МИНОБРНАУКИ РОССИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Национальный исследовательский университет  
«Московский институт электронной техники»  
Институт микроприборов и систем управления

Степанов Павел Александрович  
АО «НТЦ ЭЛИНС», Отдел 72

**Видеокодирование на ПЛИС**

Магистерская работа по направлению   
09.04.01 «Информатика и вычислительная техника»

|  |  |  |
| --- | --- | --- |
| Студент |  | Степанов П.А. |
| Руководитель, |  |  |
| кандидат технических наук |  | Ухандеев В.И. |

Аннотация

Annotation

ОГЛАВЛЕНИЕ

[1 Список аббревиатур 4](#_Toc95569492)

[2 Введение 5](#_Toc95569493)

[3 Основная часть 8](#_Toc95569494)

[3.1 Почему и зачем 8](#_Toc95569495)

[3.2 Как 14](#_Toc95569496)

[3.3 Как это делаю я 21](#_Toc95569497)

[3.4 Что у меня получилось 21](#_Toc95569498)

[4 Заключение 22](#_Toc95569499)

[5 Список использованных источников 23](#_Toc95569500)

[6 Приложения 25](#_Toc95569501)

# Список аббревиатур

# Введение

**Актуальность.**

Видеокодирование является неотъемлемой частью хранения и передачи цифрового видео, поскольку несжатое видео занимает огромное количество места, причём затраты по памяти порядка , где – наименьшее количество пикселей по вертикали или по горизонтали.

В настоящее время существует множество стандартов видеокодеков, предоставляющих возможность сжатия (кодирования) и восстановления (декодирования) видео, основанных на избыточности видео, то есть на устранении той информации, которая не влияет на субъективную оценку качества и восприятия информации человеком. Основным недостатком существующих стандартов является то, что они предоставляют информацию о реализации только декодеров на процессорной системе, с последовательным выполнением команд по восстановлению видео. Стандарты не рассчитаны на  
real-time кодирование и декодирование и требуют больших вычислительных мощностей, из-за чего невозможна реализация на микропроцессоре во страиваемой системе.

В диссертационной работе представлена адаптация стандарта декодера H.264 под ПЛИС и реализация кодера на ПЛИС совместимого с декодером H.264 с учётом требований real-time обработки входного видеопотока.

**Объектом исследования** являются системы обработки видео и видеопотоков.

**Предметом исследования** являются методы и алгоритмы кодирования и декодирования видео, оптимизации и обработки информации в системах на ПЛИС.

**Цель диссертационной работы:** разработка и реализация алгоритма кодирования и декодирования видеопотока в real-time на ПЛИС.

**Задачи диссертации:**

1. Анализ областей применения видеокодирования;
2. Изучение алгоритмов сжатия видео;
3. Анализ существующих стандартов видеокодеков;
4. Реализация алгоритма декодера по стандарту H.264 и совместимого кодера на ПЛИС;
5. Экспериментальная проверка разработанного устройства.

**Методы исследований.**

Анализ существующих стандартов видеокодеков. Анализ процессорных реализаций видеокодеков с открытым кодом. Моделирование в среде MATLAB. Экспериментальные исследования разработанных алгоритмов и методов.

**Новые научные результаты:**

1. Разработан алгоритм кодирования видео на ПЛИС совместимый со стандартом декодирования H.264;
2. Адаптирован под ПЛИС стандарт процессорной реализации декодирования видео H.264.

**Достоверность полученных результатов, выводов и рекомендаций.**

**Практическая значимость работы:**

1. Результаты работы могут быть использованы для реализации встраиваемой системы захвата видеопотока с видеокамеры с последующей отправкой сжатого видео по радиоканалу;
2. Результаты работы могут быть использованы для реализации встраиваемой системы захвата видеопотока с видеокарты сервера и передачи сжатого видео по интерфейсу Ethernet для предоставления терминального доступа;
3. Результаты работы могут быть использованы для реализации встраиваемой системы real-time восстановления сжатого видеопотока, принятого по радиоканалу или Ethernet.

**Личный вклад автора.**

Автором была проделана работа по анализу существующих стандартов и решений, моделированию алгоритмов сжатия, обсуждению структуры и алгоритмов устройства, реализации части функциональных блоков на языке описания аппаратуры VHDL.

