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

ФАКУЛЬТЕТ **ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭВМ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ (ИУ7)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 Программная инженерия**

О Т Ч Е Т

по лабораторной работе № 5

Название: Разработка ускорителей вычислений средствами САПР высокоуровневого синтеза Xilinx Vitis HLS

Дисциплина: Архитектура ЭВМ

Студент	ИУ7-54Б	28.12.2021	С.Д.Параскун
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			А.Ю.Попов
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2021

Цель работы: разработать ускоритель вычислений и код для его тестирования, изучить технологии синтеза аппаратных устройств ускорения вычислений по описаниям на языках высокого уровня, рассмотреть маршрут проектирования устройств, представленных на C/C++.

Индивидуальный вариант: 12.

Код варианта представлен на рисунке 1.

```
extern "C" {
void var012_no_pragmas(int* c, const int* a, const int* b, const int len) {
    int ptr = 0;
    for (int i = 0; i < len; i++) {
        ptr = b[i] % len;
        c[i] = a[ptr] + i;
     }
}
}
```

Рисунок 1 — Код индивидуального варианта

Функция обрабатывает два массива и формирует третий, в котором i-ый элемент является суммой текущего значения индекса и элементом первого массива, расположенным по индексу, равному остатку от деления i-го элемента второго массива на его длину. На основе этого кода было создано еще три файла, в которых были добавлены дополнительные директивы, позволяющие реализовать частично развернутый цикл, конвейерное исполнение цикла и частично развернутый конвейерный цикл.
Три новых функции представлены на рисунках 2-4.

```
extern "C" {
void var012_pipelined(int* c, const int* a, const int* b, const int len) {
    int ptr = 0;
    for (int i = 0; i < len; i++) {
        #pragma HLS PIPELINE
        ptr = b[i] % len;
        c[i] = a[ptr] + i;
     }
}
}
```

Рисунок 2 — Конвейерная организация цикла

```
extern "C" {
void var012_unrolled(int* c, const int* a, const int* b, const int len) {
    int ptr = 0;
    for (int i = 0; i < len; i++) {
        #pragma HLS UNROLL  factor=2
        ptr = b[i] % len;
        c[i] = a[ptr] + i;
     }
}
}
```

Рисунок 3 — Частичное разворачивание цикла

```
1  extern "C" {
2  void var012_pipe_unroll(int* c, const int* a, const int* b, const int len) {
3      int ptr = 0;
4      for (int i = 0; i < len; i++) {
5          #pragma HLS PIPELINE
6          #pragma HLS UNROLL  factor=2
7          ptr = b[i] % len;
8          c[i] = a[ptr] + i;
9       }
10 }
11 }
```

Рисунок 4 — Конвейеризация и частичное разворачивание цикла

При развораччивании указывается значение factor = 2. Это означает, что число итераций цикла уменьшается в 2 раза за счет того, что операторы тела цикла повторяются дважды.

Сборка и отладка проекта осуществляется в трех режимах: программная эмуляция (Emulation-SW), аппаратная эмуляция (Emulation-HW), аппаратное исполнение (Hardware).

Emulation-SW.

Проект был собран в режиме программной эмуляции и запущен. Результаты работы приложения представлены на рисунке 5.

```
Found Platform
Platform Name: Xilinx
INFO: Reading /iu_home/iu7102/lab_05/lab_05_system/Emulation-SW/binary_container_1.xclbin
Loading: '/iu_home/iu7102/lab_05/lab_05_system/Emulation-SW/binary_container_1.xclbin'
Trying to program device[0]: xilinx_u200_xdma_201830_2
Device[0]: program successful!
|-------------------------+-------------------------|
| Kernel                  |    Wall-Clock Time (ns) |
|-------------------------+-------------------------|
| var012_no_pragmas       |                 2974716 |
|-------------------------+-------------------------|
| var012_unrolled         |                 5993197 |
|-------------------------+-------------------------|
| var012_pipelined        |                 6230501 |
|-------------------------+-------------------------|
| var012_pipe_unroll      |                  620815 |
|-------------------------+-------------------------|
Note: Wall Clock Time is meaningful for real hardware execution only, not for emulation.
Please refer to profile summary for kernel execution time for hardware emulation.
TEST PASSED.
```

Рисунок 5 — Результаты работы приложения в режиме Emulation-SW

Emulation-HW.

