



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

ОТЧЕТ

по лабораторной работе № 5

Название: Разработка ускорителей вычислений средствами САПР
высокоуровневого синтеза XILINX VITIS HLS

Дисциплина: Архитектура ЭВМ

Студент

ИУ7И-56Б
(Группа)

(Подпись, дата)

Андрич К
(И.О. Фамилия)

Преподаватель

(Подпись, дата)

А.Ю. Попов
(И.О. Фамилия)

Москва, 2022

Цель работы

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-уровневого синтеза, выполнить его отладку.

Ход работы

Неоптимизированный цикл:

```
extern "C" {
void var010_no_pragmas(int* c, const int* a, const int* b, const int len)
{
    int maxA = a[len-1];
    for (int i = len-1; i >=0 ; i--) {
        if (maxA < a[i]) {
            maxA = a[i];
            c[i] = maxA;
        } else {
            c[i] = b[i];
        }
    }
}
}
```

Конвейерная организация цикла:

```
extern "C" {
void var010_pipelined(int* c, const int* a, const int* b, const int len)
{
    int maxA = a[len-1];
    for (int i = len-1; i >=0 ; i--) {
        #pragma HLS PIPELINE
        if (maxA < a[i]) {
            maxA = a[i];
            c[i] = maxA;
        } else {
            c[i] = b[i];
        }
    }
}
}
```

Частично развёрнутый цикл:

```
extern "C" {
void var010_unrolled(int* c, const int* a, const int* b, const
int len) {
    int maxA = a[len-1];
    for (int i = len-1; i >=0 ; i--) {
        #pragma HLS UNROLL factor=2
        if (maxA < a[i]) {
            maxA = a[i];
            c[i] = maxA;
        } else {
            c[i] = b[i];
        }
    }
}
}
```

Конвейерный и частично развёрнутый цикл:

```
extern "C" {
void var010_pipe_unroll(int* c, const int* a, const int* b,
const int len) {
    int maxA = a[len-1];
    #pragma HLS DATAFLOW
    for (int i = len-1; i >=0 ; i--) {
        if (maxA < a[i]) {
            maxA = a[i];
            c[i] = maxA;
        } else {
            c[i] = b[i];
        }
    }
}
}
```

Emulation SW

Результаты работы:

```
Device[0]: program successful!
-----+-----
| Kernel                | Wall-Clock Time (ns) |
|-----+-----|
| var010_no_pragmas     | 6969276               |
|-----+-----|
| var010_unrolled       | 1309564               |
|-----+-----|
| var010_pipelined      | 1751307               |
|-----+-----|
| var010_pipe_unroll    | 10264029              |
|-----+-----|
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.
```

Emulation HW