**Положения и выводы, выносимые на защиту:**

1. Адаптированный к реализации на ПЛИС алгоритм декодирования видео по стандарту H.264;
2. Алгоритм кодирования видео на ПЛИС совместимый со стандартом декодирования H.264;
3. Результаты экспериментальной оценки метрик кодера при разных параметрах.

**~~Публикации автора по теме диссертации~~**

**Структура и объем работы.**

Диссертация состоит из введения, четырёх глав, заключения и списка литературы из 9 наименований. Объём диссертации составляет страниц, включая х рисунков, х таблиц ~~и 1 приложение~~.

**Краткое содержание диссертации.**

В диссертационной работе рассматривается почему и зачем необходимо видеокодирование, области применения видеокодирования, основные алгоритмы сжатия видео, существующие стандарты видеокодеков. Так же в работе описаны алгоритмы и методы, адаптированные с процессорной реализации под ПЛИС с учётом real-time обработки. Приведены замеры метрик получившегося видеокодека.

**Заключение**

Разработаны алгоритмы кодера и декодера на основе стандарта видеокодека H.264, которые позволяют сжимать и восстанавливать видеопоток в real-time.

Алгоритмы реализованы и протестированы на ПЛИС Xilinx Kintex-7 xc7k160t. Разработанное устройство может обрабатывать видео с разрешением 1920 на 1080 пикселей и частотой кадров равной 60 кадров/с.

**~~Приложения~~**

# Основная часть

## Почему и зачем

Человечество ежедневно просматривает большое количество видеоматериала с помощью телевизора, компьютера или смартфона, не задумываясь о том, каким образом оно его получает. Учёные, ещё в начале 20 века, были озадачены тем, как запечатлеть то, что люди видят своими глазами, чтобы была возможность вновь воспроизвести эту информацию в другом месте.

Изначально был придумана механическая система, работающая по прицепу факса [1] (Рисунок 3.1.1) – металлическая подложка, на которую было нанесено статичное изображение непроводящей электричество краской, сканировалась построчно маятником, передавая сигнал приёмнику о том фон сейчас (металлическая подложка) или изображение. Приёмник состоял из маятника, который синхронизирован с маятником передатчика, и пластинки, которая темнела, когда на неё подавалось напряжение. Когда на приёмник приходил сигнал – это означало что сейчас передаётся фон и на пластинку маятником подавалось напряжение, из-за чего строка темнела.

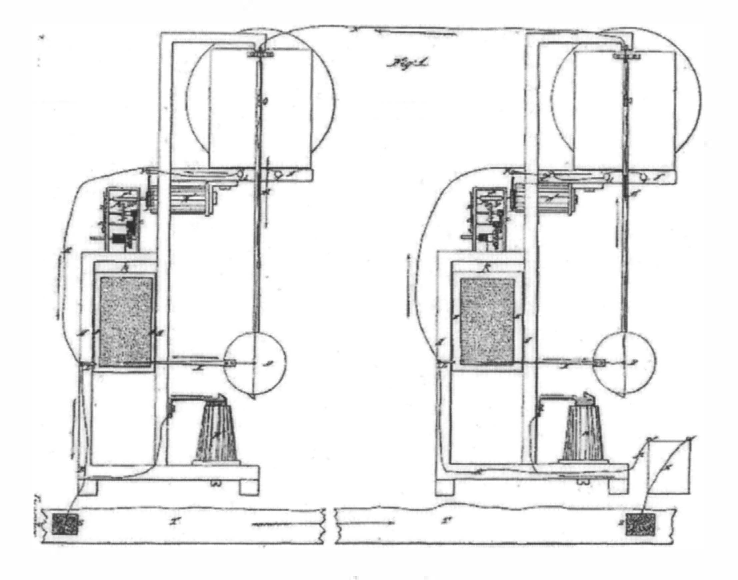


Рисунок . – Схематичное представление машины Александра Бэйна [1]

Однако, от этой системы в итоге отказались, по очевидным причинам: необходимо синхронизировать маятники, можно передавать только статичное изображение.

Чуть позже была придумана ещё одна механическая система, использующая диск Пауля Нипкова (Рисунок 3.1.2) для сканирования изображений [2, 3].

