|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.04 ПРОГРАММНАЯ ИНЖЕНЕРИЯ**

**Отчет**

|  |  |
| --- | --- |
| **по лабораторной работе №** | 5 |

**Название:**

Разработка ускорителей вычислений средствами САПР высокоуровневого синтеза Xilinx Vitis HLS

**Дисциплина:** Архитектура ЭВМ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ7-56Б |  | 20.12.2021 | В.В. Леонов |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | А.Ю. Попов |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2021

# Введение

Целью данной работы является изучение методики и технологии синтеза аппаратных устройств ускорения вычислений по описаниям на языках высокого уровня. В ходе лабораторной работы рассматривается маршрут проектирования устройств, представленных в виде синтаксических конструкций ЯВУ C/C++, изучаются принципы работы IDE Xilinx Vitis HLS и методика анализа и отладки устройств. В ходе работы необходимо разработать ускоритель вычислений по индивидуальному заданию, разработать код для тестирования ускорителя, реализовать ускоритель с помощью средств высоко-уровненного синтеза, выполнить его отладку.

# Индивидуальное задание

1. Неоптимизированный цикл.

extern "C"

{

void var008\_no\_pragmas(int \*c, const int \*a, const int \*b, const int len)

{

int min;

for (int i = 0; i < len; i++)

{

if (min > a[i])

min = a[i];

if (min > b[i])

min = b[i];

}

for (int i = 0; i < len; i++)

{

c[i] = min + i;

}

}

}

1. Конвейерная организация цикла.

extern "C"

{

void var008\_pipelined(int \*c, const int \*a, const int \*b, const int len)

{

int min;

for (int i = 0; i < len; i++)

{

if (min > a[i])

min = a[i];

if (min > b[i])

min = b[i];

}

for (int i = 0; i < len; i++)

{

#pragma HLS PIPELINE

c[i] = min + i;

}

}

}

1. Частично развернутый цикл.

extern "C"

{

void var008\_unrolled(int \*c, const int \*a, const int \*b, const int len)

{

int min;

for (int i = 0; i < len; i++)

{

#pragma HLS UNROLL factor = 5

if (min > a[i])

min = a[i];

if (min > b[i])

min = b[i];

}

for (int i = 0; i < len; i++)

{

#pragma HLS UNROLL factor = 5

c[i] = min + i;

}

}

}

1. Конвейерный и частично развернутый цикл.

extern "C"

{

void var008\_pipe\_unroll(int \*c, const int \*a, const int \*b, const int len)

{

int min;

for (int i = 0; i < len; i++)

{

#pragma HLS UNROLL factor = 5

if (min > a[i])

min = a[i];

if (min > b[i])

min = b[i];

}

for (int i = 0; i < len; i++)

{

#pragma HLS PIPELINE

c[i] = min + i;