Проект был собран в режиме аппаратной эмуляции. В окне Assistant View можно посмотреть и изучить отчеты о сборке аппаратных ядер, где можно увидеть основные параметры сборки, используемую память, ширину шины. Для аппаратной отладки это окно наиболее информативно. На рисунке 6 представлено окно Assistant View для сборки Emulation-HW. При этой сборке многие значения установлены по умолчанию или определены автоматически.

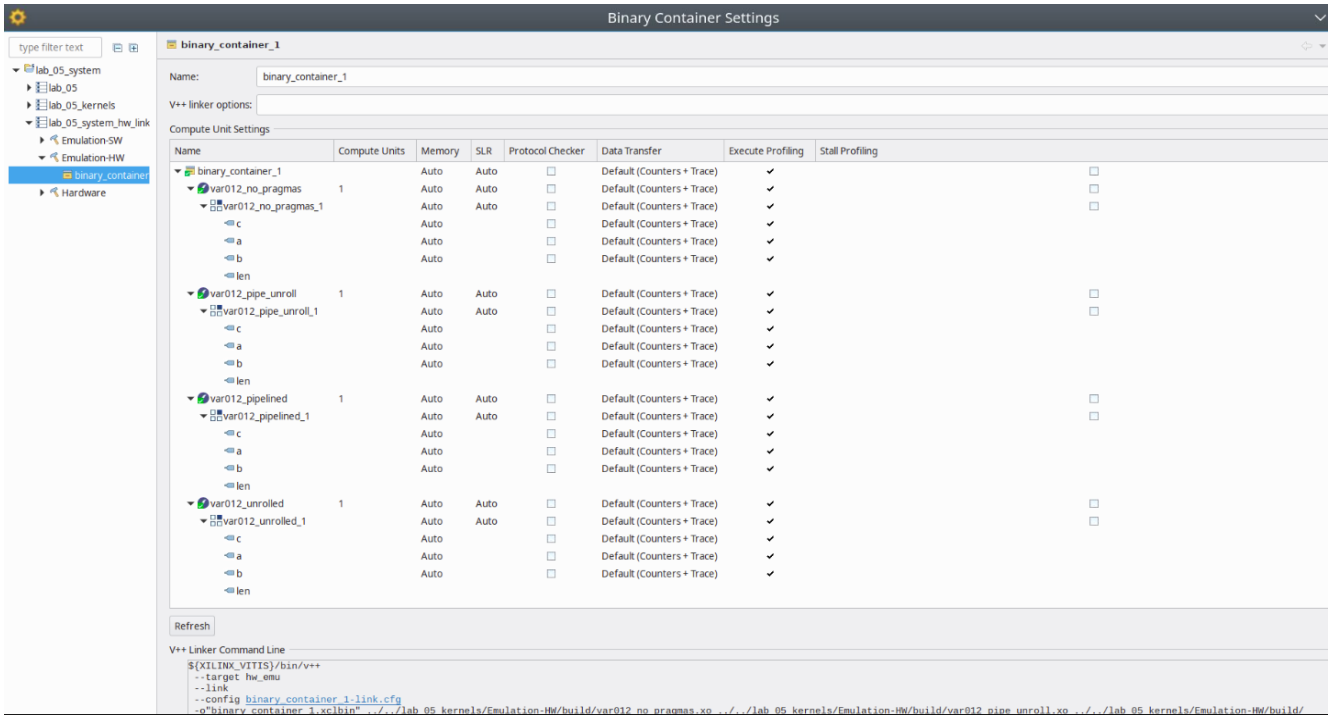

Рисунок 6 — Экран Assistant View

При запуске проекта открывается проект в Xilinx Vivado и запускается симуляция работы всех ускорительных ядер. В результате на диаграмме можно увидеть работу всех четырех запрограммированных ядер, как показано на рисунке 7.