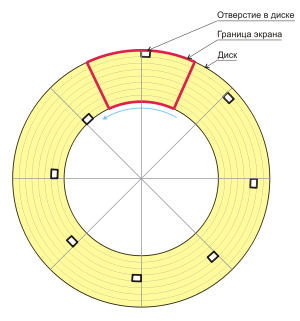


Рисунок . – Диск Нипкова

В передатчике находились диск Нипкова и фотоприёмник. Информация передавалась с фотоприёмника в приёмник, который состоял из диска Нипкова и источника яркого света. Оба диска Нипкова обязаны быть синхронизированными, вращаться с одной и той же скоростью в одну сторону.

Принцип работы этой механической системы состоял в том, что объект съёмки, находящийся перед диском, подсвечивался и построчно сканировался с помощью первого диска и фотоприёмника, расположенного за диском. После чего, сигнал с датчика использовался для модуляции яркости источника света, расположенного за вторым диском, из-за чего перед диском построчно появлялось изображение.

Разрешение изображения было ограничено размерами диска (вертикальное разрешение) и количеством отверстий в диске (горизонтальное разрешение).

Позже была разработана электронная система, основанная на электронно-лучевой трубке (Рисунок 3.1.3).

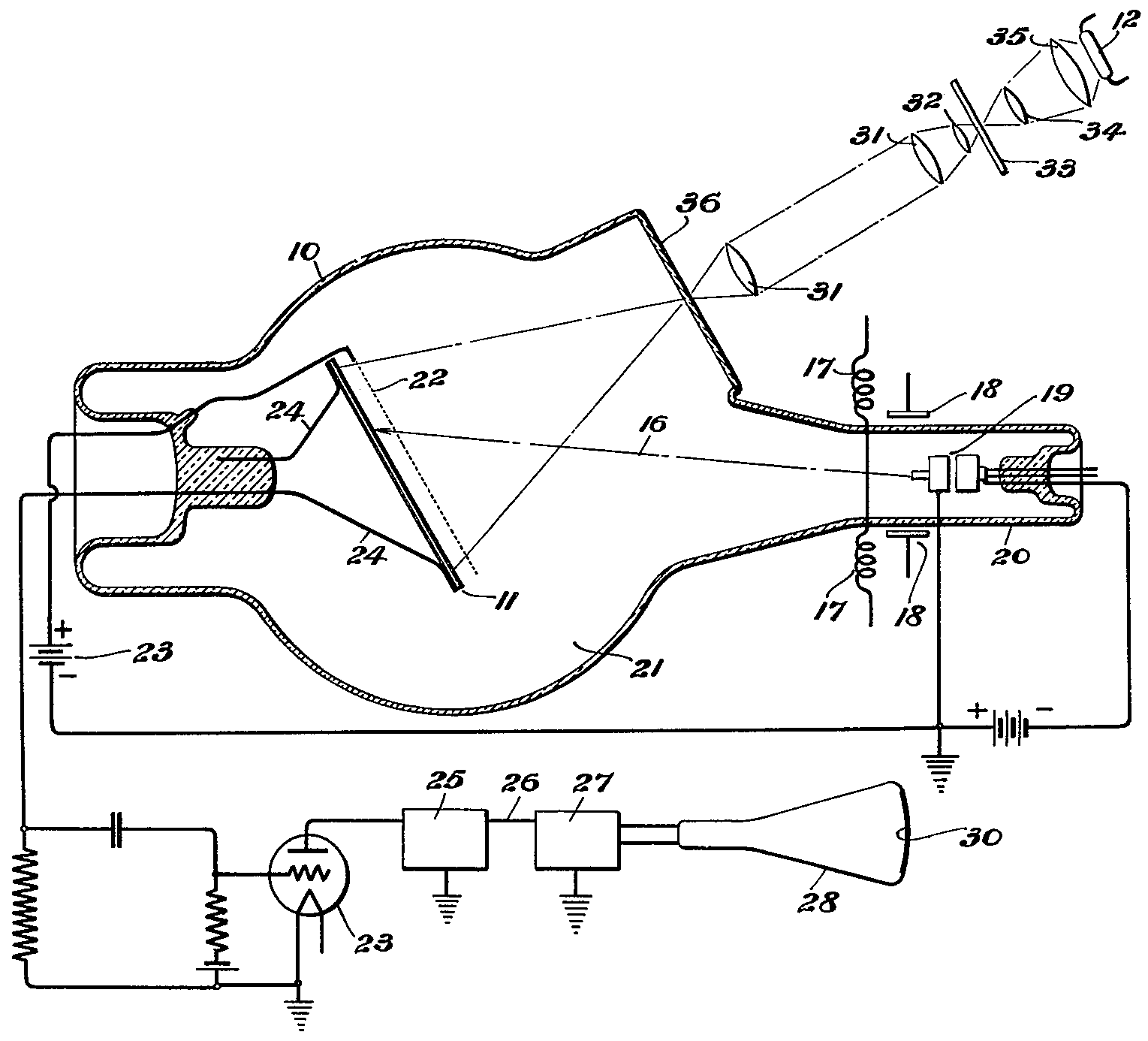


Рисунок . – Иконоскоп Зворыкина [4]

Изображение проецировалось на светочувствительный элемент, который состоял из двух слоёв, за счёт фотоэффекта [5] из слоя, на который проецировалось изображение выбивались электроны и накапливались на втором слое. Положительно заряженный слой построчно сканировался пучком электронов, из-за чего происходила перезарядка конденсатора, по току которого определялась яркость части изображения.

Изображение выводилось при помощи луча электронов, который построчно подсвечивал люминофор, находившийся на экране. Так было изобретено телевидение.

Основная задача передачи изображения была решена, но не было понятно, как сохранять запечатлённую информацию, чтобы использовать её вновь. Десятилетиями человечество записывало экран с телевизора на плёнку, после чего эта плёнка проявлялась, и, если была необходимость вновь передать содержимое плёнки на расстояние, плёнка воспроизводилась и транслировалась по технологии, описанной выше.

С развитием технологий, человечество перестало использовать электронно-лучевые трубки и видеоплёнки, на замену им пришли жидкокристаллические экраны / экраны на органических светодиодах и жёсткие диски / твердотельные накопители соответственно. В наше время подавляющее большинство информации представлено в цифровом виде и хранится она тоже в цифровом виде.

