

Оптимизация альфа-смешения цветов в пространстве RGBA

Презентация к курсовому проекту

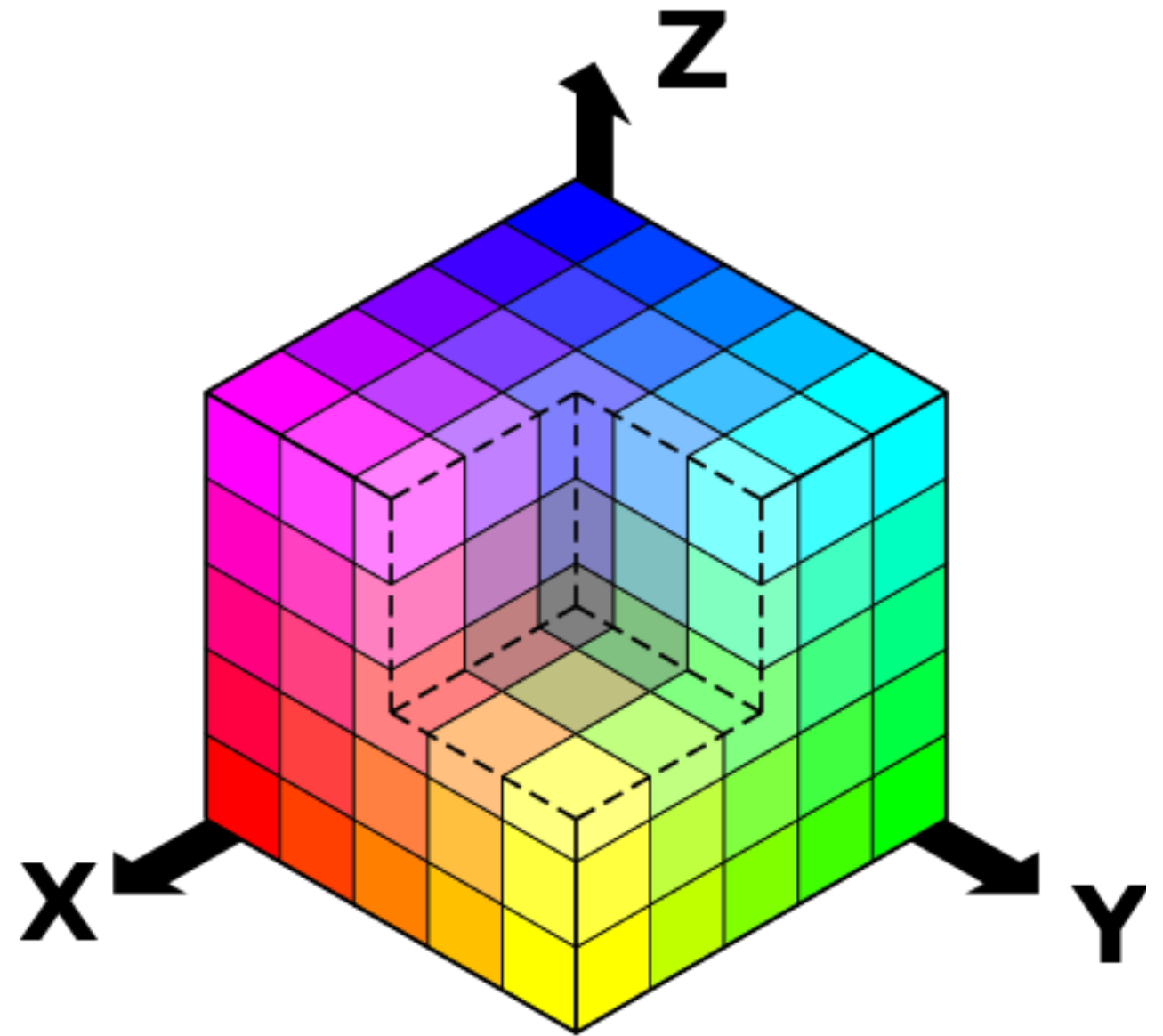
*Щербатюк Дарья
ИУ7-54
2017*

Постановка задачи

- В соответствии с техническим заданием на курсовой проект необходимо реализовать программный модуль для сравнения различных алгоритмов смешения цветов в пространстве RGBA. Внедрение оптимизированного алгоритма будет производиться в открытый графический редактор KDE Krita, имеющий несколько реализаций для работы с цветом.

