

## ИСПОЛЬЗОВАНИЕ C++ В ОБРАБОТКЕ МУЛЬТИМЕДИА ДАННЫХ НА ПРИМЕРЕ ДЕКОДИРОВАНИЯ ВИДЕО КОДЕКА AV1

Малашенко С.В., студент гр.050701

Белорусский государственный университет информатики и радиоэлектроники  
г. Минск, Республика Беларусь

Порхун М.И. – магистр техн. наук

**Аннотация.** В работе рассмотрено применение языка программирования C++ в сфере мультимедиа на примере реализации аппаратного декодирования видео кодека AV1. Основой алгоритма аппаратного декодирования являются 3 основных этапа: парсинг битстрима, обработка полученных данных и отправка данных и команд непосредственно в аппаратный декодер. Для взаимодействия с аппаратным декодером использован VA-API и Intel media driver (media SDK).

**Ключевые слова.** Аппаратное декодирование, видео, декодирование, кодек, AV1, VA-API, C++.

Сфера мультимедиа является одной из ключевых в жизни современного общества — она является основным источником получения информации о событиях окружающей действительности. В данной сфере существуют 4 основные задачи: хранение, обработка, передача и презентация медиаданных конечному потребителю. Для решения первых 2 задач были разработаны кодеки — стандарты кодирования и декодирования медиаданных, позволяющие снизить объем размер файлов и тем самым ускорить их передачу по сети. Проблема использования кодеков заключается в том, что после получения медиаданных их необходимо декодировать и только потом можно презентовать конечному потребителю. В данном вопросе производительность является ключевым параметром, задающим требования к разрабатываемому ПО. Как следствие в сфере декодеров и энкодеров доминируют языки с высокой производительностью: С и C++, возможно также применение языка Rust, но до стандарта индустрии ему крайне далеко. Ввиду поддержки языком C++ объектно ориентированной парадигмы, он является наиболее удобным для разработки крупных проектов с высокой производительностью и без потерь читаемости и поддерживаемости.

Хорошим практическим примером такого применения языка C++ является реализация системы аппаратного ускорения декодирования кодека AV1. Данная задача является весьма комплексной и ее следует рассматривать как последовательность взаимосвязанных этапов.

Общая схема выполнения данных этапов приведена на рисунке 1.

