.

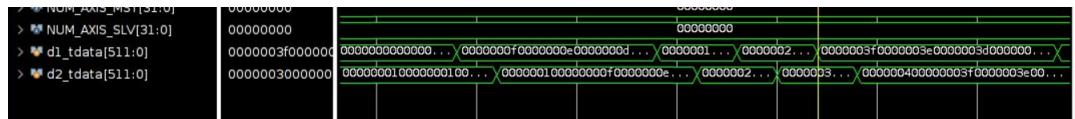


Рисунок 2.3 – Инкремент данных

Теперь изменим rtl_kernel_wizard_0_example_adder.v в соответствии с индивидуальным заданием (вариант 12), чтобы ускоритель выполнял требуемую функцию:

$$R[i] = (A[i] \& 0xf0f0f0f0f0f0f0f0) + 10 \quad (2.1)$$

Код функции указан в листинге ниже.

Листинг 2.1 – Код инкремента данных

```

1 // Adder function
2 always @ (posedge s_axis_aclk) begin
3     for (i = 0; i < LP_NUM_LOOPS; i = i + 1) begin
4         d2_tdata[i*C_ADDER_BIT_WIDTH+:C_ADDER_BIT_WIDTH] <=
5             d1_tdata[C_ADDER_BIT_WIDTH*i+:C_ADDER_BIT_WIDTH] &
6             'hf0f0f0f0f0 + 10;
7     end
8 end

```

На рис. 2.5 - 2.7 представлены представлены транзакции чтения данных вектора, записи результата инкремента данных и инкремент данных в rtl_kernel_wizard_0_example_adder.v после изменения данного файла.

Также пришлось исправить модуль проверки в rtl_kernel_wizard_0_tb.sv, чтобы не возникало ошибок при проверке (рис. 2.4).

```

rtl_kernel_wizard_0_tb.v | rtl_kernel_wizard_0_tb_behav.wcfg | ? ⓘ
/uu_home/uu7102/lab_04/lab_04_kernels/vivado_rtl/kernel/rtl_kernel_wizard_0_ex/imports/rtl_kernel_wizard_0_tb.v

Q | F | ← | → | X | D | B | // | E | I | ? |
488 : integer error_counter;
489 : error_counter = 0;
490 :
491 //////////////////////////////////////////////////////////////////
492 begin
493 : for (longint unsigned slot = 0; slot < LP_MAX_LENGTH; slot++) begin
494 :   ret_rd_value = m00_axi.mem_model.backdoor_memory_read_4byte(axi00_ptr0_ptr + (slot * 4));
495 :   if (slot < LP_MAX_TRANSFER_LENGTH) begin
496 :     if (ret_rd_value != (slot & 'hfofofofo + 10)) begin
497 :       $error("Memory Mismatch: m00_axi : @0x% : Expected 0x% -> Got 0x% ", axi00_ptr0_ptr + (slot * 4), slot & 'hfofofofo + 10, ret_rd_value);
498 :       error_found |= 1;
499 :       error_counter++;
500 :     end
501 :   end else begin
502 :     if (ret_rd_value != slot) begin
503 :       $error("Memory Mismatch: m00_axi : @0x% : Expected 0x% -> Got 0x% ", axi00_ptr0_ptr + (slot * 4), slot, ret_rd_value);
504 :       error_found |= 1;
505 :       error_counter++;
506 :     end
507 :   end
508 :   if (error_counter > 5) begin
509 :     $display("Too many errors found. Exiting check of m00_axi.");
510 :     slot = LP_MAX_LENGTH;
511 :   end
512 : end
513 : error_counter = 0;
514 :
515 : return(error_found);
516 : endfunction