В наше время съемка и запись видео используется повсеместно:

* Телевидение – практически все передачи записываются заранее, иногда появляется необходимость прямых эфиров, например, для показа матча по футболу, который проводится в какой-либо стране и транслируется в этот же момент на телевизоры зрителей по всему земному шару;
* Камеры в банках, метро, магазинах, на дорогах, в подъездах, в автомобилях (видеорегистраторы, запись которых может помочь определить виновника дорожно-транспортного происшествия) и так далее – камеры записывают материал и сохраняют его либо на локальном носителе информации, типа флэш-карт, либо отправляют его на удалённый сервер по проводным или беспроводным интерфейсам;
* Видео в интернете – люди заранее снимают видео на камеру или смартфон, после чего обрабатывают его на компьютере и загружают на сайт, либо с новостями, либо на хостинг видео (типа YouTube или подобные). Так же в интернете присутствует аналог прямого эфира телевидения, когда снимаемое видео сразу же воспроизводится у пользователей сайта (это называется стриминг (от английского stream - поток), основные площадки YouTube и Twitch).

В каждой из областей применения видео используется память устройства или пропускная способность интерфейса.

Если изначально размеры экранов достигали значений 480p, т.е. 640 на 480 пикселей, то сейчас разрешения экранов могут достигать значений 16K   
[6, 7], что равняется 15360 на 8640 пикселей. Для отображения видео на экране определённого разрешения без искажений, необходимо чтобы разрешение видео соответствовало разрешению экрана.

Посчитать количество памяти необходимой для хранения несжатого видео можно следующим образом:

|  |  |  |
| --- | --- | --- |
|  | , | () |

где W – (width) ширина одного кадра в пикселях, H – (height) высота одного кадра в пикселях, n – количество компонентов цвета для представления одного пикселя, BPC – (bits per color) количество бит на один цвет, FPS – (frame per second) количество кадров в секунду, t – (time) длительность видео.

