Министерство науки и высшего образования РФ

Федеральное государственное бюджетное

образовательное учреждение высшего образования

ВЯТСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет автоматики и вычислительной техники

Кафедра радиоэлектронных средств

Отчёт по лабораторной работе №4

**Конвейерная обработка**

дисциплина «Проектирование устройств и систем на цифровых сигнальных процессорах»

|  |  |  |
| --- | --- | --- |
| Выполнил студент гр.ИНБс-5301: |  | /И.И. Золотаева/ |
| Проверил: старший преподаватель кафедры систем автоматизации управления |  | /М.А. Земцов/ |

Киров 2023

**Цель:** получить навыки ручного написания кода конвейерной обработки.

**Ход работы**

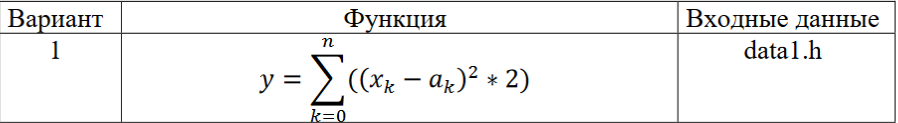


Рисунок 1 – Задание к лабораторной работе

**1 Оптимизация программы**

До оптимизации с помощью конвейера, программа выполнялась за 15.124 мкс, а после оптимизации время выполнения составило 11.932 мкс. Также в качестве оптимизации домножение на 2 внутри функции было вынесено за знак суммы.

Исходный код, оптимизированный:

.def \_dotp

\_dotp:

MVK .S1 256, A1

MVK .S2 2,B6

ZERO .L1 A7

loop:

LDH .D1 \*A4++, A2

|| LDH .D2 \*B4++, B2

NOP 4

|| SUB .L2x A2,B2,B7

NOP 2

|| MPY .M2x B7,B7,B5

NOP

|| ADD .L1 A7, B5, A7

NOP

|| SUB .S1 A1, 1, A1

[A1] B .S2 loop

NOP 5

ADD .L1 A7,A7,A7

MV .S1 A7, A4

B .S2 B3

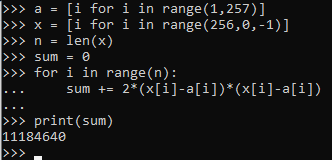


Рисунок 2 – Скрипт на Python для проверки правильности расчётов



Рисунок 3 – Результат работы программы

Результаты совпадают.

Выводы:

Конвейерная обработка - это процесс обработки данных, при котором каждый шаг обработки выполняется последовательно и параллельно. Написание кода конвейерной обработки имеет свои особенности:

* Разделение шагов. Каждый шаг обработки должен быть разделен на отдельную функцию (или метод), которая будет выполнять конкретную обработку и возвращать результат. Это позволит легко распараллеливать выполнение шагов.
* Организация потоков. Код конвейерной обработки должен быть организован таким образом, чтобы каждый шаг выполнялся в отдельном потоке. Это позволит ускорить выполнение процесса обработки данных.
* Обработка ошибок. Каждый шаг обработки должен быть обернут в блок try-catch, чтобы обрабатывать возможные ошибки в конвейерной обработке.
* Оптимизация. Код конвейерной обработки должен быть оптимизирован для максимальной производительности. Например, можно использовать специальные библиотеки для работы с параллельными потоками или использовать асинхронное программирование.
* Тестирование. Наконец, код конвейерной обработки должен быть тестирован для проверки работоспособности и эффективности. Необходимо тестировать каждый шаг обработки, а также проверять корректность обработки ошибок и правильную работу с потоками.