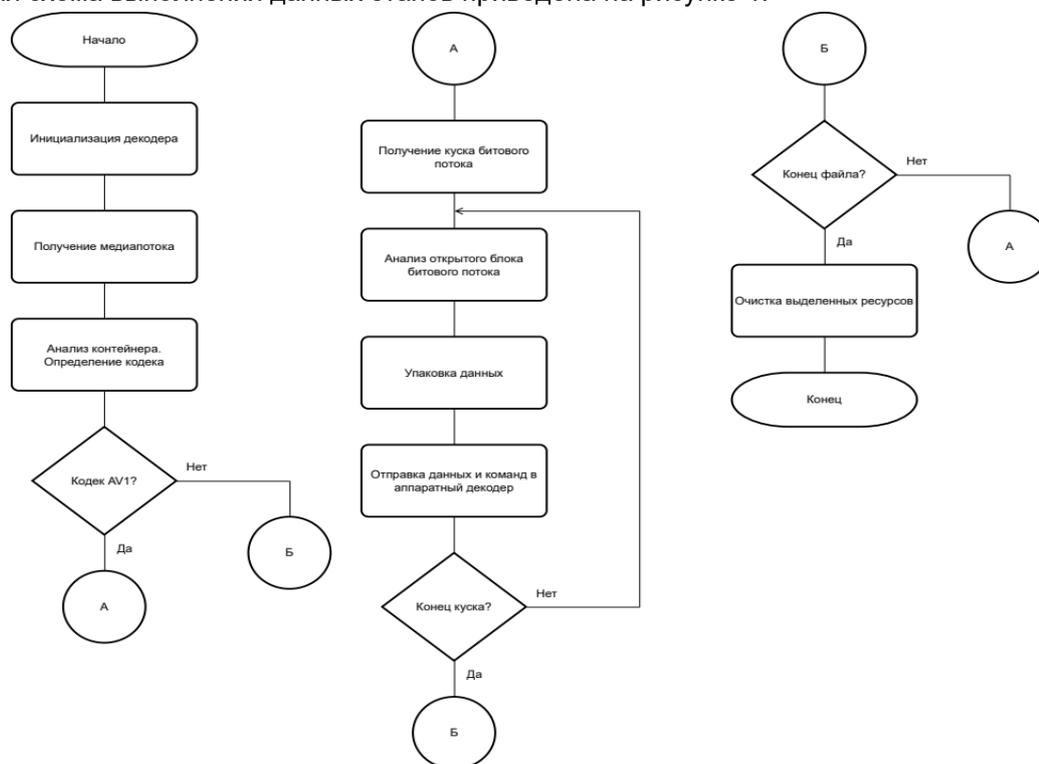


Рисунок 1 – Общая схема аппаратного декодирования видео кодека AV1

Реализацию системы следует начать с разработки парсера битстрима и различных его элементов. Ключевой задачей является эффективная обработка данных без байтового выравнивания

в соответствии с описанием из спецификации[1]. Для этого был использован буфер, превышающий размер наибольшего из элементов битстрима, применяемый в качестве скользящего окна обработки в сочетании с битовыми сдвигами и масками для изоляции и извлечения релевантных данных.

После решения данной задачи становится возможным анализ открытых блоков битового потока кодека AV1 — синтаксически независимых элементов битового потока высшего уровня. Данные элементы содержат в себе часть информации необходимой для декодирования кадра: контекст последовательности — данные наивысшего уровня общие для нескольких последовательных кадров; контекст кадра — данные специфичные для каждого отдельного кадра и кодированные данные слайсов кадра. Алгоритм анализа приведен на рисунке 2.