Используя формулу (1) можно вычислить сколько будет занимать памяти видео длительностью одна минута с частотой кадров шестьдесят кадров в секунду в цветовом пространстве RGB888 (это цветовое пространство, в котором пиксель представляется тремя цветами: красным, зелёным и синим; количество бит на каждый цвет равно восьми) в разрешениях 480p (640 на 480 пикселей) и FullHD (1920 на 1080 пикселей): примерно 24,72 Гбит и 166,85 Гбит соответственно.

Не сложно посчитать, что, например, одна серия какого-либо сериала, длительностью (обычно) сорок пять минут будет занимать 139 Гбайт для разрешения 480p и 939 Гбайт для разрешения FullHD.

Получаются очень большие значения, учитывая, что сейчас наиболее распространёнными считаются жёсткие диски объёмом 1 Тбайт и 2 Тбайта, а твердотельные накопители объёмом от 250 Гбайт до 1 Тбайта.

В настоящее время видео может передаваться по различным интерфейсам, например по Ethernet, при просмотре видео в интернете или использовании IP-телевидения видео передаётся непрерывно и сразу же воспроизводится – это называют видеопотоком.

Приравняв время одной секунде в формуле (1) можно получить формулу вычисления необходимой пропускной способности для видеопотока:

|  |  |  |
| --- | --- | --- |
|  | , | () |

По формуле (2) для видеопотоков 480p и 1080p (другое обозначение FullHD разрешения) RGB888 при частоте кадров равной шестидесяти кадрам в секунду, потребовалась бы пропускная способность канала передачи равная 0,412 Гбит/с и 2,78 Гбит/с соответственно.

Если всё-таки рассматривать передачу видеопотока по интернету, то можно сделать следующий вывод – в настоящее время наиболее распространённой и доступной скоростью интернета является 100 Мбит/с, очевидно, что ни 480p, ни 1080p несжатые видеопотоки не получится передать.

Исходя из расчётов и выводов выше, можно сказать, что требуется как-то сжимать видео, либо перед отправкой, если это видеопоток с камеры, либо перед тем, как сохранять видео в память. Для этого были придуманы видеокодеки.

## Как

Видеокодек – это программная или аппаратная обработка видеопотока, которая сжимает входное видео и упаковывает его в определённом формате так, чтобы в последствии его можно было распаковать и получить исходное видео.

Основная идея видео кодирования заключается в том, чтобы убрать избыточность, представленную в несжатом видеопотоке. Под избыточностью подразумевается повторение данных и то, что субъективно для человеческого глаза одинаково.

Помимо уже упомянутого цветового пространства RGB было разработано световое пространство YCbCr (также известное как YUV, где U это Cb, а V это Cr), которое так же состоит из трёх компонент, но цветовых компонент только две: Y – яркостная составляющая, Cb – цветность синяя, Cr – цветность красная. Существуют формулы, закреплённые в стандарте [8], для перехода из цветового пространства RGB в YCbCr и обратно, при использовании 8-бит на канал:

|  |  |  |
| --- | --- | --- |
|  |  | () |

|  |  |  |
| --- | --- | --- |
|  |  | () |

Пример преобразования цветового пространства RGB в YCbCr продемонстрирован на рисункеРисунок 3.2.1, цветовые каналы представлены в градациях серого.



Рисунок . – Преобразование RGB в YCbCr

Человеческий глаз лучше определяет изменение яркостной составляющей изображения чем цветовой, поэтому были разработаны различные методы цветовой субдискретизации изображения [9].

Цветовая субдискретизация – способ кодирования изображения, когда разрешение цветовой составляющей видео отличается от яркостной. Формат субдискретизации обычно представлен соотношением трёх чисел вида X:a:b, где X – частота дискретизации яркостного канала, a – частота дискретизации цветовой составляющей в первой строке, b – частота дискретизации цветовой составляющей во второй строке.

В видео кодировании наиболее распространённым считается формат 4:2:0, в котором блоку 4 на 4 значений яркости соответствует блок 2 на 2 значений цветности (пример Рисунок 3.2.2), то есть одному значению цветности соответствует 4 значения яркости.