}

}

}

# Сборка и отладка проекта в режиме программной эмуляции (Emulation-SW)

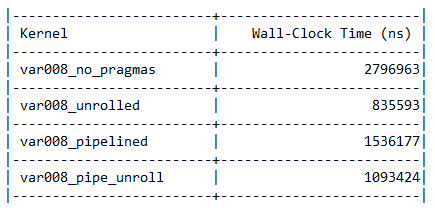


Рисунок 1. Результат работы программы

# Сборка и отладка проекта в режиме аппаратной эмуляции (Emulation-HW)

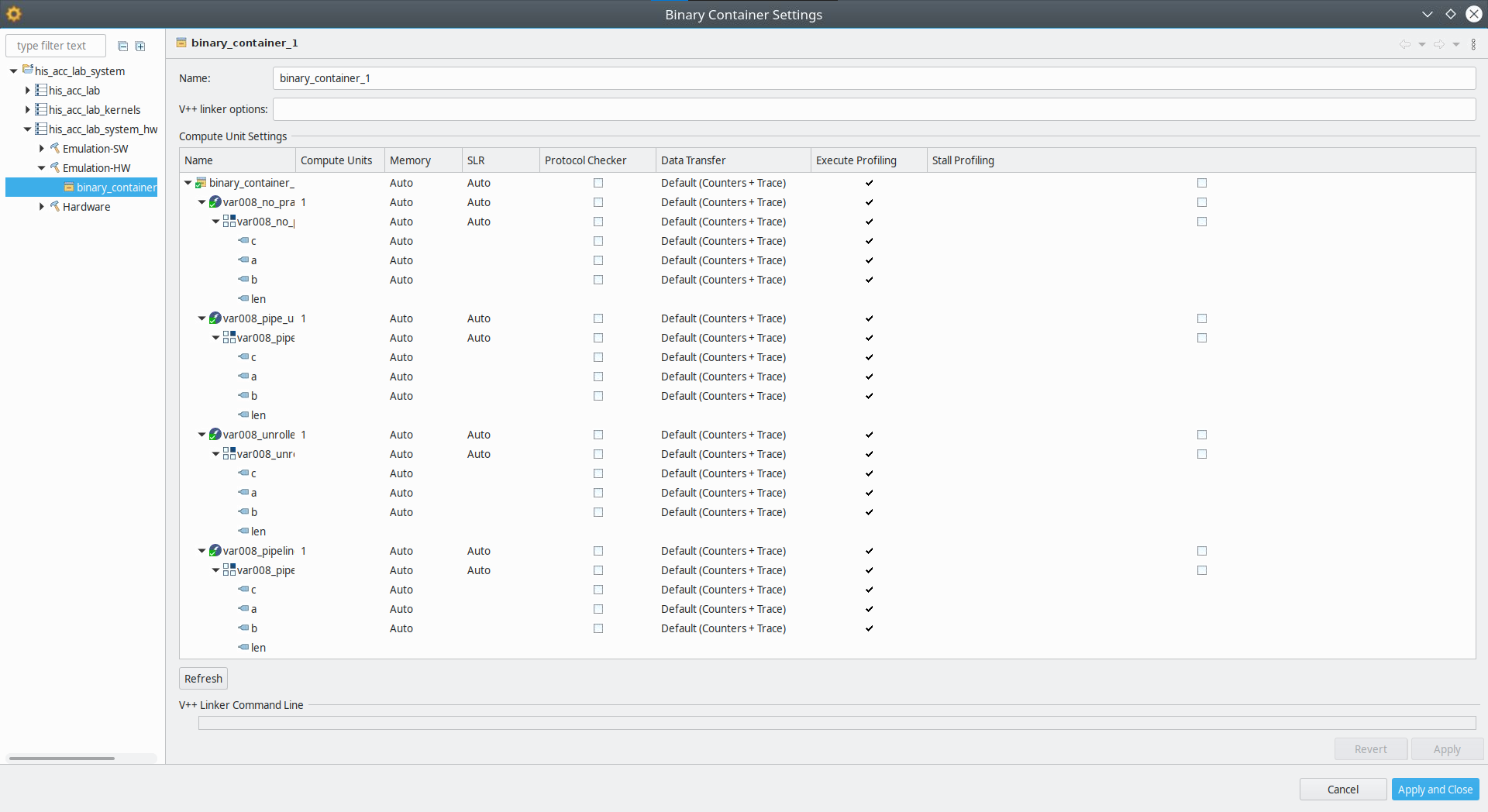


Рисунок 2. Assistant View

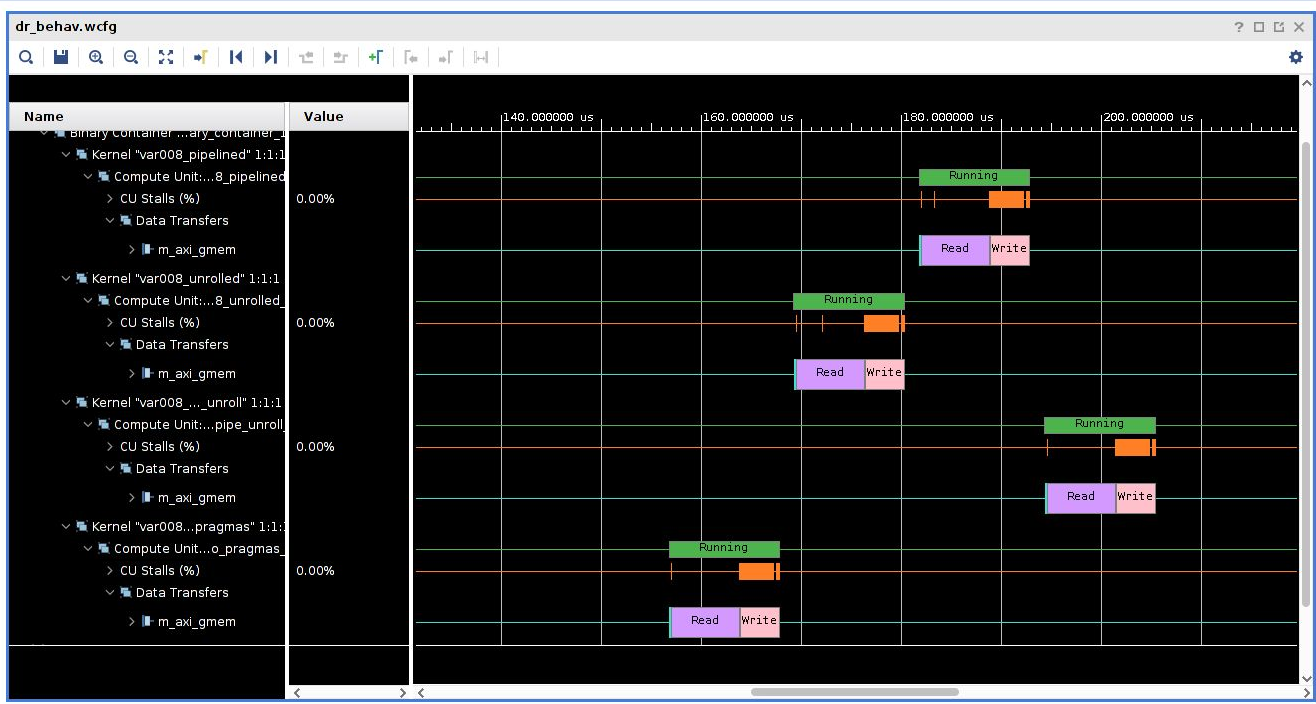


Рисунок . Окно внутрисхемового отладчика Vivado