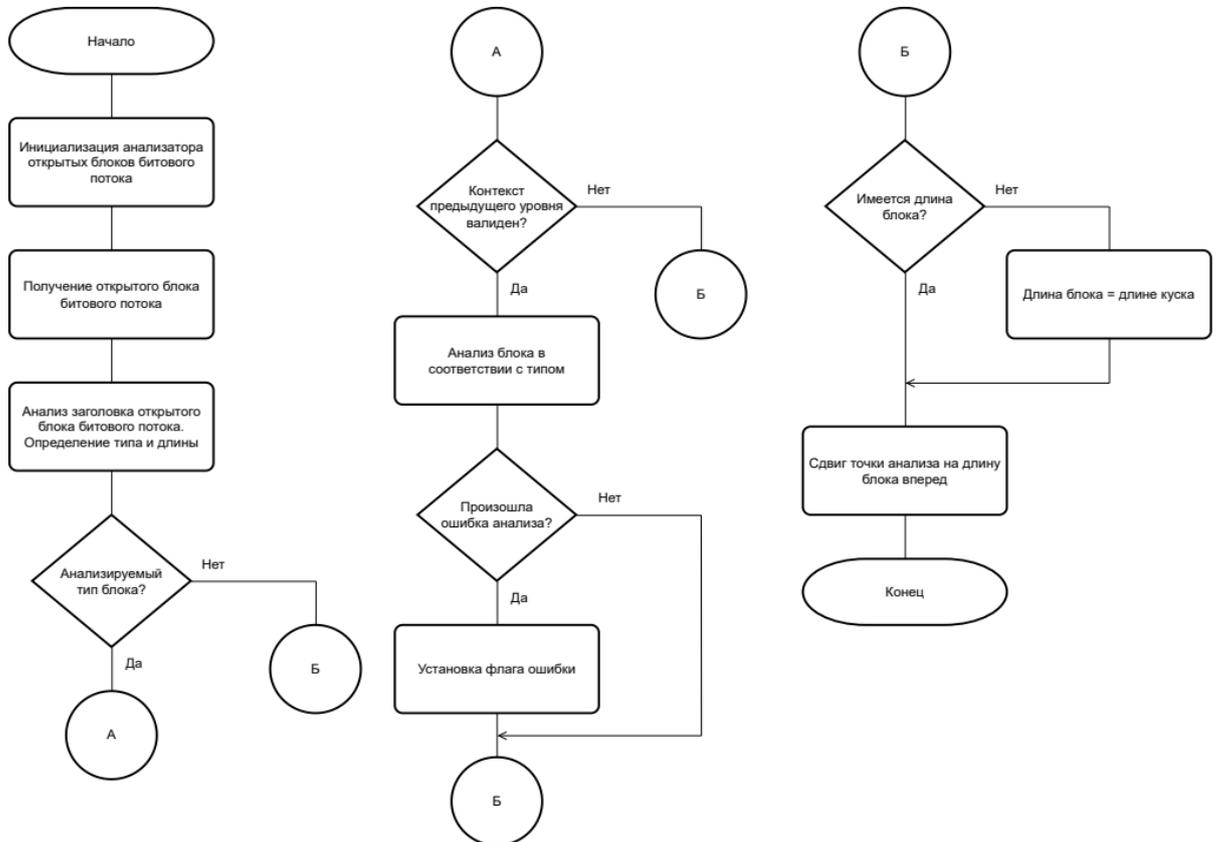


Рисунок 2 – Общая схема анализа открытых блоков битового потока видео кодека AV1

Следующим этапом является упаковка данных в соответствии с форматом, ожидаемым аппаратным декодером. Данный формат задается структурами, входящими в состав VA-API — программного интерфейса аппаратных декодеров видео на платформах Intel[2].

После того, как данные упакованы и готовы к отправке в декодер можно переходить к следующим шагам: передаче контекста кадра, а также слайсов в декодер. Слайс в данном контексте представляет собой кодированную информацию о группе плиток составляющих часть кадра или целый кадр в случае видео с низким разрешением. Далее необходимо передать команду, отвечающую за начало декодирования. Алгоритм управления аппаратным декодером и передачи данных в него представлен на рисунке 3.