```

Рисунок 2.4 – Код проверки в модуле rtl_kernel_wizard_0_tb.v

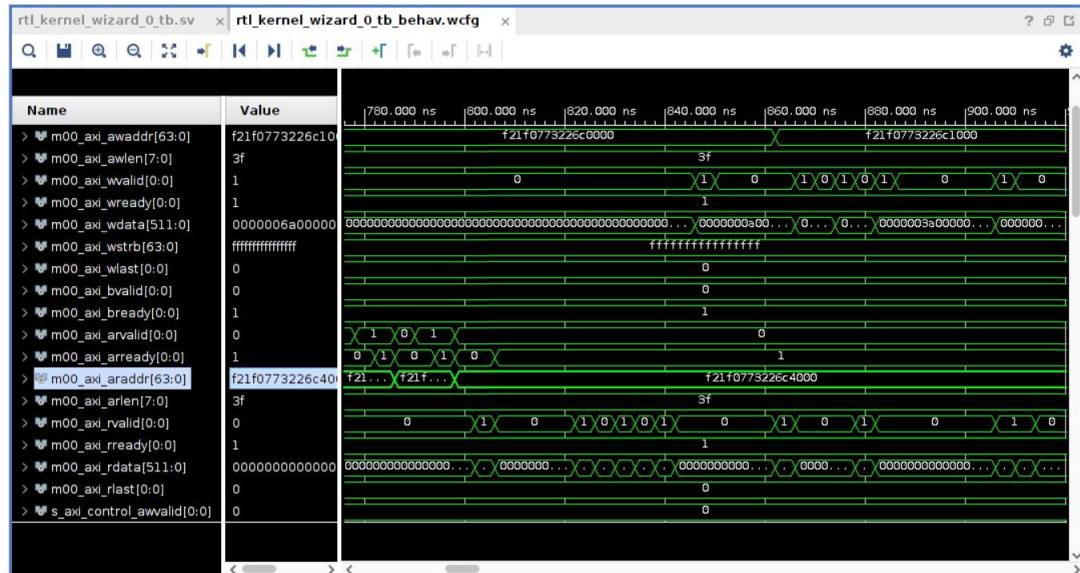


Рисунок 2.5 – Транзакции чтения данных вектора

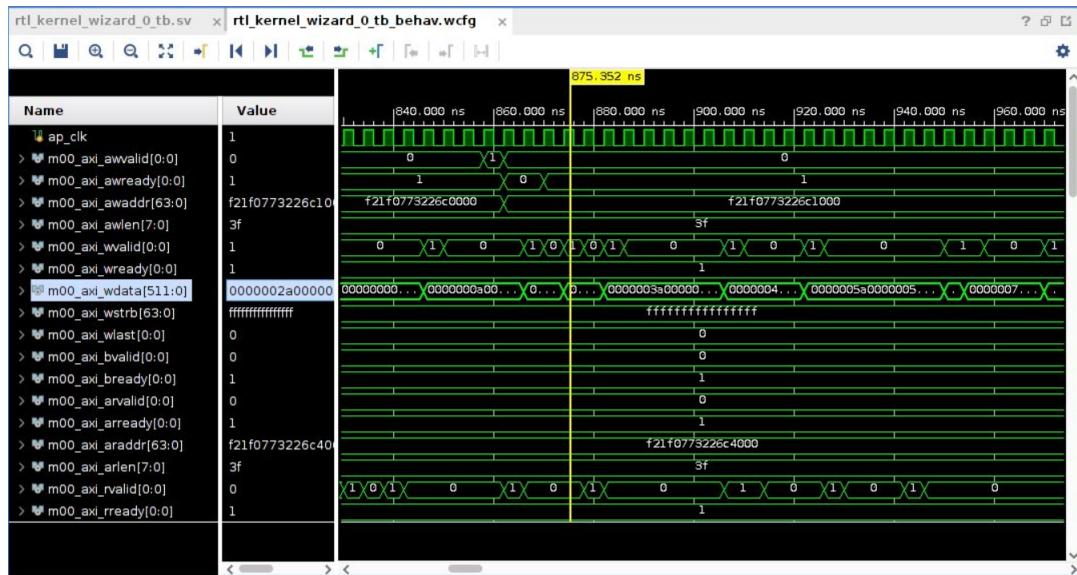


Рисунок 2.6 – Транзакции записи результата инкремента данных



Рисунок 2.7 – Инкремент данных

3. Сборка проекта

Был создан конфигурационный файл, листинг которого представлен ниже.

Листинг 3.1 – Листинг конфигурационного файла rtl_kernel_wizard_0.cfg

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

В приложении А представлен листинг файла v++_vinc.log, в приложении Б - vinc.xclbin.info.

4. Тестирование

Для тестирования используется программа, код которой представлен в файле host_example.cpp (полный листинг в приложении В). Ниже представлена измененная часть, в соответствии с вариантом.

Листинг 4.1 – Модифицированный модуль host_example.cpp

```
1 for (cl_uint i = 0; i < number_of_words; i++) {
2     if ((h_data[i] & 0xf0f0f0f0f0 + 10) != h_axi00_ptr0_output[i]) {
3         printf("ERROR in rtl_kernel_wizard_0::m00_axi - array index
4             %d (host addr 0x%03x) - input=%d (0x%x), output=%d (0x%x)\n",
5             i, i*4, h_data[i], h_data[i], h_axi00_ptr0_output[i],
6             h_axi00_ptr0_output[i]);
7         check_status = 1;
8     }
}
```

На рис. 4.1 и 4.2 представлено тестирование проекта.

```
iu7102@dl580:~/lab_04$ xgdb --args /iu_home/iu7102/lab_04/lab_04/Emulation-SW/lab_04 /iu_home/iu7102/lab_04/vin
c.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 /iu_home/iu7102/lab_04/lab_04/Emulation-SW/lab_04...
```

Рисунок 4.1 – Тестирование

```
(gdb) run
Starting program: /iu_home/iu7102/lab_04/lab_04/Emulation-SW/lab_04 /iu_home/iu7102/lab_04/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 27662)]
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/iu7102/lab_04/vinc.xclbin
[New Thread 0x7ffff532e700 (LWP 29200)]
[New Thread 0x7ffff4b2d700 (LWP 29202)]
[New Thread 0x7ffffeff700 (LWP 29203)]
[New Thread 0x7ffffeff7fe700 (LWP 29204)]
[New Thread 0x7ffffefffd700 (LWP 29205)]
[New Thread 0x7ffffee7fc700 (LWP 29210)]
INFO: Test completed successfully.
[Thread 0x7ffff532e700 (LWP 29200) exited]
[Thread 0x7ffff5b2f700 (LWP 27662) exited]
[Thread 0x7ffffeff7fe700 (LWP 29204) exited]
[Thread 0x7ffffee7fc700 (LWP 29210) exited]
[Thread 0x7ffffeff700 (LWP 29203) exited]
[Thread 0x7ffff4b2d700 (LWP 29202) exited]
[Thread 0x7ffffefffd700 (LWP 29205) exited]
[Inferior 1 (process 27618) exited normally]
```

Рисунок 4.2 – Тестирование (часть 2)

Все тесты пройдены успешно, а значит программа на ускорителе работает корректно.

Вывод

В ходе лабораторной работы были изучены основные сведения о платформе Xilinx Alveo U200, разработано RTL описание ускорителя по варианту, выполнена генерацию ядра ускорителя, выполнены синтез и сборка бинарного модуля ускорителя, разработаны и отлажены тестирующее программное обеспечение на серверной хост-платформе, проведены тесты работы ускорителя вычислений. Поставленные цели были достигнуты.