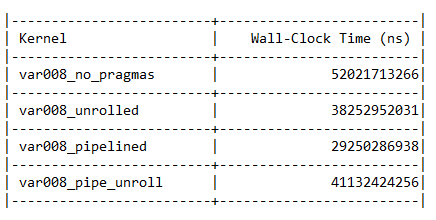


Рисунок . Результат работы программы

# Сборка и отладка проекта в режиме аппаратного исполнения (Hardware)

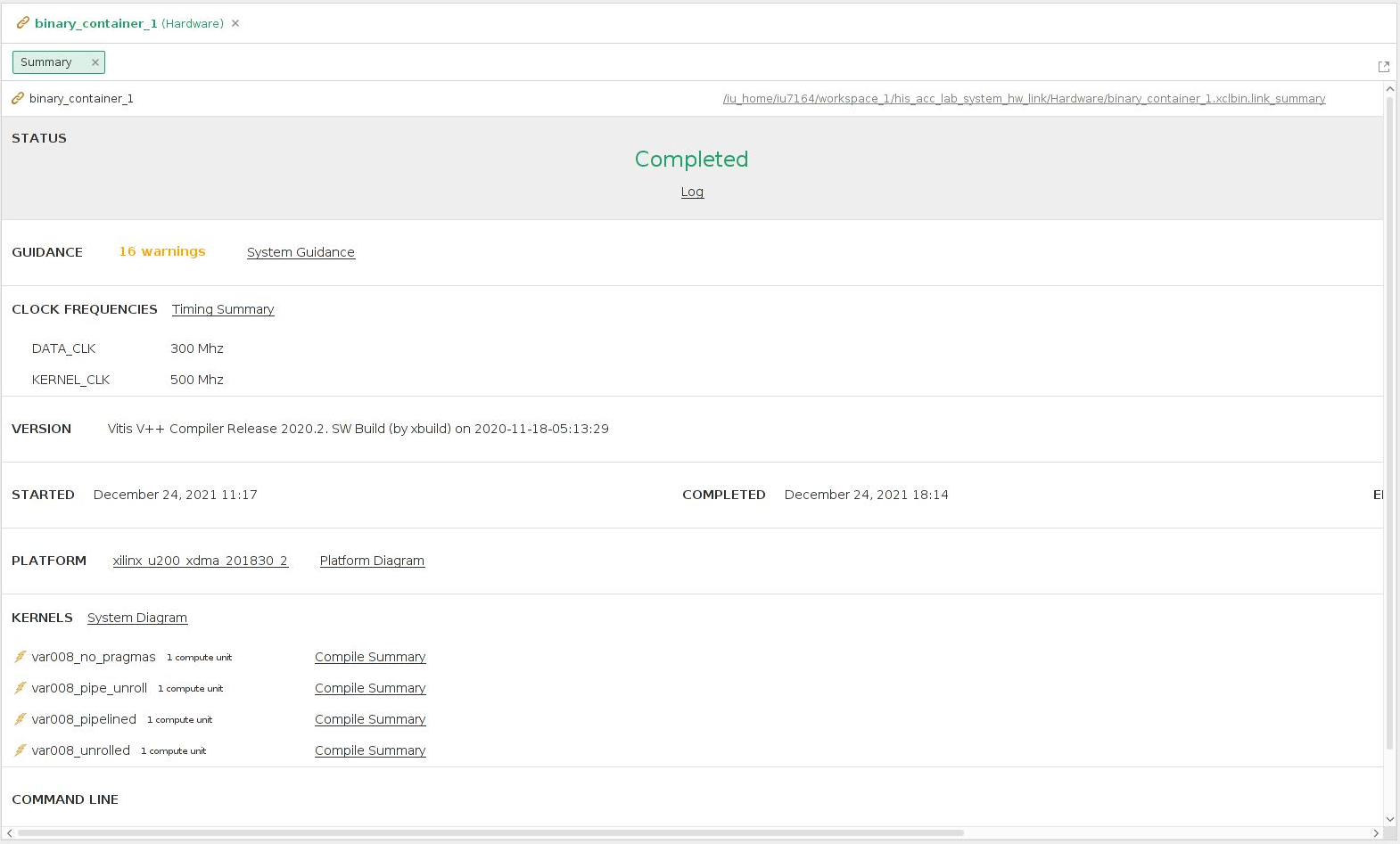


Рисунок 5. Содержимое вкладки Summary

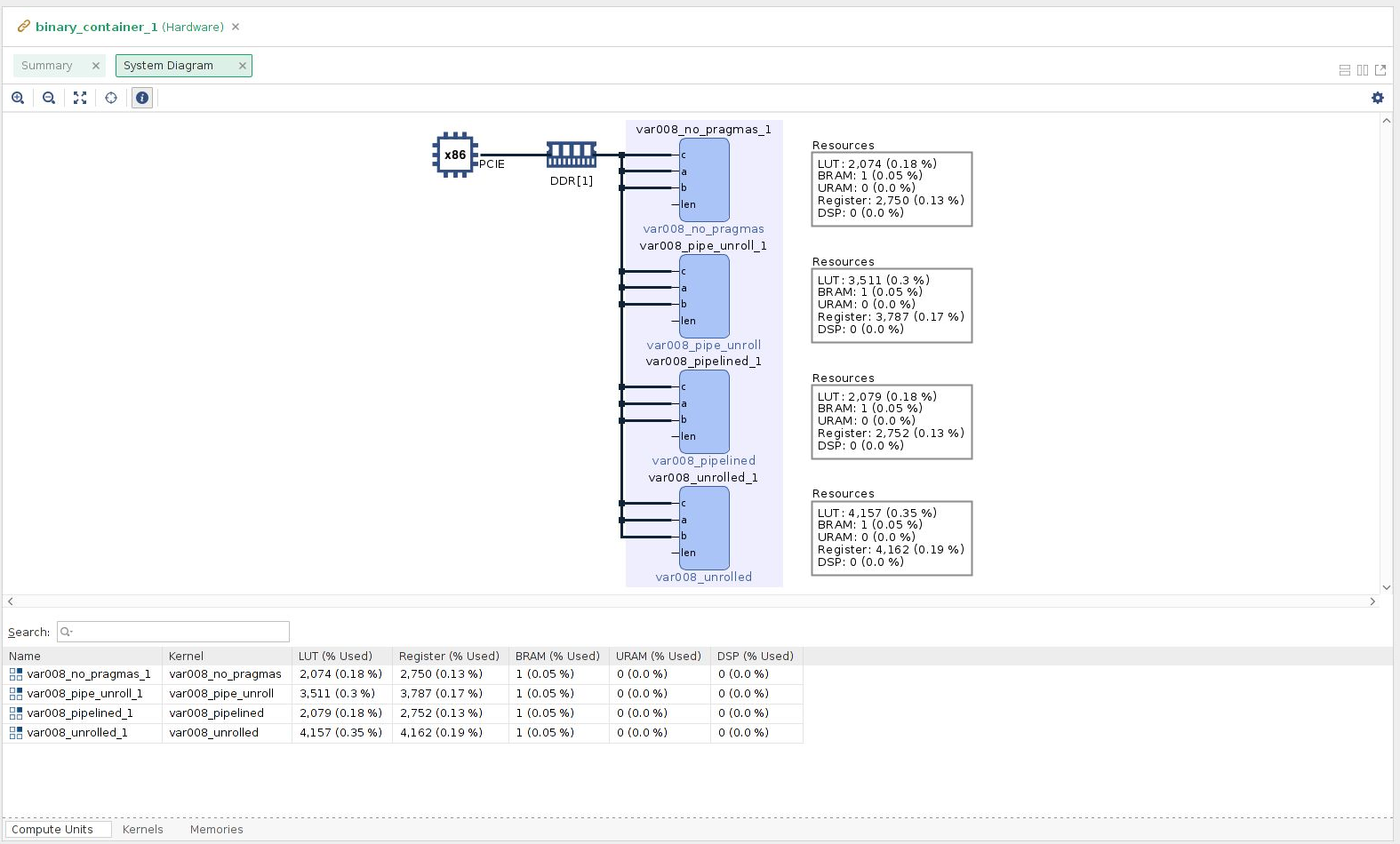


Рисунок 6. Содержимое вкладки System Diagram

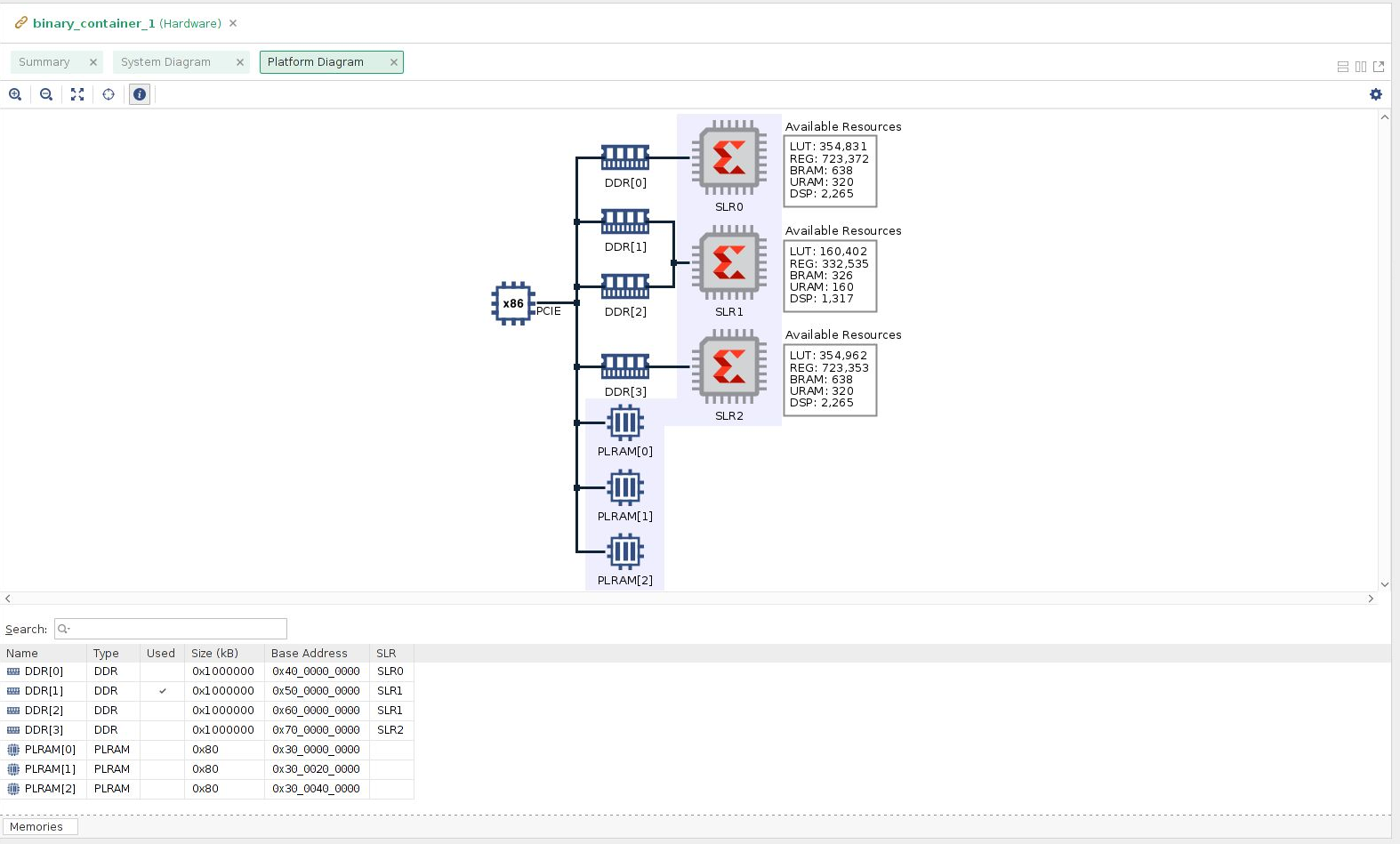


