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 №2

**Обработка изображений**

дисциплина «Проектирование устройств и систем на цифровых сигнальных процессорах»

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

Киров 2023

**Цель:** изучение дискретного косинусного преобразования изображения; получение навыков обработки изображений на DSK.

**Ход работы**

Программный код приведён в <https://github.com/Argento-prg/SignalProcessors>/tree/lab2/code

**1 Формирование файла-пробника**

В первую очередь необходимо преобразовать полутоновое изображение в удобный для обработки формат. Реализовано это через скрипт Matlab.



Рисунок 1 – Исходное изображение

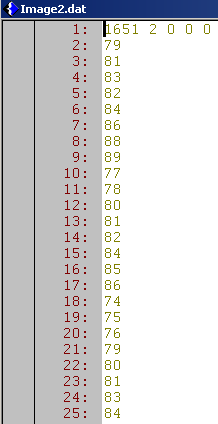


Рисунок 2 – Данные для обработки

**2 Обработка данных**

Далее, необходимо преобразовать входные данные по алгоритму дискретного косинусного преобразования в частотную форму. Для загрузки файла из которого данные будут читаться в память, необходимо выставить Probe Point в месте, до обработки данных.

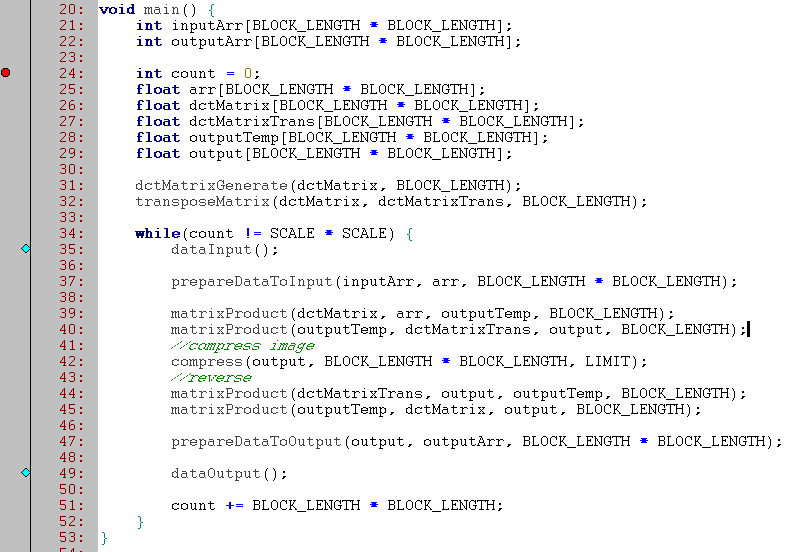


Рисунок 3 – Места расположения Brake Point и Probe Point

Brake Point в 24 строке служит для паузы в выполнении программы, в ходе которой можно будет установить адреса входного inputArr и выходного outputArr массивов.

После установки точек, показанных на рисунке 3, нужно пересобрать проект (Rebuild All), чтобы в проекте эти точки стали доступны. Потом заходим в пункт меню File IO (рисунок 4).