Приложение А

Листинг 4.2 – Листинг файла v++_vinc.log

```
1 INFO: [v++ 60-1306] Additional information associated with this v++ link can be found at:  
2     Reports: /iu_home/iu7102/lab_04/_x/reports/link  
3     Log files: /iu_home/iu7102/lab_04/_x/logs/link  
4 INFO: [v++ 60-1548] Creating build summary session with primary output /iu_home/iu7102/lab_04/vinc.xclbin.link_summary  
6 00:25:36 2022  
5 INFO: [v++ 60-1316] Initiating connection to rulecheck server, at Thu Jan 6 00:25:36 2022  
6 INFO: [v++ 60-1315] Creating rulecheck session with output '/iu_home/iu7102/lab_04/_x/reports/link/v++_link_vinc_guide'  
6 00:25:55 2022  
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.xsa'  
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] [00:26:56] Run run_link: Step system_link: Started  
15 INFO: [v++ 60-1453] Command Line: system_link --xo /iu_home/iu7102/lab_04/lab_04_kernels/vivado_rtl_kernel/rtl_kernel  
16 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link  
17 INFO: [SYSTEM_LINK 60-1316] Initiating connection to rulecheck server, at Thu Jan 6 00:27:11 2022  
18 INFO: [SYSTEM_LINK 82-70] Extracting xo v3 file /iu_home/iu7102/lab_04/lab_04_kernels/vivado_rtl_kernel/rtl_kernel_wiz  
19 INFO: [SYSTEM_LINK 82-53] Creating IP database /iu_home/iu7102/lab_04/_x/link/sys_link/_sys1/.cdb/xd_ip_db.xml  
20 INFO: [SYSTEM_LINK 82-38] [00:27:14] build_xd_ip_db started: /data/Xilinx/Vitis/2020.2/bin/build_xd_ip_db -p_search  
-sds-pf /iu_home/iu7102/lab_04/_x/link/sys_link/xilinx_u200_xdma_201830_2.hpfm -clkid 0 -ip /iu_home/iu7102/lab_04/_x/  
21 INFO: [SYSTEM_LINK 82-37] [00:27:49] build_xd_ip_db finished successfully  
22 Time (s): cpu = 00:00:36 ; elapsed = 00:00:36 . Memory (MB): peak = 1557.898 ; gain = 0.000 ; free physical = 46268  
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/iu7102/lab_04/_x/  
25 INFO: [SYSTEM_LINK 82-38] [00:27:50] cfgen started: /data/Xilinx/Vitis/2020.2/bin/cfgen -nk rtl_kernel_wizard_0:1:v  
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:  
29 INFO: [CFGEN 83-0] kernel: vinc0, k_port: m00_axi, sptag: DDR[3]  
30 INFO: [CFGEN 83-0] SLR Specs:  
31 INFO: [CFGEN 83-0] instance: vinc0, SLR: SLR2  
32 INFO: [CFGEN 83-2228] Creating mapping for argument vinc0.axi00_ptr0 to DDR[3] for directive vinc0.m00_axi:DDR[3]  
33 INFO: [SYSTEM_LINK 82-37] [00:28:16] cfgen finished successfully  
34 Time (s): cpu = 00:00:26 ; elapsed = 00:00:27 . Memory (MB): peak = 1557.898 ; gain = 0.000 ; free physical = 45902  
35 INFO: [SYSTEM_LINK 82-52] Create top-level block diagram  
36 INFO: [SYSTEM_LINK 82-38] [00:28:16] cf2bd started: /data/Xilinx/Vitis/2020.2/bin(cf2bd --linux --trace_buffer 1024  
37 INFO: [CF2BD 82-31] Launching cf2xd: cf2xd-linux-trace-buffer 1024 -i /iu_home/iu7102/lab_04/_x/link/sys_link/cfgra  
38 INFO: [CF2BD 82-28] cf2xd finished successfully  
39 INFO: [CF2BD 82-31] Launching cf_xsd: cf_xsd-disable-address-gen -bd pfm_dynamic.bd -dn dr -dp /iu_home/iu7102/lab_04/_x/  
40 INFO: [CF2BD 82-28] cf_xsd finished successfully  
41 INFO: [SYSTEM_LINK 82-37] [00:28:32] cf2bd finished successfully  
42 Time (s): cpu = 00:00:13 ; elapsed = 00:00:16 . Memory (MB): peak = 1557.898 ; gain = 0.000 ; free physical = 45998  
43 INFO: [v++ 60-1441] [00:28:33] Run run_link: Step system_link: Completed  
44 Time (s): cpu = 00:01:31 ; elapsed = 00:01:36 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 46083  
45 INFO: [v++ 60-1443] [00:28:33] Run run_link: Step cf2sw: Started  
46 INFO: [v++ 60-1453] Command Line: cf2sw-sdsl /iu_home/iu7102/lab_04/_x/link/int/sdsl.dat -rtd /iu_home/iu7102/lab_04/_x/  
47 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link  
48 INFO: [v++ 60-1441] [00:28:51] Run run_link: Step cf2sw: Completed  
49 Time (s): cpu = 00:00:16 ; elapsed = 00:00:18 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 46018  
50 INFO: [v++ 60-1443] [00:28:51] Run run_link: Step rtd2_system_diagram: Started  
51 INFO: [v++ 60-1453] Command Line: rtd2SystemDiagram  
52 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link  
53 INFO: [v++ 60-1441] [00:29:02] Run run_link: Step rtd2_system_diagram: Completed  
54 Time (s): cpu = 00:00:00.02 ; elapsed = 00:00:12 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 45918  
55 INFO: [v++ 60-1443] [00:29:02] Run run_link: Step vpl: Started  
56 INFO: [v++ 60-1453] Command Line: vpl-t hw-f xilinx_u200_xdma_201830_2--remote_ip_cache /iu_home/iu7102/lab_04/_x/  
57 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link  
58  
59 ***** vpl v2020.2 (64-bit)  
60 **** SW Build (by xbuild) on 2020-11-18-05:13:29  
61 ** Copyright 1986–2020 Xilinx, Inc. All Rights Reserved.  
62  
63 INFO: [VPL 60-839] Read in kernel information from file '/iu_home/iu7102/lab_04/_x/link/int/kernel_info.dat'.  
64 INFO: [VPL 74-74] Compiler Version string: 2020.2  
65 INFO: [VPL 60-423] Target device: xilinx_u200_xdma_201830_2  
66 INFO: [VPL 60-1032] Extracting hardware platform to /iu_home/iu7102/lab_04/_x/link/vivado/vpl/.local/hw_platform  
67 WARNING: /data/Xilinx/Vitis/2020.2/tps/lnx64/jre9.0.4 does not exist.  
68 [00:35:58] Run vpl: Step create_project: RUNNING...  
69 [00:35:52] Run vpl: Step create_project: Started  
70 Creating Vivado project.
```

```

71 [00:36:27] Run vpl: Step create_project: Completed
72 [00:36:27] Run vpl: Step create_bd: Started
73 [00:38:08] Run vpl: Step create_bd: RUNNING...
74 [00:39:55] Run vpl: Step create_bd: RUNNING...
75 [00:41:34] Run vpl: Step create_bd: RUNNING...
76 [00:43:28] Run vpl: Step create_bd: RUNNING...
77 [00:45:17] Run vpl: Step create_bd: RUNNING...
78 [00:46:58] Run vpl: Step create_bd: RUNNING...
79 [00:48:24] Run vpl: Step create_bd: Completed
80 [00:48:24] Run vpl: Step update_bd: Started
81 [00:48:27] Run vpl: Step update_bd: Completed
82 [00:48:27] Run vpl: Step generate_target: Started
83 [00:50:03] Run vpl: Step generate_target: RUNNING...
84 [00:51:44] Run vpl: Step generate_target: RUNNING...
85 [00:53:14] Run vpl: Step generate_target: RUNNING...
86 [00:54:49] Run vpl: Step generate_target: RUNNING...
87 [00:56:16] Run vpl: Step generate_target: RUNNING...
88 [00:57:50] Run vpl: Step generate_target: RUNNING...
89 [00:59:18] Run vpl: Step generate_target: RUNNING...
90 [01:00:49] Run vpl: Step generate_target: RUNNING...
91 [01:01:14] Run vpl: Step generate_target: Completed
92 [01:01:14] Run vpl: Step config_hw_runs: Started
93 [01:02:51] Run vpl: Step config_hw_runs: Completed
94 [01:02:51] Run vpl: Step synth: Started
95 [01:05:21] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
96 [01:05:56] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
97 [01:06:33] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
98 [01:07:11] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
99 [01:07:49] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
100 [01:08:26] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
101 [01:09:04] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
102 [01:09:41] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
103 [01:10:20] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
104 [01:10:58] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
105 [01:11:37] Block-level synthesis in progress, 0 of 66 jobs complete, 8 jobs running.
106 [01:12:14] Block-level synthesis in progress, 1 of 66 jobs complete, 7 jobs running.
107 [01:12:55] Block-level synthesis in progress, 4 of 66 jobs complete, 4 jobs running.
108 [01:13:32] Block-level synthesis in progress, 6 of 66 jobs complete, 2 jobs running.
109 [01:14:13] Block-level synthesis in progress, 6 of 66 jobs complete, 5 jobs running.
110 [01:14:50] Block-level synthesis in progress, 6 of 66 jobs complete, 8 jobs running.
111 [01:15:28] Block-level synthesis in progress, 7 of 66 jobs complete, 7 jobs running.
112 [01:16:05] Block-level synthesis in progress, 8 of 66 jobs complete, 6 jobs running.
113 [01:16:49] Block-level synthesis in progress, 8 of 66 jobs complete, 6 jobs running.
114 [01:17:26] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
115 [01:18:06] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
116 [01:18:44] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
117 [01:19:25] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
118 [01:20:01] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
119 [01:20:41] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
120 [01:21:19] Block-level synthesis in progress, 8 of 66 jobs complete, 8 jobs running.
121 [01:21:58] Block-level synthesis in progress, 10 of 66 jobs complete, 6 jobs running.
122 [01:22:37] Block-level synthesis in progress, 11 of 66 jobs complete, 5 jobs running.
123 [01:23:17] Block-level synthesis in progress, 13 of 66 jobs complete, 5 jobs running.
124 [01:23:56] Block-level synthesis in progress, 13 of 66 jobs complete, 6 jobs running.
125 [01:24:35] Block-level synthesis in progress, 13 of 66 jobs complete, 8 jobs running.
126 [01:25:12] Block-level synthesis in progress, 15 of 66 jobs complete, 6 jobs running.
127 [01:25:54] Block-level synthesis in progress, 15 of 66 jobs complete, 6 jobs running.
128 [01:26:34] Block-level synthesis in progress, 16 of 66 jobs complete, 7 jobs running.
129 [01:27:12] Block-level synthesis in progress, 16 of 66 jobs complete, 7 jobs running.
130 [01:27:50] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
131 [01:28:29] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
132 [01:29:06] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
133 [01:29:45] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
134 [01:30:24] Block-level synthesis in progress, 16 of 66 jobs complete, 8 jobs running.
135 [01:31:05] Block-level synthesis in progress, 17 of 66 jobs complete, 7 jobs running.
136 [01:31:45] Block-level synthesis in progress, 18 of 66 jobs complete, 6 jobs running.
137 [01:32:29] Block-level synthesis in progress, 19 of 66 jobs complete, 7 jobs running.
138 [01:33:09] Block-level synthesis in progress, 19 of 66 jobs complete, 7 jobs running.
139 [01:33:49] Block-level synthesis in progress, 19 of 66 jobs complete, 8 jobs running.
140 [01:34:31] Block-level synthesis in progress, 19 of 66 jobs complete, 8 jobs running.
141 [01:35:12] Block-level synthesis in progress, 20 of 66 jobs complete, 7 jobs running.
142 [01:35:51] Block-level synthesis in progress, 21 of 66 jobs complete, 6 jobs running.
143 [01:36:34] Block-level synthesis in progress, 21 of 66 jobs complete, 7 jobs running.
144 [01:37:13] Block-level synthesis in progress, 21 of 66 jobs complete, 8 jobs running.
145 [01:37:53] Block-level synthesis in progress, 22 of 66 jobs complete, 7 jobs running.
146 [01:38:33] Block-level synthesis in progress, 23 of 66 jobs complete, 6 jobs running.
147 [01:39:15] Block-level synthesis in progress, 23 of 66 jobs complete, 7 jobs running.
148 [01:39:54] Block-level synthesis in progress, 23 of 66 jobs complete, 8 jobs running.
149 [01:40:36] Block-level synthesis in progress, 24 of 66 jobs complete, 7 jobs running.
150 [01:41:15] Block-level synthesis in progress, 25 of 66 jobs complete, 6 jobs running.
151 [01:42:00] Block-level synthesis in progress, 26 of 66 jobs complete, 5 jobs running.
152 [01:42:39] Block-level synthesis in progress, 26 of 66 jobs complete, 7 jobs running.