Рисунок 7. Содержимое вкладки Platform Diagram

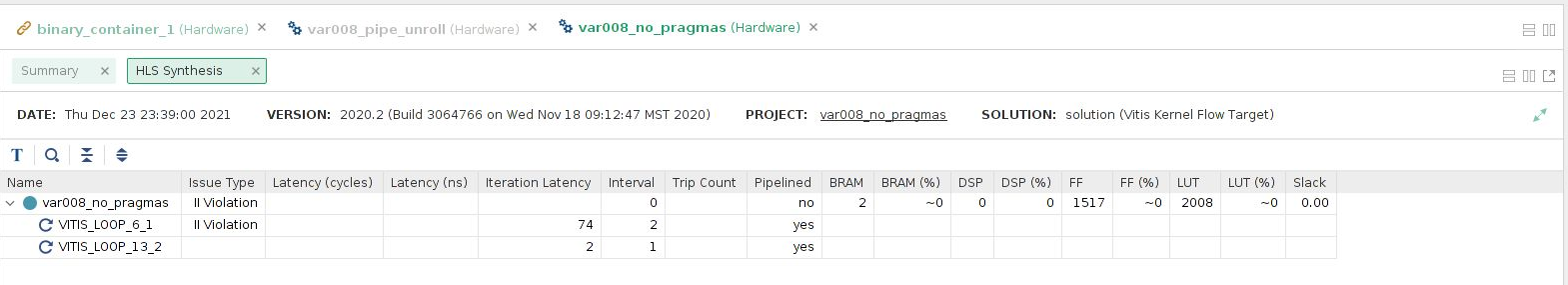


Рисунок 8. HLS (1)

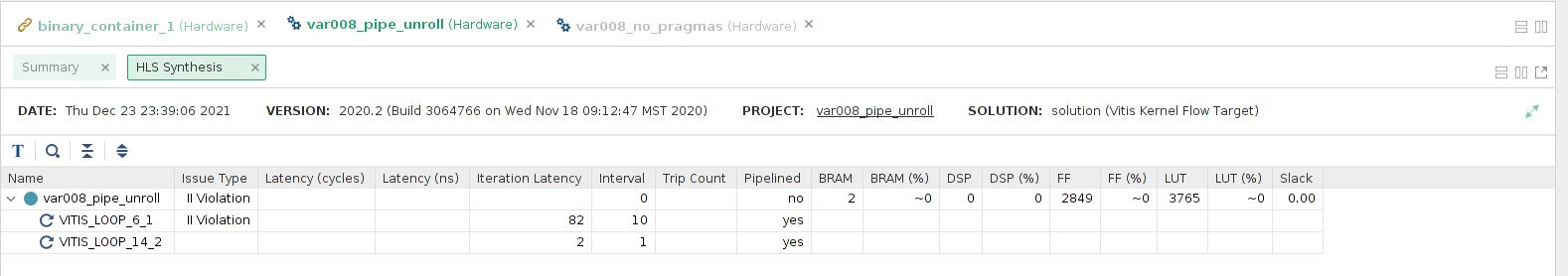


Рисунок 9. HLS (2)

