



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)»

---

## Отчёт

по лабораторной работе №5

Название «Разработка ускорителей вычислений средствами САПР  
высокоуровневого синтеза Xilinx Vitis HLS»

---

Дисциплина «Архитектура ЭВМ»

---

Студент ИУ7-55Б

---

(подпись, дата)

Бугаенко А.П.  
(Фамилия И.О.)

---

Преподаватель

---

(подпись, дата)

А.Ю. Попов  
(Фамилия И.О.)

---

Москва, 2022

## Содержание

Введение . . . . .	3
Практическая часть . . . . .	4
Результаты сборки и отладки в режиме Emulation SW . . . . .	6
Результаты сборки и отладки в режиме Emulation HW . . . . .	7
Результаты сборки и отладки в режиме Hardware . . . . .	9
Вывод . . . . .	12

## 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 уровня синтеза, выполнить его отладку.

Для достижения данной цели необходимо выполнить следующие задачи:

- 1) Разработать ускоритель вычислений по индивидуальному заданию;
- 2) Разработать код для тестирования ускорителя, реализовать ускоритель с помощью средств высоко уровня синтеза;
- 3) Выполнить отладку реализованного ускорителя.

## Практическая часть

Листинг 1 — Листинг неоптимизированного кода

```
1 extern "C" {
2 void var002_no_prragmas(int* c, const int* a, const int* b, const int len) {
3     for (int i = 0; i < len; i+=2) {
4         if (b[i] > a[i]) {
5             c[i] = b[i];
6         } else {
7             c[i]= a[i];
8         }
9     }
10    for (int i = 1; i < len; i+=2) {
11        if (b[i] < a[i]) {
12            c[i] = b[i];
13        } else {
14            c[i]= a[i];
15        }
16    }
17 }
18 }
```

Листинг 2 — Листинг кода с конвейерной организацией

```
1 extern "C" {
2 void var002_pipelined(int* c, const int* a, const int* b, const int len) {
3     for (int i = 0; i < len; i+=2) {
4 #pragma HLS PIPELINE
5         if (b[i] > a[i]) {
6             c[i] = b[i];
7         } else {
8             c[i]= a[i];
9         }
10    }
11    for (int i = 1; i < len; i+=2) {
12        if (b[i] < a[i]) {
13            c[i] = b[i];
14        } else {
15            c[i]= a[i];
16        }
17    }
18 }
19 }
```

Листинг 3 — Листинг кода с развёрткой

```
1 extern "C" {
```

```

2 void var002_unrolled(int* c, const int* a, const int* b, const int len) {
3     for (int i = 0; i < len; i+=2) {
4 #pragma HLS UNROLL factor = 2
5         if (b[i] > a[i]) {
6             c[i] = b[i];
7         } else {
8             c[i] = a[i];
9         }
10    }
11    for (int i = 1; i < len; i+=2) {
12 #pragma HLS UNROLL factor = 2
13        if (b[i] < a[i]) {
14            c[i] = b[i];
15        } else {
16            c[i] = a[i];
17        }
18    }
19 }
20 }

```

Листинг 4 — Листинг кода с развёрткой и конвейерной организацией

```

1 extern "C" {
2 void var002_pipe_unroll(int* c, const int* a, const int* b, const int len) {
3     for (int i = 0; i < len; i+=2) {
4 #pragma HLS PIPELINE
5 #pragma HLS UNROLL factor = 2
6         if (b[i] > a[i]) {
7             c[i] = b[i];
8         } else {
9             c[i] = a[i];
10        }
11    }
12    for (int i = 1; i < len; i+=2) {
13 #pragma HLS UNROLL factor = 2
14        if (b[i] < a[i]) {
15            c[i] = b[i];
16        } else {
17            c[i] = a[i];
18        }
19    }
20 }
21 }