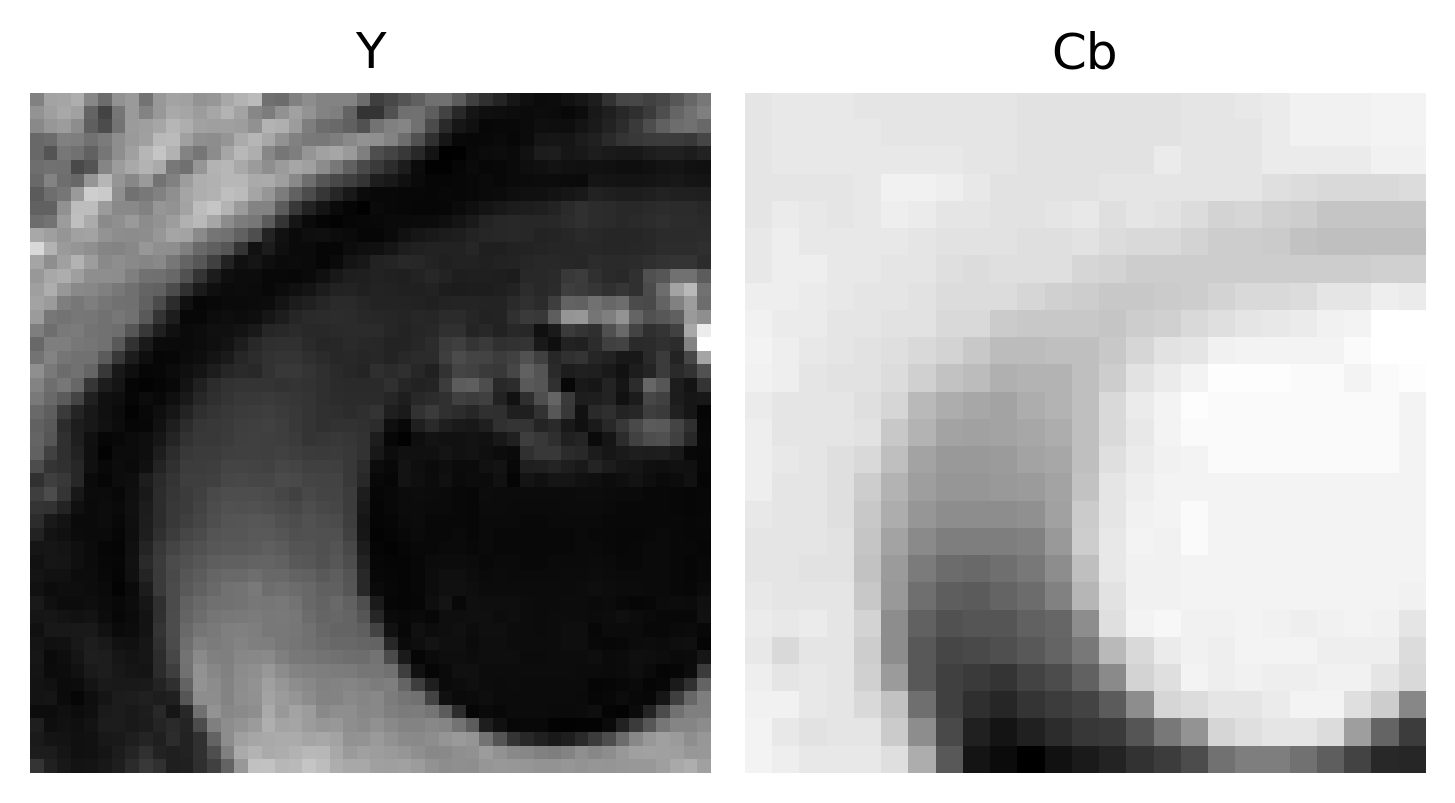


Рисунок . – YCbCr в формате субдискретизации 4:2:0

За счёт цветовой субдискретизации происходит сжатие в два раза, потому что до субдискретизации (формат 4:4:4) четыре пикселя занимали (где 4 – пикселя, 8 – бит на канал, 3 – канала), после субдискретизации четыре пикселя занимают (где 4 – значения яркости, 8 – бит на канал, 2 – значения цветности).

Пространственное сжатие – сжатие на основе предсказаний внутри одного кадра. Кадр видеопотока разбивается на блоки 4х4 пикселя или 16х16, после чего для каждого следующего блока происходит выбор одного из предсказания: вертикальное, горизонтальное, DC (среднеарифметическое), диагональное вниз-влево, диагональное вниз-вправо, вертикально-вправо, горизонтально-вниз, вертикально-влево, горизонтально-вверх. Предсказания вычисляются из соседних уже обработанных пикселей блока, после чего выбирается то предсказание, мощность (например, вычисленная по формуле SAE) разности которого с исходным изображением будет наименьшим.