```



```

235 [02:38:58] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
236 [02:39:37] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
237 [02:40:21] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
238 [02:41:02] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
239 [02:41:46] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
240 [02:42:27] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
241 [02:43:12] Block-level synthesis in progress, 65 of 66 jobs complete, 1 job running.
242 [02:43:52] Block-level synthesis in progress, 66 of 66 jobs complete, 0 jobs running.
243 [02:44:35] Block-level synthesis in progress, 66 of 66 jobs complete, 0 jobs running.
244 [02:45:15] Top-level synthesis in progress.
245 [02:45:56] Top-level synthesis in progress.
246 [02:46:36] Top-level synthesis in progress.
247 [02:47:18] Top-level synthesis in progress.
248 [02:47:58] Top-level synthesis in progress.
249 [02:48:42] Top-level synthesis in progress.
250 [02:49:21] Top-level synthesis in progress.
251 [02:50:04] Top-level synthesis in progress.
252 [02:50:45] Top-level synthesis in progress.
253 [02:51:29] Top-level synthesis in progress.
254 [02:52:12] Top-level synthesis in progress.
255 [02:52:54] Top-level synthesis in progress.
256 [02:53:35] Top-level synthesis in progress.
257 [02:54:19] Top-level synthesis in progress.
258 [02:55:00] Top-level synthesis in progress.
259 [02:55:42] Top-level synthesis in progress.
260 [02:56:22] Top-level synthesis in progress.
261 [02:57:23] Run vpl: Step synth: Completed
262 [02:57:23] Run vpl: Step impl: Started
263 [04:06:13] Finished 2nd of 6 tasks (FPGA linking synthesized kernels to platform). Elapsed time: 03h 36m 56s
264
265 [04:06:13] Starting logic optimization..
266 [04:14:14] Phase 1 Generate And Synthesize MIG Cores
267 [05:03:02] Phase 2 Generate And Synthesize Debug Cores
268 [05:38:01] Phase 3 Retarget
269 [05:41:36] Phase 4 Constant propagation
270 [05:43:43] Phase 5 Sweep
271 [05:51:29] Phase 6 BUFG optimization
272 [05:53:35] Phase 7 Shift Register Optimization
273 [05:55:01] Phase 8 Post Processing Netlist
274 [06:14:36] Finished 3rd of 6 tasks (FPGA logic optimization). Elapsed time: 02h 08m 22s
275
276 [06:14:36] Starting logic placement..
277 [06:20:47] Phase 1 Placer Initialization
278 [06:20:47] Phase 1.1 Placer Initialization Netlist Sorting
279 [06:38:14] Phase 1.2 IO Placement/ Clock Placement/ Build Placer Device
280 [06:50:52] Phase 1.3 Build Placer Netlist Model
281 [07:06:58] Phase 1.4 Constrain Clocks/Macros
282 [07:08:22] Phase 2 Global Placement
283 [07:09:02] Phase 2.1 Floorplanning
284 [07:13:56] Phase 2.1.1 Partition Driven Placement
285 [07:13:56] Phase 2.1.1.1 PBP: Partition Driven Placement
286 [07:16:44] Phase 2.1.1.2 PBP: Clock Region Placement
287 [07:22:21] Phase 2.1.1.3 PBP: Compute Congestion
288 [07:23:01] Phase 2.1.1.4 PBP: UpdateTiming
289 [07:25:46] Phase 2.1.1.5 PBP: Add part constraints
290 [07:27:12] Phase 2.2 Update Timing before SLR Path Opt
291 [07:27:12] Phase 2.3 Global Placement Core
292 [08:09:58] Phase 2.3.1 Physical Synthesis In Placer
293 [08:26:03] Phase 3 Detail Placement
294 [08:26:03] Phase 3.1 Commit Multi Column Macros
295 [08:26:47] Phase 3.2 Commit Most Macros & LUTRAMs
296 [08:33:48] Phase 3.3 Small Shape DP
297 [08:33:48] Phase 3.3.1 Small Shape Clustering
298 [08:36:40] Phase 3.3.2 Flow Legalize Slice Clusters
299 [08:37:21] Phase 3.3.3 Slice Area Swap
300 [08:42:59] Phase 3.4 Place Remaining
301 [08:43:43] Phase 3.5 Re-assign LUT pins
302 [08:45:50] Phase 3.6 Pipeline Register Optimization
303 [08:46:33] Phase 3.7 Fast Optimization
304 [08:51:25] Phase 4 Post Placement Optimization and Clean-Up
305 [08:51:25] Phase 4.1 Post Commit Optimization
306 [09:03:15] Phase 4.1.1 Post Placement Optimization
307 [09:03:56] Phase 4.1.1.1 BUFG Insertion
308 [09:03:56] Phase 1 Physical Synthesis Initialization
309 [09:07:30] Phase 4.1.1.2 BUFG Replication
310 [09:11:41] Phase 4.1.1.3 Replication
311 [09:19:21] Phase 4.2 Post Placement Cleanup
312 [09:20:47] Phase 4.3 Placer Reporting
313 [09:20:47] Phase 4.3.1 Print Estimated Congestion
314 [09:22:55] Phase 4.4 Final Placement Cleanup
315 [10:48:13] Finished 4th of 6 tasks (FPGA logic placement). Elapsed time: 04h 33m 37s
316