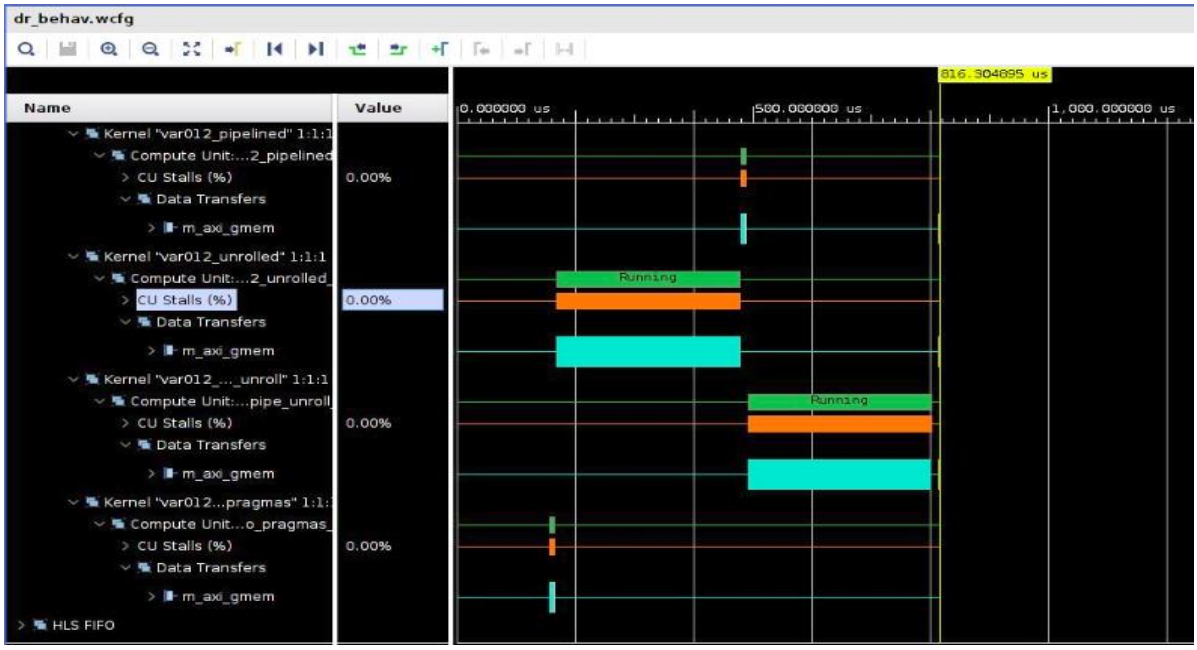

Рисунок 7 — Диаграммы работы четырех ядер

При этом в консоль при выполнении собранного проекта выводится следующая информация, приведенная на рисунках 8-9.

```
INFO: [HW-EMU 01] Hardware emulation runs simulation underneath. Using a large data set will re
INFO::[ Vitis-EM 22 ] [Time elapsed: 2 minute(s) 41 seconds, Emulation time: 0.0539458 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 0.000 KB              WR = 0.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 0.000 KB              WR = 0.000 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 0.000 KB              WR = 0.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 0.000 KB              WR = 0.000 KB

Device[0]: program successful!
|-------------------------+-------------------------|
| Kernel                  |    Wall-Clock Time (ns) |
|-------------------------+-------------------------|
| var012_no_pragmas       |             38009500899 |
|-------------------------+-------------------------|
INFO::[ Vitis-EM 22 ] [Time elapsed: 7 minute(s) 41 seconds, Emulation time: 0.168249 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 0.000 KB              WR = 0.000 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 0.000 KB              WR = 0.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 0.000 KB              WR = 0.000 KB

INFO::[ Vitis-EM 22 ] [Time elapsed: 12 minute(s) 42 seconds, Emulation time: 0.275843 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 0.000 KB              WR = 0.000 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 0.000 KB              WR = 0.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 2.773 KB              WR = 1.375 KB

INFO::[ Vitis-EM 22 ] [Time elapsed: 17 minute(s) 44 seconds, Emulation time: 0.385262 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 0.000 KB              WR = 0.000 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 0.000 KB              WR = 0.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 5.609 KB              WR = 2.750 KB
```