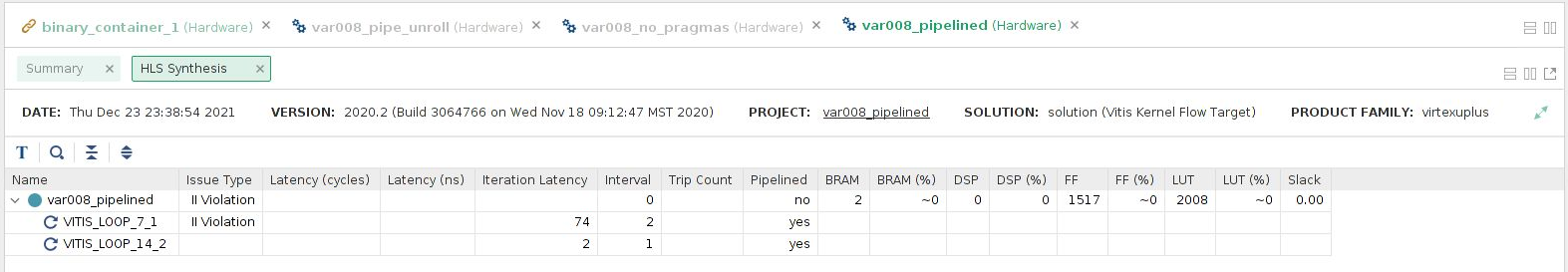


Рисунок 10. HLS (3)

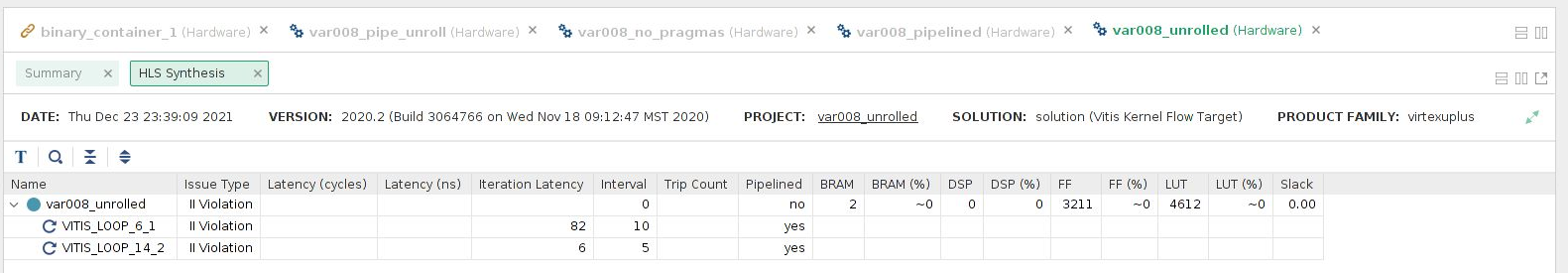


Рисунок 11. HLS (4)

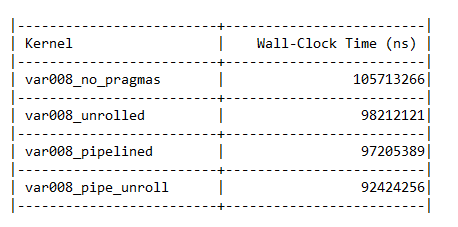


Рисунок 12. Результат работы программы

# Заключение

В ходе лабораторной работы были изучены архитектура гетерогенных вычислительных систем и технологии разработки ускорителей вычислений на базе ПЛИС фирмы Xilinx. Была выполнена генерация ядра ускорителя с последующим синтезом, сборкой и тестированием бинарного модуля ускорителя.

В результате сборки проекта было выяснено, что использование оптимизаций приводит к реальному повышению быстродействия работы программы. Однако следует отметить, что в режиме программной эмуляции выигрыш получился наиболее существенным (до 3-х раз), а в режимах аппаратной эмуляции и аппаратного исполнения ускорение осталось, но оно не настолько существенное (до 5-10%). Это можно объяснить тем, что возможно объем тестирования был недостаточным и небольшая выборка данных не позволяет получить наиболее точные результаты, также ввиду большой загруженности удаленного сервера и разного количества пользователей на нем, тестирование происходило не в равных условиях.