```

```

317 [10:48:13] Starting logic routing..
318 [10:55:05] Phase 1 Build RT Design
319 [11:08:18] Phase 2 Router Initialization
320 [11:08:18] Phase 2.1 Fix Topology Constraints
321 [11:09:02] Phase 2.2 Pre Route Cleanup
322 [11:09:42] Phase 2.3 Global Clock Net Routing
323 [11:13:15] Phase 2.4 Update Timing
324 [11:29:15] Phase 2.5 Update Timing for Bus Skew
325 [11:29:15] Phase 2.5.1 Update Timing
326 [11:35:33] Phase 3 Initial Routing
327 [11:35:33] Phase 3.1 Global Routing
328 [11:41:46] Phase 4 Rip-up And Reroute
329 [11:41:46] Phase 4.1 Global Iteration 0
330 [12:09:00] Phase 4.2 Global Iteration 1
331 [12:16:35] Phase 4.3 Global Iteration 2
332 [12:23:42] Phase 5 Delay and Skew Optimization
333 [12:23:42] Phase 5.1 Delay CleanUp
334 [12:23:42] Phase 5.1.1 Update Timing
335 [12:32:09] Phase 5.2 Clock Skew Optimization
336 [12:32:53] Phase 6 Post Hold Fix
337 [12:32:53] Phase 6.1 Hold Fix Iter
338 [12:32:53] Phase 6.1.1 Update Timing
339 [12:39:49] Phase 7 Route finalize
340 [12:39:49] Phase 8 Verifying routed nets
341 [12:41:53] Phase 9 Depositing Routes
342 [12:46:47] Phase 10 Route finalize
343 [12:46:47] Phase 11 Post Router Timing
344 [12:55:03] Finished 5th of 6 tasks (FPGA routing). Elapsed time: 02h 06m 50s
345
346 [12:55:03] Starting bitstream generation..
347 [15:02:45] Creating bitmap...
348 [15:55:25] Writing bitstream ./pfm_top_i_dynamic_region_my_rm_partial.bit...
349 [15:56:05] Finished 6th of 6 tasks (FPGA bitstream generation). Elapsed time: 03h 01m 01s
350 [16:00:11] Run vpl: Step impl: Completed
351 [16:00:24] Run vpl: FINISHED. Run Status: impl Complete!
352 INFO: [v++ 60-1441] [16:01:01] Run run_link: Step vpl: Completed
353 Time (s): cpu = 01:17:25 ; elapsed = 15:31:58 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 19007
354 INFO: [v++ 60-1443] [16:01:01] Run run_link: Step rtdgen: Started
355 INFO: [v++ 60-1453] Command Line: rtdgen
356 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link
357 INFO: [v++ 60-991] clock name 'clkwiz_kernel_clk_out1' (clock ID '0') is being mapped to clock name 'DATA_CLK' in the
358 INFO: [v++ 60-991] clock name 'clkwiz_kernel2_clk_out1' (clock ID '1') is being mapped to clock name 'KERNEL_CLK' in the
359 INFO: [v++ 60-1230] The compiler selected the following frequencies for the runtime controllable kernel clock(s) and
360 INFO: [v++ 60-1453] Command Line: cf2sw -a /iu_home/iu7102/lab_04/_x/link/int/address_map.xml -sdsl /iu_home/iu7102/
361 INFO: [v++ 60-1652] Cf2sw returned exit code: 0
362 INFO: [v++ 60-2311] HPISystemDiagram::writeSystemDiagramAfterRunningVivado, rtdInputFilePath: /iu_home/iu7102/lab_04/_x/link/int/address_map.xml
363 INFO: [v++ 60-2312] HPISystemDiagram::writeSystemDiagramAfterRunningVivado, systemDiagramOutputFilePath: /iu_home/iu7102/lab_04/_x/link/int/address_map.xml
364 INFO: [v++ 60-1618] Launching
365 INFO: [v++ 60-1441] [16:01:14] Run run_link: Step rtdgen: Completed
366 Time (s): cpu = 00:00:11 ; elapsed = 00:00:13 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 17360
367 INFO: [v++ 60-1443] [16:01:14] Run run_link: Step xlbinutil: Started
368 INFO: [v++ 60-1453] Command Line: xlbinutil --add-section DEBUG_IP_LAYOUT:JSON:/iu_home/iu7102/lab_04/_x/link/int/address_map.xml
369 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link
370 XRT Build Version: 2.8.743 (2020.2)
371     Build Date: 2020-11-16 00:19:11
372     Hash ID: 77d5484b5c4daa691af78235053fb036829b1e9
373 Creating a default 'in-memory' xlbin image.
374
375 Section: 'DEBUG_IP_LAYOUT'(9) was successfully added.
376 Size : 440 bytes
377 Format : JSON
378 File : '/iu_home/iu7102/lab_04/_x/link/int/debug_ip_layout.rtd'
379
380 Section: 'BITSTREAM'(0) was successfully added.
381 Size : 42618246 bytes
382 Format : RAW
383 File : '/iu_home/iu7102/lab_04/_x/link/int/partial.bit'
384
385 Section: 'MEM_TOPOLOGY'(6) was successfully added.
386 Format : JSON
387 File : 'mem_topology'
388
389 Section: 'IP_LAYOUT'(8) was successfully added.
390 Format : JSON
391 File : 'ip_layout'
392
393 Section: 'CONNECTIVITY'(7) was successfully added.
394 Format : JSON
395 File : 'connectivity'
396
397 Section: 'CLOCK_FREQ_TOPOLOGY'(11) was successfully added.
398 Size : 274 bytes