Рисунок 8 — Результаты работы приложения в режиме Emulation-HW

```
| var012_unrolled         |            846225693645 |
|-------------------------+-------------------------|
| var012_pipelined        |             31013149840 |
|-------------------------+-------------------------|
INFO::[ Vitis-EM 22 ] [Time elapsed: 22 minute(s) 49 seconds, Emulation time: 0.504529 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 0.312 KB              WR = 0.125 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB

INFO::[ Vitis-EM 22 ] [Time elapsed: 27 minute(s) 49 seconds, Emulation time: 0.581895 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 2.312 KB              WR = 1.125 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB

INFO::[ Vitis-EM 22 ] [Time elapsed: 32 minute(s) 52 seconds, Emulation time: 0.687996 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 5.062 KB              WR = 2.500 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB

INFO::[ Vitis-EM 22 ] [Time elapsed: 37 minute(s) 52 seconds, Emulation time: 0.799625 ms]
Data transfer between kernel(s) and global memory(s)
var012_no_pragmas_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_pipe_unroll_1:m_axi_gmem-DDR[1]         RD = 7.953 KB              WR = 3.938 KB
var012_pipelined_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB
var012_unrolled_1:m_axi_gmem-DDR[1]          RD = 8.000 KB              WR = 4.000 KB

| var012_pipe_unroll      |            947246719029 |
|-------------------------+-------------------------|
Note: Wall Clock Time is meaningful for real hardware execution only, not for emulation.
Please refer to profile summary for kernel execution time for hardware emulation.
TEST PASSED.
```

Рисунок 9 — Результаты работы приложения в режиме Emulation-HW

Здесь время работы каждой функции отличается от предыдущей сборки, но рассматривать его как истинный результат нельзя. Такое время работы обусловлено тем, что при выполнении самих функций параллельно также производилось моделирование ядер в Vivado, что замедляло работу программы, так как для каждого такта необходимо производить вычисления и отображать результаты на диаграмме.

Hardware.

Проект был собран в режиме аппаратного исполнения. В окне Assistant можно найти сводный отчет Link Summary, который можно открыть в среде Vitis Analyser. Здесь хранится информация о собранных ядрах, о времени сборки, схема собранной системы и многое другое. На рисунках 10-16 показаны скриншоты из данной среды, на которых представлена общая информация о прошедшей сборке (Summary), схему ускорителя и непосредственно ускорительной карты (System Diagram, Platform Diagram), а также информация о HLS синтезе каждого из четырех ядер.