Временное сжатие – сжатие, основанное на компенсации движения, происходящего между кадрами. Для каждого блока 4х4 текущего кадра вычисляется блок предыдущего кадра, который даёт наименьшую разностную мощность (например SAE), строится вектор сдвига текущего блока относительно блока с предыдущего кадра – этот вектор, состоящий из двух чисел, и представляет блок 4х4 в текущем кадре. При восстановлении кадров, информация о блоке будет браться с предыдущих кадров.

Дискретное косинусное преобразование (ДКП или DCT – от английского discrete cosine transform) – ортогональное преобразование, обычно используется ДКП-2, потому что оно обладает свойством «уплотнения энергии» в левом верхнем углу выходной матрицы (пример Рисунок 3.2.3).



Рисунок . – Пример результата ДКП над блоком 8х8

Преобразования проводятся над блоками от 4х4 до 32х32. «Уплотнение энергии» на практике означает, что ДКП собирает низкочастотную составляющую блока в левом верхнем углу, а высокочастотную составляющую в правом нижнем углу. Это очень полезное свойство, потому что результат ДКП можно квантовать, т.е. поэлементно разделить результат ДКП на соответствующие коэффициенты матрицы квантования (Рисунок 3.2.4), тогда в блоках, взятых с однородного участка кадра (например небо) в правом нижнем углу матрицы будут нули (Рисунок 3.2.5)

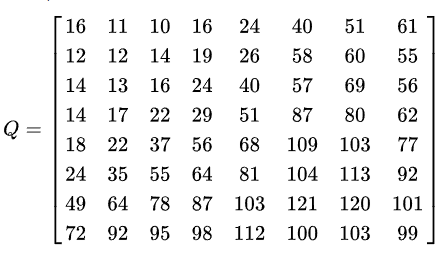


Рисунок . – Пример матрицы квантования 8х8

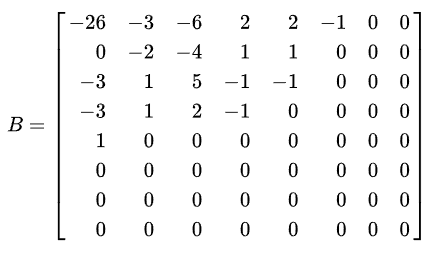


Рисунок . – Пример результата квантования блока 8х8

Сжатие при помощи ДКП осуществляется путём разворачивания двумерного квантованного блока при помощи алгоритма зигзаг (Рисунок 3.2.6) в одномерный вектор. У получившегося вектора будут серии чисел, то есть последовательно идущие одинаковые значения, которые кодируются при помощи алгоритмов энтропийного кодирования (RLE, код Хаффмана, CAVLC, CABAC). Например, последовательность нулей, может быть закодирована всего двумя числами (36, 0), что будет занимать всего два байта, вместо 36 байт.

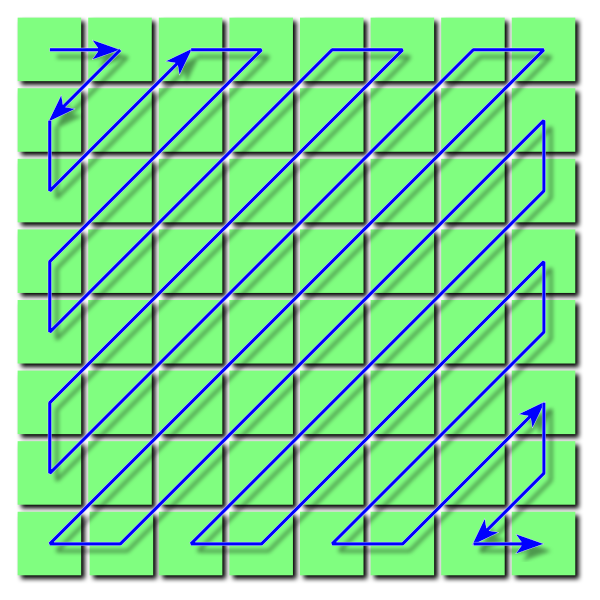


Рисунок . – Алгоритм сканирования блока «зигзаг»

Так же стоит упомянуть, что реконструированное изображение будет отличаться от исходного из-за квантования, чем сильнее квантование, тем лучше сжатие и хуже восстановленное изображение.