```

```

399 Format : JSON
400 File    : '/iu_home/iu7102/lab_04/_x/link/int/vinc_xml.rtd'
401
402 Section: 'BUILD_METADATA'(14) was successfully added.
403 Size   : 3057 bytes
404 Format : JSON
405 File   : '/iu_home/iu7102/lab_04/_x/link/int/vinc_build.rtd'
406
407 Section: 'EMBEDDED_METADATA'(2) was successfully added.
408 Size   : 2754 bytes
409 Format : RAW
410 File   : '/iu_home/iu7102/lab_04/_x/link/int/vinc.xml'
411
412 Section: 'SYSTEM_METADATA'(22) was successfully added.
413 Size   : 5761 bytes
414 Format : RAW
415 File   : '/iu_home/iu7102/lab_04/_x/link/int/systemDiagramModelSlrBaseAddress.json'
416
417 Section: 'IP_LAYOUT'(8) was successfully appended to.
418 Format : JSON
419 File   : 'ip_layout'
420 Successfully wrote (42640602 bytes) to the output file: /iu_home/iu7102/lab_04/vinc.xclbin
421 Leaving xclbinutil.
422 INFO: [v++ 60-1441] [16:01:17] Run run_link: Step xclbinutil: Completed
423 Time (s): cpu = 00:00:00.47 ; elapsed = 00:00:02 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 1738
424 INFO: [v++ 60-1443] [16:01:17] Run run_link: Step xclbinutilinfo: Started
425 INFO: [v++ 60-1453] Command Line: xclbinutil --quiet --force --info /iu_home/iu7102/lab_04/vinc.xclbin.info --input /
426 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link
427 INFO: [v++ 60-1441] [16:01:20] Run run_link: Step xclbinutilinfo: Completed
428 Time (s): cpu = 00:00:03 ; elapsed = 00:00:03 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 17881
429 INFO: [v++ 60-1443] [16:01:20] Run run_link: Step generate_sc_driver: Started
430 INFO: [v++ 60-1453] Command Line:
431 INFO: [v++ 60-1454] Run Directory: /iu_home/iu7102/lab_04/_x/link/run_link
432 INFO: [v++ 60-1441] [16:01:20] Run run_link: Step generate_sc_driver: Completed
433 Time (s): cpu = 00:00:00.01 ; elapsed = 00:00:00.03 . Memory (MB): peak = 1585.129 ; gain = 0.000 ; free physical = 17881
434 INFO: [v++ 60-244] Generating system estimate report...
435 INFO: [v++ 60-1092] Generated system estimate report: /iu_home/iu7102/lab_04/_x/reports/link/system_estimate_vinc.xml
436 INFO: [v++ 60-586] Created /iu_home/iu7102/lab_04/vinc.lt
437 INFO: [v++ 60-586] Created /iu_home/iu7102/lab_04/vinc.xclbin
438 INFO: [v++ 60-1307] Run completed. Additional information can be found in:
        Guidance: /iu_home/iu7102/lab_04/_x/reports/link/v++_link_vinc_guidance.html
        Timing Report: /iu_home/iu7102/lab_04/_x/reports/link/imp/impl_1_xilinx_u200_xdma_201830_2_bb_locked_timing_summary.html
        Vivado Log: /iu_home/iu7102/lab_04/_x/logs/link/vivado.log
        Steps Log File: /iu_home/iu7102/lab_04/_x/logs/link/link.steps.log
439
440 INFO: [v++ 60-2343] Use the vitis_analyzer tool to visualize and navigate the relevant reports. Run the following command:
        vitis_analyzer /iu_home/iu7102/lab_04/vinc.xclbin.link_summary
441 INFO: [v++ 60-791] Total elapsed time: 15h 36m 14s
442 INFO: [v++ 60-1653] Closing dispatch client.