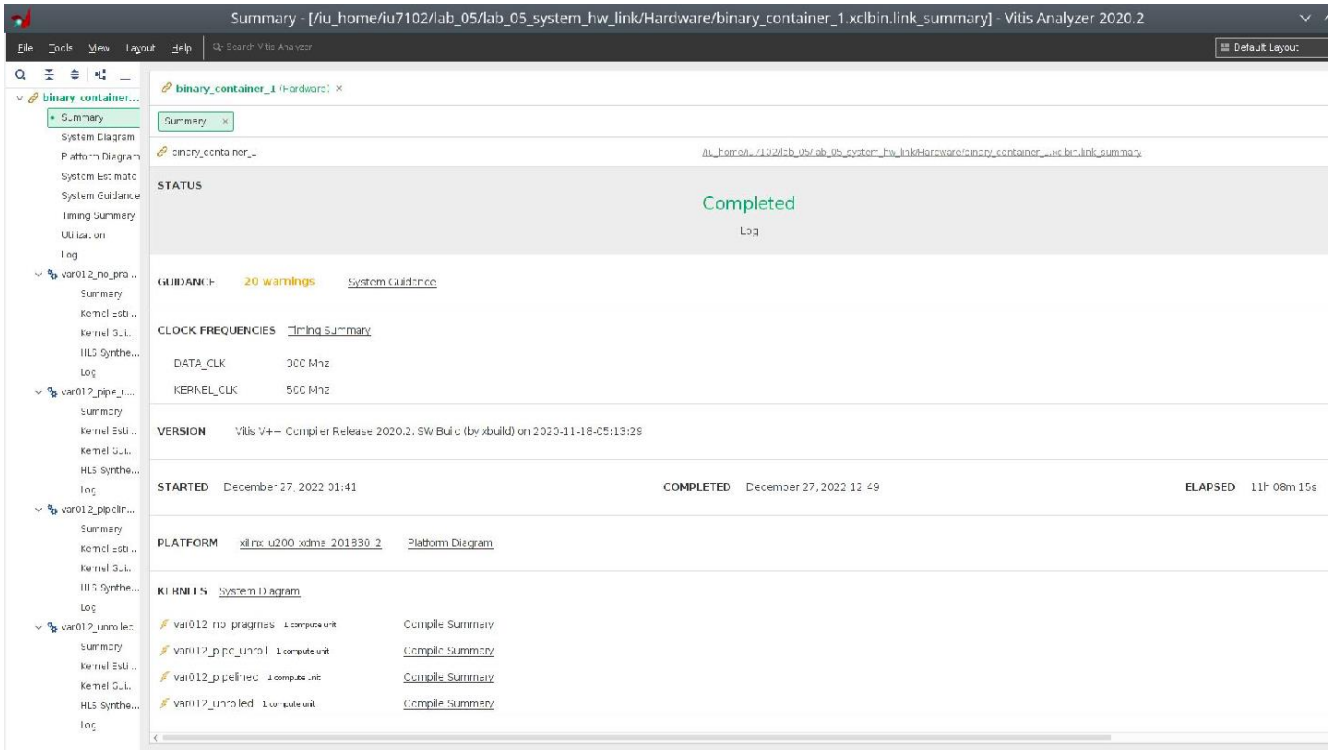

Рисунок 10 — Общая информация о сборке (Summary)