```

## Результаты сборки и отладки в режиме Emulation SW

```
Console Vitis Log Guidance Progress
<terminated> (exit value: 0) SystemDebugger_hls_acc_lab_system_hls_acc_lab [OpenCL] /iu_home/iu7122/workspa
[Console output redirected to file:/iu_home/iu7122/workspace_lab_2/hls_acc_lab/Emulation
Found Platform
Platform Name: Xilinx
INFO: Reading /iu_home/iu7122/workspace_lab_2/hls_acc_lab_system/Emulation-SW/binary_cor
Loading: '/iu_home/iu7122/workspace_lab_2/hls_acc_lab_system/Emulation-SW/binary_contain
Trying to program device[0]: xilinx_u200_xdma_201830_2
Device[0]: program successful!
+-----+-----+
| Kernel | Wall-Clock Time (ns) |
+-----+-----+
| var002_no_pragmas | 4272545 |
+-----+-----+
| var002_unrolled | 4100841 |
+-----+-----+
| var002_pipelined | 713143 |
+-----+-----+
| var002_pipe_unroll | 1364804 |
+-----+-----+
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 0.1 — Результат работы программы

## Результаты сборки отладки в режиме Emulation HW

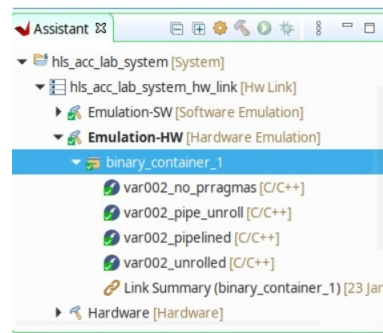


Рисунок 0.1 — Assistant View

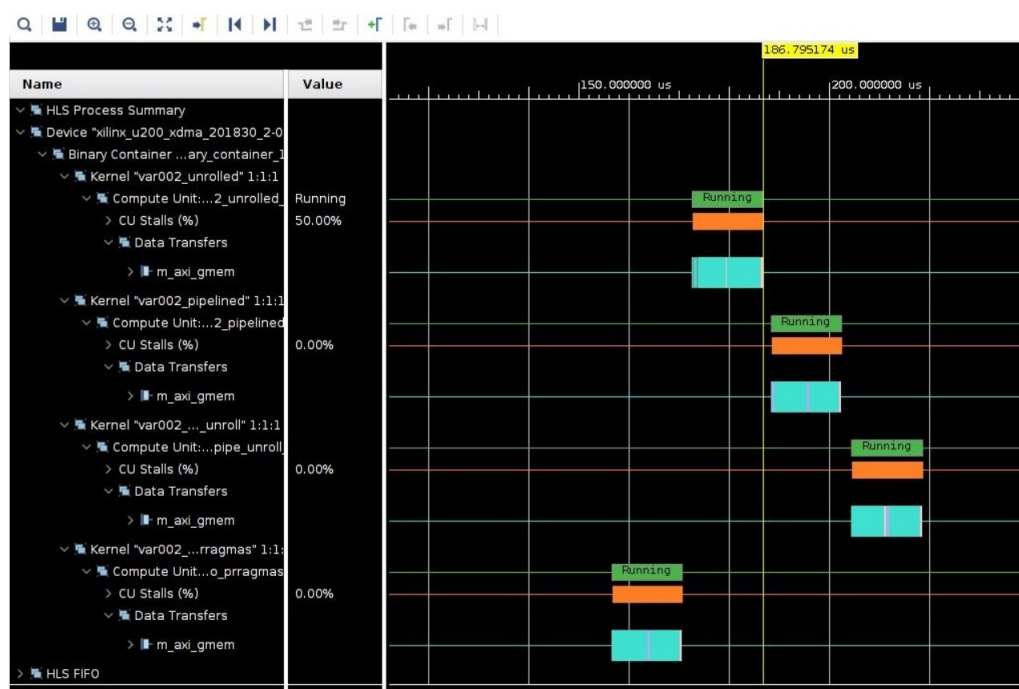


Рисунок 0.2 — Окно внутрисхемового отладчика Vivado

Console

Vitis Log

Guidance

Progress

SystemDebugger\_hls\_acc\_lab\_system\_hls\_acc\_lab [OpenCL] hls\_acc\_lab/Emulation-HW/hls\_acc\_lab (10/16/21, 8:03 PM)