```

Приложение Б

Листинг 4.3 – Листинг файла vinc.xclbin.info

```
1
2 -----
3 XRT Build Version: 2.8.743 (2020.2)
4     Build Date: 2020-11-16 00:19:11
5         Hash ID: 77d5484b5c4daa691a7f78235053fb036829b1e9
6 -----
7 xclbin Information
8 -----
9     Generated by:           v++ (2020.2) on 2020-11-18-05:13:29
10    Version:                2.8.743
11    Kernels:                rtl_kernel_wizard_0
12    Signature:              -
13    Content:                Bitstream
14    UUID (xclbin):          b348d66d-acaa4-4862-be75-402313af40a2
15    Sections:               DEBUG_IP_LAYOUT, BITSTREAM, MEM_TOPOLOGY, IP_LAYOUT,
16                           CONNECTIVITY, CLOCK_FREQ_TOPOLOGY, BUILD_METADATA,
17                           EMBEDDED_METADATA, SYSTEM_METADATA,
18                           GROUP_CONNECTIVITY, GROUP_TOPOLOGY
19 -----
20 Hardware Platform (Shell) Information
21 -----
22     Vendor:                 xilinx
23     Board:                  u200
24     Name:                   xdma
25     Version:                201830.2
26     Generated Version:      Vivado 2018.3 (SW Build: 2568420)
27     Created:                Tue Jun 25 06:55:20 2019
28     FPGA Device:            xcu200
29     Board Vendor:          xilinx.com
30     Board Name:             xilinx.com:au200:1.0
31     Board Part:             xilinx.com:au200:part0:1.0
32     Platform VBNV:          xilinx_u200_xdma_201830_2
33     Static UUID:            c102e7af-b2b8-4381-992b-9a00cc3863eb
34     Feature ROM TimeStamp: 1561465320
35
36 Clocks
37 -----
38     Name:       DATA_CLK
39     Index:     0
40     Type:      DATA
41     Frequency: 300 MHz
42
43     Name:       KERNEL_CLK
44     Index:     1
45     Type:      KERNEL
46     Frequency: 500 MHz
47
48 Memory Configuration
49 -----
50     Name:       bank0
51     Index:     0
52     Type:      MEM_DDR4
53     Base Address: 0x4000000000
54     Address Size: 0x4000000000
55     Bank Used:  No
56
57     Name:       bank1
58     Index:     1
59     Type:      MEM_DDR4
60     Base Address: 0x5000000000
61     Address Size: 0x4000000000
62     Bank Used:  No
63
64     Name:       bank2
65     Index:     2
66     Type:      MEM_DDR4
67     Base Address: 0x6000000000
68     Address Size: 0x4000000000
69     Bank Used:  No
70
71     Name:       bank3
72     Index:     3
73     Type:      MEM_DDR4
```

```

74 Base Address: 0x7000000000
75 Address Size: 0x400000000
76 Bank Used: Yes
77
78 Name: PLRAM[0]
79 Index: 4
80 Type: MEM_DRAM
81 Base Address: 0x3000000000
82 Address Size: 0x20000
83 Bank Used: No
84
85 Name: PLRAM[1]
86 Index: 5
87 Type: MEM_DRAM
88 Base Address: 0x3000200000
89 Address Size: 0x20000
90 Bank Used: No
91
92 Name: PLRAM[2]
93 Index: 6
94 Type: MEM_DRAM
95 Base Address: 0x3000400000
96 Address Size: 0x20000
97 Bank Used: No
98
99 Kernel: rtl_kernel_wizard_0
100
101 Definition
102
103 Signature: rtl_kernel_wizard_0 (uint num, int* axi00_ptr0)
104
105 Ports
106
107 Port: s_axi_control
108 Mode: slave
109 Range (bytes): 0x1000
110 Data Width: 32 bits
111 Port Type: addressable
112
113 Port: m00_axi
114 Mode: master
115 Range (bytes): 0xFFFFFFFFFFFFFF
116 Data Width: 512 bits
117 Port Type: addressable
118
119
120 Instance: vinc0
121 Base Address: 0x1e00000
122
123 Argument: num
124 Register Offset: 0x010
125 Port: s_axi_control
126 Memory: <not applicable>
127
128 Argument: axi00_ptr0
129 Register Offset: 0x018
130 Port: m00_axi
131 Memory: bank3 (MEM_DDR4)
132
133 Generated By
134
135 Command: v++
136 Version: 2020.2 - 2020-11-18-05:13:29 (SW BUILD: 0)
137 Command Line: v++ --config /iu_home/iu7102/lab_04/lab_04/rtl_kernel_wizard_0.cfg
138 --connectivity.nk rtl_kernel_wizard_0:1:vinc0 --connectivity.slr vinc0:SLR2 --connectivity.sp
139 vinc0.m00_axi:DDR[3] --input_files /iu_home/iu7102/lab_04/lab_04_kernels/vivado_rtl_kernel/rtl_kernel
140 _wizard_0_ex/exports/rtl_kernel_wizard_0.xo
141 --link --optimize 0 --output /iu_home/iu7102/lab_04/vinc.xelbin
142 --platform xilinx_u200_xdma_201830_2
143 --report_level 0 --target hw --vivado.prop run.impl_1.STEPS.OPT_DESIGN.ARGS.DIRECTIVE=Explore
144 --vivado.prop run.impl_1.STEPS.PLACE_DESIGN.ARGS.DIRECTIVE=Explore
145 --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.IS_ENABLED=true
146 --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.ARGS.DIRECTIVE=AggressiveExplore
147 --vivado.prop run.impl_1.STEPS.ROUTE_DESIGN.ARGS.DIRECTIVE=Explore
148 Options: --config /iu_home/iu7102/lab_04/lab_04/rtl_kernel_wizard_0.cfg
149 --connectivity.nk rtl_kernel_wizard_0:1:vinc0
150 --connectivity.slr vinc0:SLR2
151 --connectivity.sp vinc0.m00_axi:DDR[3]
152 --input_files /iu_home/iu7102/lab_04/lab_04_kernels/vivado_rtl_kernel/rtl_kernel_wizard_
153 0_ex/exports/rtl_kernel_wizard_0.xo
154 --link
155 --optimize 0