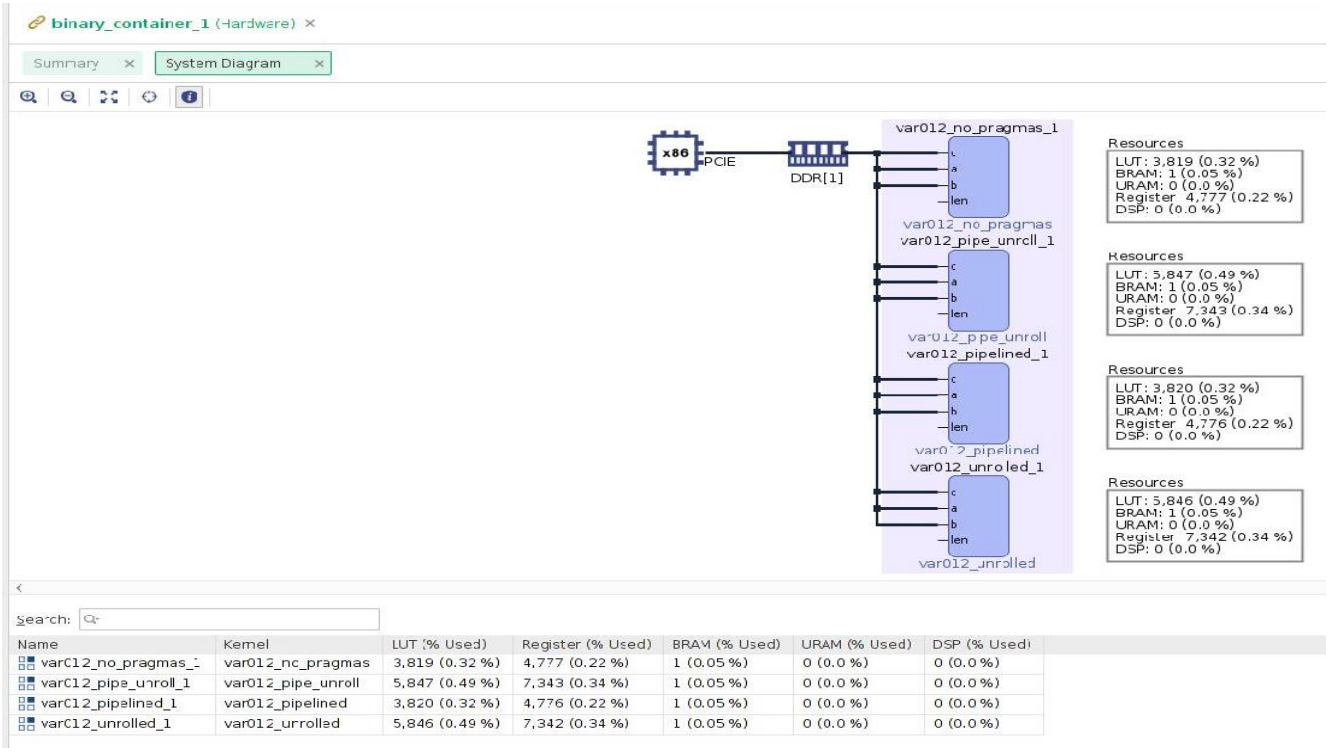

Рисунок 11 — System Diagram

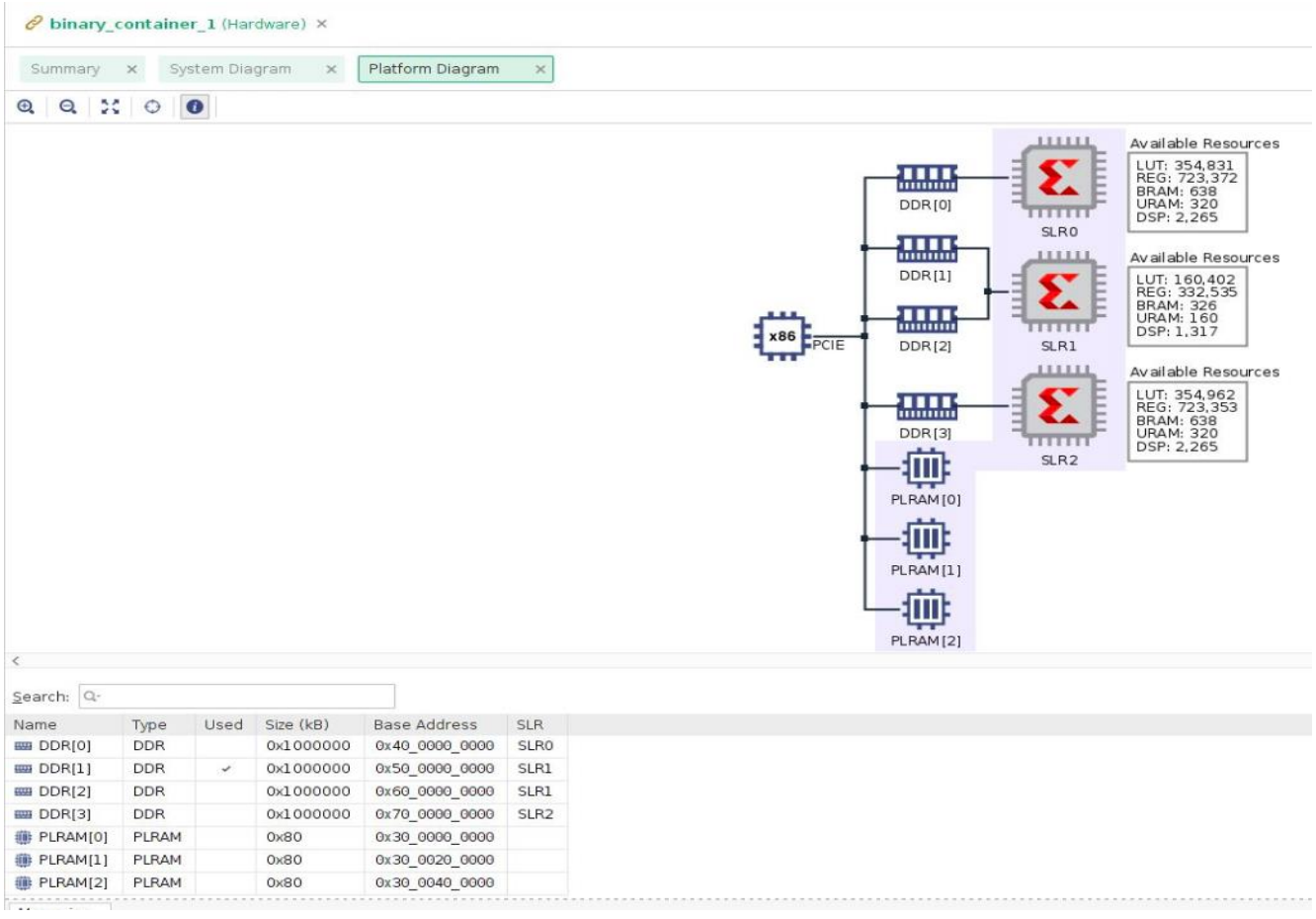

Рисунок 12 — Platform Diagram

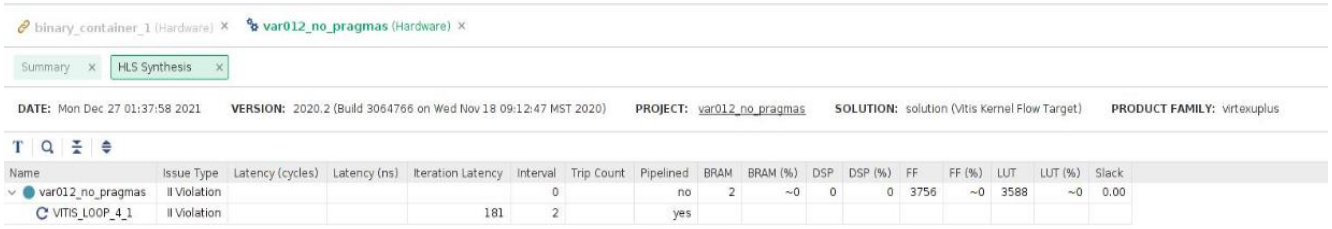

Рисунок 13 — Информация о ядре без оптимизаций цикла

Рисунок 14 — Информация о ядре с конвейерным циклом

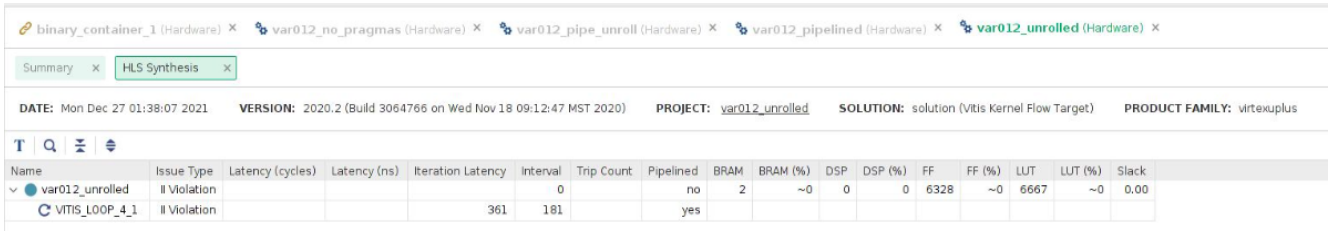

Рисунок 15 — Информация о ядре с частично развернутым циклом

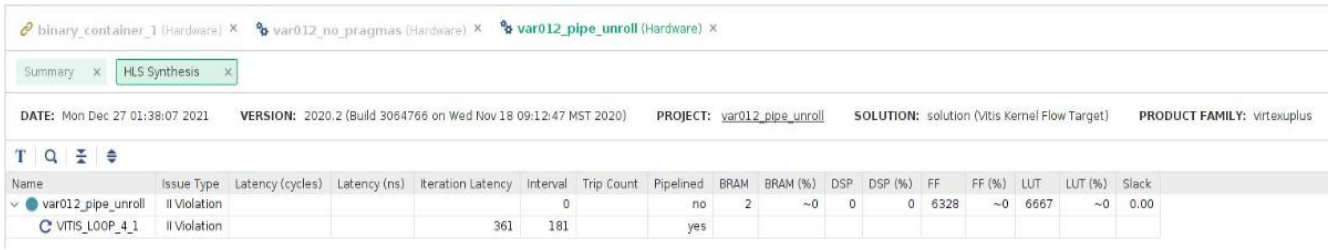

Рисунок 16 — Информация о ядре с конвейерным частично развернутым циклом

В результате запуска собранного приложения в консоли отобразилась таблица, в которой приведено количество времени, затрачиваемого на выполнение задачи на разных ускорительных ядрах. Она приведена на рисунке 17.

```
[Console output redirected to file:/iu_home/iu7102/lab_05/lab_05/Hardware/SystemDebugger_lab_05_system_lab_05.launch.log]
Found Platform
Platform Name: Xilinx
INFO: Reading /iu_home/iu7102/lab_05/lab_05_system/Hardware/binary_container_1.xclbin
Loading: '/iu_home/iu7102/lab_05/lab_05_system/Hardware/binary_container_1.xclbin'