```

Device[0]: program successful!
|-----|
| Kernel | Wall-Clock Time (ns) |
|-----|
| var002_no_prragmas | 99067774063 |
|-----|
INFO::[ Vitis-EM 22 ] [Time elapsed: 16 minute(s) 53 seconds, Emulation time: 0.180494 ms]
Data transfer between kernel(s) and global memory(s)
var002_no_prragmas_1:m_axi_gmem-DDR[1] RD = 8.000 KB WR = 4.000 KB
var002_pipe_unroll_1:m_axi_gmem-DDR[1] RD = 0.000 KB WR = 0.000 KB
var002_pipelined_1:m_axi_gmem-DDR[1] RD = 0.000 KB WR = 0.000 KB
var002_unrolled_1:m_axi_gmem-DDR[1] RD = 4.379 KB WR = 2.137 KB

| var002_unrolled | 96061189903 |
|-----|
| var002_pipelined | 99064311771 |
|-----|
| var002_pipe_unroll | 96073351537 |
|-----|
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.

```

Рисунок 0.3 — Результат работы программы



## Результаты сборки отладки в режиме Hardware

The screenshot shows the 'Summary' tab for a hardware debugging session. At the top, there's a tab labeled 'binary\_container\_1 (Hardware)' with a close button. Below it, a 'Summary' sub-tab is active. The main content area has a large green 'Completed' status with a 'Log' link underneath. Below this, there are sections for 'GUIDANCE' (with '2 Infos' and a 'System Guidance' link), 'CLOCK FREQUENCIES' (with a 'Timing Summary' link), and 'VERSION' (showing 'Vitis V++ Compiler Release 2020.2. SW Build (by xbuild) on 2020-11-18-05:13:29'). A timeline shows 'STARTED' on October 16, 2021 at 20:58 and 'COMPLETED' on October 17, 2021 at 13:54. The 'PLATFORM' section lists 'xilinx\_u200\_xdma\_201830\_2' and a 'Platform Diagram' link.

binary\_container\_1 (Hardware) ×

Summary ×

binary\_container\_1 [/iu\\_home/lu7122/workspace\\_lab\\_2/hls\\_acc](#)

**STATUS** Completed [Log](#)

**GUIDANCE** 2 Infos [System Guidance](#)

**CLOCK FREQUENCIES** [Timing Summary](#)

DATA_CLK	300 Mhz
KERNEL_CLK	500 Mhz

**VERSION** Vitis V++ Compiler Release 2020.2. SW Build (by xbuild) on 2020-11-18-05:13:29

**STARTED** October 16, 2021 20:58 **COMPLETED** October 17, 2021 13:54

**PLATFORM** [xilinx\\_u200\\_xdma\\_201830\\_2](#) [Platform Diagram](#)

Рисунок 0.1 — Содержимое вкладки Summary часть 1

This screenshot continues the 'Summary' tab content. It shows the 'VERSION' section again, followed by the 'STARTED' and 'COMPLETED' timeline. The 'PLATFORM' section is repeated. Below that, the 'KERNELS' section is expanded, showing a list of kernels with their compute units and links to their 'Compile Summary' pages. Finally, the 'COMMAND LINE' section is shown, displaying the full command used for the build.

binary\_container\_1 (Hardware) ×

Summary ×

**VERSION** Vitis V++ Compiler Release 2020.2. SW Build (by xbuild) on 2020-11-18-05:13:29

**STARTED** October 16, 2021 20:58 **COMPLETED** October 17, 2021 13:54

**PLATFORM** [xilinx\\_u200\\_xdma\\_201830\\_2](#) [Platform Diagram](#)

**KERNELS** [System Diagram](#)

var002_no_prragmas	1 compute unit	<a href="#">Compile Summary</a>
var002_pipe_unroll	1 compute unit	<a href="#">Compile Summary</a>
var002_pipelined	1 compute unit	<a href="#">Compile Summary</a>
var002_unrolled	1 compute unit	<a href="#">Compile Summary</a>

**COMMAND LINE**