```

```
156      --output /iu_home/iu7102/lab_04/vinc.xclbin
157      --platform xilinx_u200_xdma_201830_2
158      --report_level 0
159      --target hw
160      --vivado.prop run.impl_1.STEPS.OPT_DESIGN.ARGS.DIRECTIVE=Explore
161      --vivado.prop run.impl_1.STEPS.PLACE_DESIGN.ARGS.DIRECTIVE=Explore
162      --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.IS_ENABLED=true
163      --vivado.prop run.impl_1.STEPS.PHYS_OPT_DESIGN.ARGS.DIRECTIVE=AggressiveExplore
164      --vivado.prop run.impl_1.STEPS.ROUTE_DESIGN.ARGS.DIRECTIVE=Explore
165
166 User Added Key Value Pairs
167 -----
168     <empty>
169 -----
```

Приложение B

Листинг 4.4 – Листинг файла host_example.cpp

```
1 // This is a generated file. Use and modify at your own risk.
2 //////////////////////////////////////////////////////////////////
3
4 //***** Vendor: Xilinx
5 // Associated Filename: main.c
6 //Purpose: This example shows a basic vector add +1 (constant) by manipulating
7 //          memory inplace.
8 //***** */
9
10 #include <fcntl.h>
11 #include <stdio.h>
12 #include <iostream>
13 #include <stdlib.h>
14 #include <string.h>
15 #include <math.h>
16 #ifdef _WINDOWS
17 #include <io.h>
18 #else
19 #include <unistd.h>
20 #include <sys/time.h>
21 #endif
22 #include <assert.h>
23 #include <stdbool.h>
24 #include <sys/types.h>
25 #include <sys/stat.h>
26 #include <CL/opencl.h>
27 #include <CL/cl_ext.h>
28 #include "xclhal2.h"
29
30 //////////////////////////////////////////////////////////////////
31
32 #define NUM_WORKGROUPS (1)
33 #define WORKGROUP_SIZE (256)
34 #define MAX_LENGTH 8192
35 #define MEM_ALIGNMENT 4096
36 #if defined(VITIS_PLATFORM) && !defined(TARGET_DEVICE)
37 #define STR_VALUE(arg) #arg
38 #define GET_STRING(name) STR_VALUE(name)
39 #define TARGET_DEVICE GET_STRING(VITIS_PLATFORM)
40 #endif
41
42 //////////////////////////////////////////////////////////////////
43
44 cl_uint load_file_to_memory(const char *filename, char **result)
45 {
46     cl_uint size = 0;
47     FILE *f = fopen(filename, "rb");
48     if (f == NULL) {
49         *result = NULL;
50         return -1; // -1 means file opening fail
51     }
52     fseek(f, 0, SEEK_END);
53     size = ftell(f);
54     fseek(f, 0, SEEK_SET);
55     *result = (char *)malloc(size+1);
56     if (size != fread(*result, sizeof(char), size, f)) {
57         free(*result);
58         return -2; // -2 means file reading fail
59     }
60     fclose(f);
61     (*result)[size] = 0;
62     return size;
63 }
64
65 int main(int argc, char** argv)
66 {
67
68     cl_int err; // error code returned from api calls
69     cl_uint check_status = 0;
70     const cl_uint number_of_words = 4096; // 16KB of data
71
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 mem
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 Platform
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 // Build the program executable
216 //
217 err = clBuildProgram(program, 0, NULL, NULL, NULL, NULL);
218 if (err != CL_SUCCESS) {
219     size_t len;
220     char buffer[2048];
221
222     printf("ERROR: Failed to build program executable!\n");
223     clGetProgramBuildInfo(program, device_id, CL_PROGRAM_BUILD_LOG, sizeof(buffer), buffer, &len);
224     printf("%s\n", buffer);
225     printf("ERROR: Test failed\n");
226     return EXIT_FAILURE;
227 }
228
229 // Create the compute kernel in the program we wish to run
230 //
231 kernel = clCreateKernel(program, "rtl_kernel_wizard_0", &err);
232 if (!kernel || err != CL_SUCCESS) {
233     printf("ERROR: Failed to create compute kernel!\n");
234     printf("ERROR: Test failed\n");
235     return EXIT_FAILURE;
236 }
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_
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);
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_num = 0;
271 err |= clSetKernelArg(kernel, 0, sizeof(cl_uint), &d_num); // 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_-
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 // Check Results
311
312
313 for (cl_uint i = 0; i < number_of_words; i++) {
314     if ((h_data[i] & 0xfofofofo + 10) != 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), outp
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
329 free(h_data);
330 clReleaseProgram(program);
331 clReleaseKernel(kernel);
332 clReleaseCommandQueue(commands);
333 clReleaseContext(context);
334
335 if (check_status) {
336     printf("ERROR: Test failed\n");
337     return EXIT_FAILURE;
338 } else {
339     printf("INFO: Test completed successfully.\n");
340     return EXIT_SUCCESS;
341 }
342
343
344 } // end of main
```