Trying to program device[0]: xilinx_u200_xdma_201830_2
Device[0]: program successful!
|-------------------------+-------------------------|
| Kernel                  |    Wall-Clock Time (ns) |
|-------------------------+-------------------------|
| var012_no_pragmas       |                60673222 |
|-------------------------+-------------------------|
| var012_unrolled         |               348666844 |
|-------------------------+-------------------------|
| var012_pipelined        |                44132466 |
|-------------------------+-------------------------|
| var012_pipe_unroll      |               348922273 |
|-------------------------+-------------------------|
Note: Wall Clock Time is meaningful for real hardware execution only, not for emulation.
Please refer to profile summary for kernel execution time for hardware emulation.
TEST PASSED.
```

Рисунок 17 — Результаты работы приложения в режиме Hardware

По результатам работы можно судить, что конвейеризация цикла дает совсем выигрыш времени порядка 25% по сравнению с неоптимизированным циклом, значит, данный алгоритм неплохо поддается конвейеризации.

В то же время разворачивание цикла вообще замедляет выполнение задачи почти в 6 раз. Причиной этому является зависимость по данным — развернутые итерации цикла должны выполняться параллельно, но при этом вычисление элемента результирующего массива требует информации об индексе ptr, соответственно, он не должен измениться прежде, чем будет использован в теле цикла. Также возникают дополнительные расходы времени на проверку выхода за границы массива и диспетчеризацию параллельных потоков.

Вывод

В результате выполнения лабораторной работы были смоделированы в трех различных режимах четыре ядра с разными уровнями оптимизаций, изучена технология синтеза ускорителей на языках высокого уровня на примере C/C++, разработан и протестирован ускоритель вычислений. В результате тестирования и замеров времени работы было установлено, что конвейеризация дает ощутимый выигрыш во времени, а разворачивание цикла не позволяет оптимизировать задачу, соответствующую индивидуальному варианту.