```
--target hw  
--link  
--config binary_container_1-link.cfg  
-obinary_container_1.xclbin ../../hls_acc_lab_kernels/Hardware/build/var002_no_prragmas.xo ../../hls_acc_lab_kernels/Hardware/build/
```

Рисунок 0.2 — Содержимое вкладки Summary часть 2

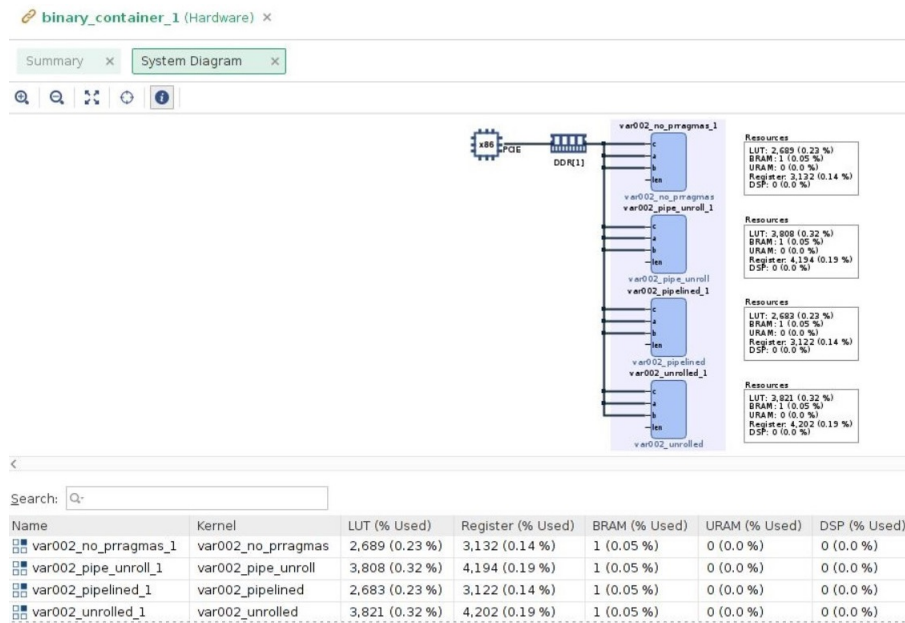


Рисунок 0.3 — Содержимое вкладки System Diagram

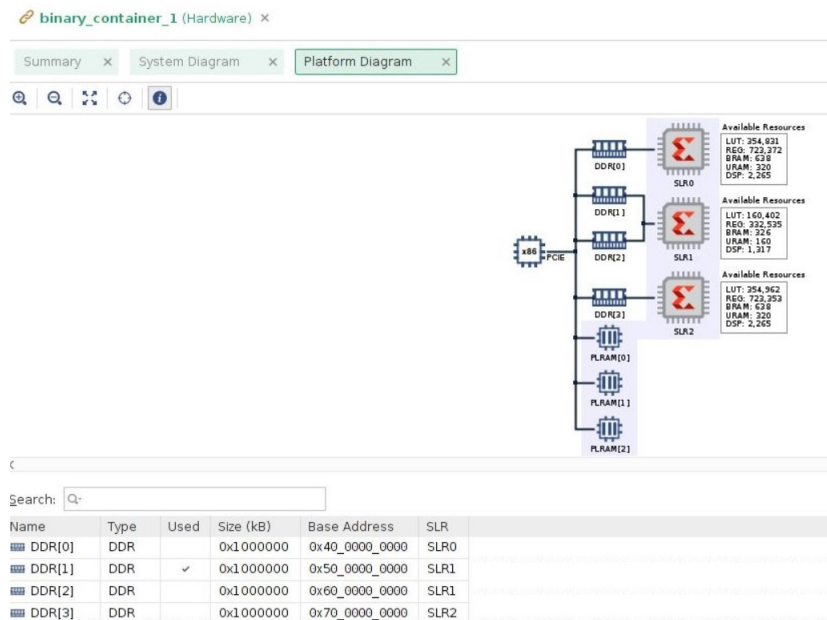


Рисунок 0.4 — Содержимое вкладки Platform Diagram

DATE: Sat Oct 16 20:53:15 2021    VERSION: 2020.2 (Build 3064766 on Wed Nov 18 09:12:47 MST 2020)    PROJECT: var002\_no\_prragmas

Name	Issue Type	Latency ...	Latency (ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	BRAM (%)	DSP	DSP (%)	FF	FF (%)	LUT	LUT (%)	Slack
var002_no_prragmas	II Violation						no	2	~0	0	0	2230	~0	2345	~0	0.00
VITIS_LOOP_3_1	II Violation			143	2		yes									
VITIS_LOOP_10_2	II Violation			143	2		yes									

Рисунок 0.5 — HLS no params

DATE: Sat Oct 16 20:53:37 2021    VERSION: 2020.2 (Build 3064766 on Wed Nov 18 09:12:47 MST 2020)    PROJECT: var002\_unrolled    SOLUTION: solution (Vitis Kernel Flow Target)

Name	Issue Type	Latency (cycles)	Latency (ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	BRAM (%)	DSP	DSP (%)	FF	FF (%)	LUT	LUT (%)	Slack
var002_unrolled	II Violation						no	2	~0	0	0	3446	~0	3415	~0	0.00
VITIS_LOOP_3_1	II Violation			285	4		yes									
VITIS_LOOP_11_2	II Violation			285	4		yes									

Рисунок 0.6 — HLS unroll

DATE: Sat Oct 16 20:53:16 2021    VERSION: 2020.2 (Build 3064766 on Wed Nov 18 09:12:47 MST 2020)    PROJECT: var002\_pipelined    SOLUTION: solution (Vitis Kernel Flow Target)