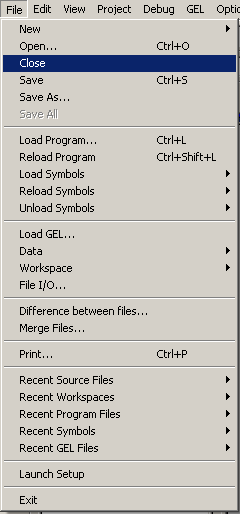


Рисунок 4 – Пункт меню File IO

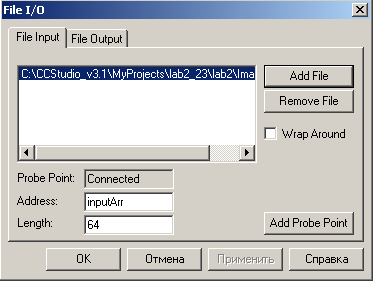


Рисунок 5 – Настройки для входного файла

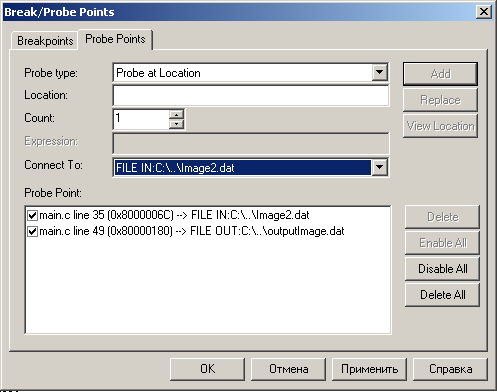


Рисунок 6 – Настройки для входного файла

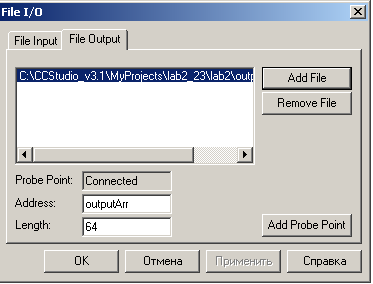


Рисунок 7 – Настройки для выходного файла

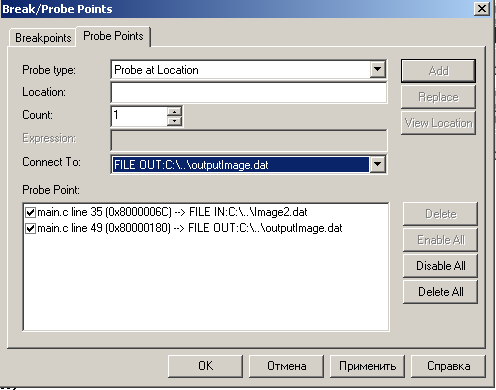


Рисунок 8 – Настройки для выходного файла

После данных действий, можно продолжать выполнение программы с точки останова, о которой говорилось ранее. Функции dataInput и dataOutput служат заглушками, на которых ставятся Probe Point.

**3 Результаты работы**

В результате получились следующие выходные данные (рисунок 9).

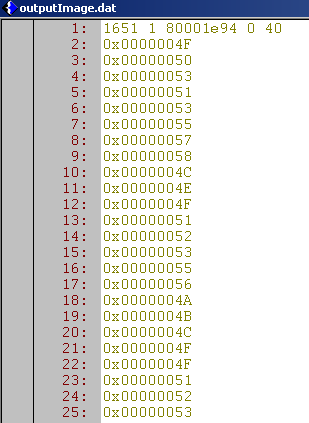


Рисунок 9 – Пример выходных данных

Потом, используя скрипт Python, получаем следующие восстановленные изображения.



Рисунок 10 – Сжатие до 13 значащих элементов из 64



Рисунок 11 – Сжатие до 40 значащих элементов из 64

**Выводы:**

Проведение лабораторной работы №2 по обработке изображений позволяет увидеть, каким образом можно использовать ДКП для обработки изображений, а именно сжатия изображения. В процессе выполнения работы мы изучаем не только математические основы ДКП, но также приобретаем практические навыки обработки изображений на цифровой сигнальной обработке (ЦСО) платформе DSK.

В результате выполнения лабораторной работы можно сделать следующие выводы:

ДКП позволяет сжимать изображения с минимальными потерями качества. Однако, при высокой степени сжатия, даже небольшие изменения в оригинальном изображении могут привести к значительным искажениям после применения ДКП.

Проведение обработки изображений на ЦСО платформе DSK позволяет эффективно применять ДКП для обработки изображений в реальном времени.

Для получения качественного сжатия изображения с помощью ДКП важно правильно выбрать параметры преобразования, такие как размер блока, используемый для ДКП, и пороговое значение, используемое для отбрасывания слабых частот. ДКП используется не только для сжатия изображений, но и для других задач, таких как распознавание образов, компрессия видео и звука, анализ временных рядов и т.д. Так же был получен навык создания входных и выходных файлов в CCS/