Описанные выше алгоритмы могут быть использованы вместе, чтобы получить наилучшее сжатие.

На сегодняшний день существует несколько видеокодеков позволяющих сжимать видеопоток:

Motion JPEG (разбивает видеопоток на отельные кадры, которые кодируются в JPEG изображения с использованием ДКП);

H.261 / H.262(MPEG-2) – видеокодеки использующие пространственные и временные предсказания для уменьшения битрейта;

H.263 / H.264(MPEG-4 или ещё AVC - Advanced Video Coding) – видеокодеки основанные на стандарте H.261, то есть с использованием пространственных и временных предсказаний и также добавлено сжатие с использованием DCT и энтропийного кодирования;

H.265 или HEVC (High Efficiency Video Coding) – дополнение стандарта позволяющее кодировать видео разрешение до 8К, улучшение временных и пространственных предсказаний для более низкого выходного битрейта;

VP9 – аналог стандарта H.265 от Google;

AV1 – видеокодек разработанный альянсом Google, Amazon, Cisco, Intel, Microsoft, Mozilla и Netflix, является следующим поколением стандарта VP9, на 50% лучше сжатие чем у H.264 и на 30% чем у VP9 при одинаковом качестве видео;

H.266 или VVC (Versatile Video Coding) – поддержка видео от 4К до 16К с глубиной цвета более 16 бит на канал, сложность кодирования до десяти раз выше чем у HEVC, сложность декодирования в 2 раза выше чем у HEVC.

Изначально была поставлена задача real-time кодирования видео с низкими задержками, поэтому было решено реализовать видеокодек на ПЛИС на основе стандарта H.264, так как он самый популярный на данный момент и относительно новых кодеков гораздо проще алгоритмическая сложность.

## Как это делаю я

## Что у меня получилось

# Заключение

# Список использованных источников

1. Giuseppe Pelosi. “The Great Men in Telecommunications:” Giovanni Caselli and the Invention of the Pantelegraph or Fax [Historical Corner] // IEEE Antennas and Propagation Magazine (Volume: 52, Issue: 3, June 2010), 27 September 2010, p. 194 – 198.
2. Б. Шефер «Самодельный телевизор» Детиздат, 1937 год, 34 стр.
3. Walter Barbirato. Fracarro, from the disk of Nipkow to the digital convergence // 2008 IEEE History of Telecommunications Conference, 12 Sept. 2008.
4. Zworykin, V. K. (n.d.) [filed 1931, patented 1935]. "Method of and Apparatus for Producing Images of Objects". Patent No. 2,021,907. United States Patent Office.
5. Яворский Б. М., Детлаф А. А., Лебедев А. К. Справочник по физике для инженеров и студентов вузов. — М., Оникс, 2007. — ISBN 978-5-488-01248-6. — с. 725
6. Video Electronics Standards Association (June 26, 2019). "VESA Publishes DisplayPort 2.0 Video Standard Enabling Support for Beyond-8K Resolutions, Higher Refresh Rates for 4K/HDR and Virtual Reality Applications". URL: <https://vesa.org/press/vesa-publishes-displayport-2-0-video-standard-enabling-support-for-beyond-8k-resolutions-higher-refresh-rates-for-4k-hdr-and-virtual-reality-applications/>. (Дата обращения: 12.01.2022).
7. DisplayDaily (August 23, 2018). "Innolux Premieres World's First 100 Inch 16K Display The Best Vision Ever".   
   URL: <https://www.displaydaily.com/article/press-releases/innolux-to-showcase-smartwatch-with-flexible-oled-display>. (Дата обращения: 12.01.2022).
8. T.871: Information technology - Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF). ITU-T. Approved in 2011-05-14. Posted in 2012-09-11.
9. Christian. J. van den Branden Lambrecht. "Vision Models and Applications to Image and Video Processing". 2001. p. 209.   
   ISBN 0-7923-7422-3.

# Приложения