Name	Issue Type	Latency (cycles)	Latency (ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	BRAM (%)	DSP	DSP (%)	FF	FF (%)	LUT	LUT (%)	Slack
var002_pipelined	II Violation						no	2	~0	0	0	2230	~0	2345	~0	0.00
VITIS_LOOP_3_1	II Violation			143	2		yes									
VITIS_LOOP_11_2	II Violation			143	2		yes									

Рисунок 0.7 — HLS pipe

DATE: Sat Oct 16 20:53:38 2021    VERSION: 2020.2 (Build 3064766 on Wed Nov 18 09:12:47 MST 2020)    PROJECT: var002\_pipe\_unroll    SOLUTION: solution (Vitis Kernel Flow Target)

Name	Issue Type	Latency (cycles)	Latency (ns)	Iteration Latency	Interval	Trip Count	Pipelined	BRAM	BRAM (%)	DSP	DSP (%)	FF	FF (%)	LUT	LUT (%)	Slack
var002_pipe_unroll	II Violation						no	2	~0	0	0	3446	~0	3415	~0	0.00
VITIS_LOOP_3_1	II Violation			285	4		yes									
VITIS_LOOP_12_2	II Violation			285	4		yes									

Рисунок 0.8 — HLS pipe unroll

```

Console Progress Vitis Serial Terminal Executables Debug Shell Vitis Log Problems
<terminated> (exit value: 0) SystemDebugger_hls_acc_lab_system_hls_acc_lab [OpenCL] /iu_home/iu7122/workspa
Trying to program device[0]: xilinx_u200_xdma_201830_2
Device[0]: program successful!
|-----+-----|
| Kernel | Wall-Clock Time (ns) |
|-----+-----|
| var002_no_prragmas | 165128757 |
|-----+-----|
| var002_unrolled | 76737205 |
|-----+-----|
| var002_pipelined | 163310756 |
|-----+-----|
| var002_pipe_unroll | 77858502 |
|-----+-----|
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 0.9 — Результаты работы программы

## Вывод

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.