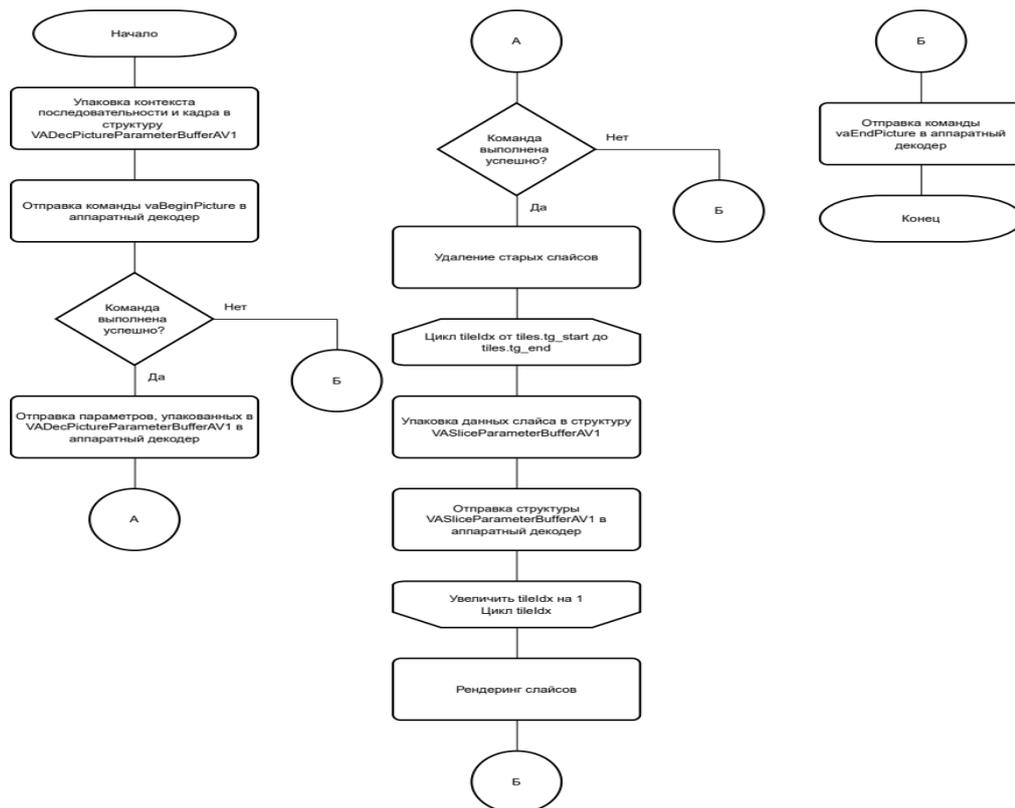


Рисунок 3 – Общая схема управления аппаратным декодером на основе VA-API

При успешном декодировании кадра к на заданном сюрфейсе находится валидный декодированный кадр в формате NV12. Полученный кадр пригоден для рендеринга и показа или постобработки.

Разработанная система успешно проходит тесты из официального набора тестовых данных для программного декодера dav1d. В процессе тестирования были выявлены 3 ошибочные контрольные суммы md5, вследствие чего значения данных контрольных сумм были заменены на полученные с использованием аппаратного декодера. Результаты прохождения тестов разработанным декодером представлены на рисунке 4.

```
[-----] Global test environment tear-down
[=====] 308 tests from 1 test suite ran. (9411 ms total)
[ PASSED ] 304 tests.
[ SKIPPED ] 3 tests, listed below:
[ SKIPPED ] VideoPipelineIntegrationTests/VideoPipelineAV1SampleTestSuite.VideoPipelineAV1SampleTestSuite/293
[ SKIPPED ] VideoPipelineIntegrationTests/VideoPipelineAV1SampleTestSuite.VideoPipelineAV1SampleTestSuite/295
[ SKIPPED ] VideoPipelineIntegrationTests/VideoPipelineAV1SampleTestSuite.VideoPipelineAV1SampleTestSuite/296
[ FAILED ] 1 test, listed below:
[ FAILED ] VideoPipelineIntegrationTests/VideoPipelineAV1SampleTestSuite.VideoPipelineAV1SampleTestSuite/0, where
GetParam() = (0x557b8409c207 pointing to "00000000.ivf", 0x557b8409aee0 pointing to "0b31f7ae90dfa22cfe0f2a1ad97c62
0", 2)

1 FAILED TEST
```

Рисунок 4 – Результаты прохождения тестов разработанным декодером

По результатам бенчмарков скорость декодирования при использовании аппаратного декодера во 30 раз превышает скорость программного декодирования в случае 8к видео.

**Список использованных источников:**

1. AV1 Bitstream & Decoding Process Specification / Peter de Rivaz, Jack Haughton // Argon Design Ltd, 2019 – P. 1–647;
2. VA-API Core API [электронный ресурс]. – Режим доступа : [https://intel.github.io/libva/group\\_api\\_core.html](https://intel.github.io/libva/group_api_core.html). - Дата доступа: 17.04.2024.

## **USAGE OF C++ IN MULTIMEDIA DATA PROCESSING BASED ON THE EXAMPLE OF AV1 VIDEO CODEC DECODING**

Malashenko S.V.

*Belarusian State University of Informatics and Radioelectronics,  
Minsk, Republic of Belarus*

*Porhun M.I. – Master of Engineering Science*

**Annotation.** The paper examines the use of the C++ programming language in the field of multimedia using the example of implementing hardware decoding of the AV1 video codec. The basis of the hardware decoding algorithm consists of 3 main stages: parsing the bitstream, processing the received data and sending data and commands directly to the hardware decoder. VA-API and Intel media driver (media SDK) are used to interact with the hardware decoder.

**Keywords.** Hardware decoding, video, decoding, codec, AV1, VA-API, C++.