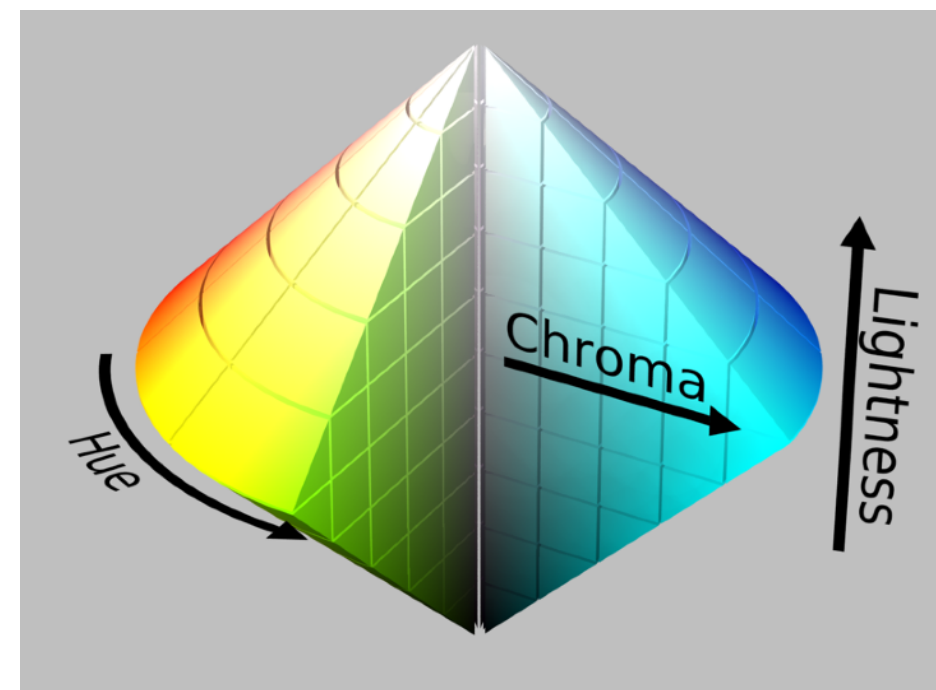
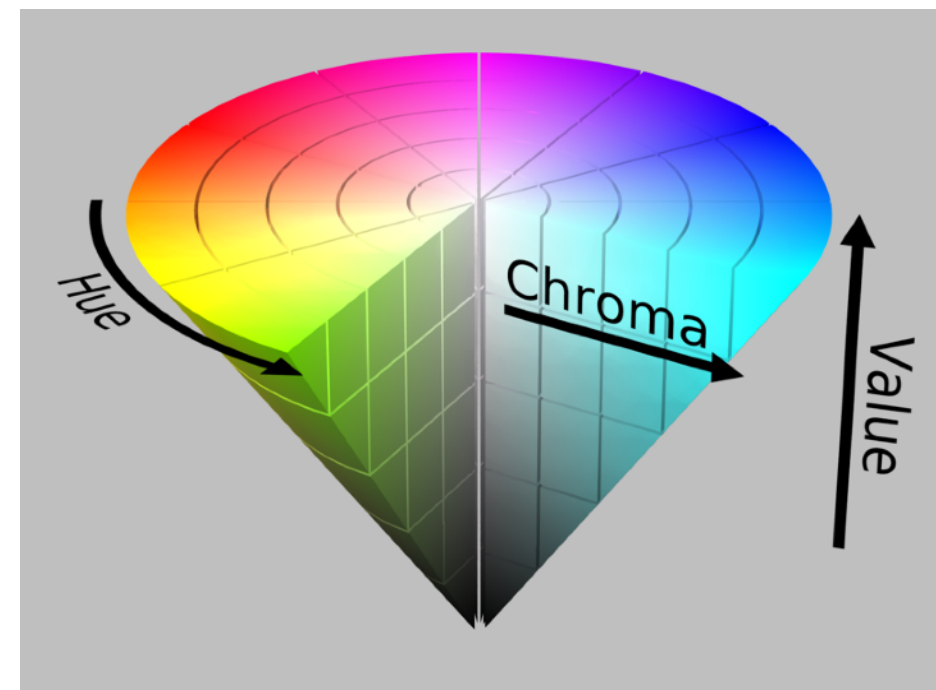
RGB

- Аддитивная цветовая модель
- Три аддитивных первичных цвета: красный **R**, зеленый **G** и синего **B**.
- Цвет можно представляется в виде триплета чисел от 0 до определенного максимального значения, соответственно представляющих долю основных **красного**, **зеленого** и **синего** цветов.

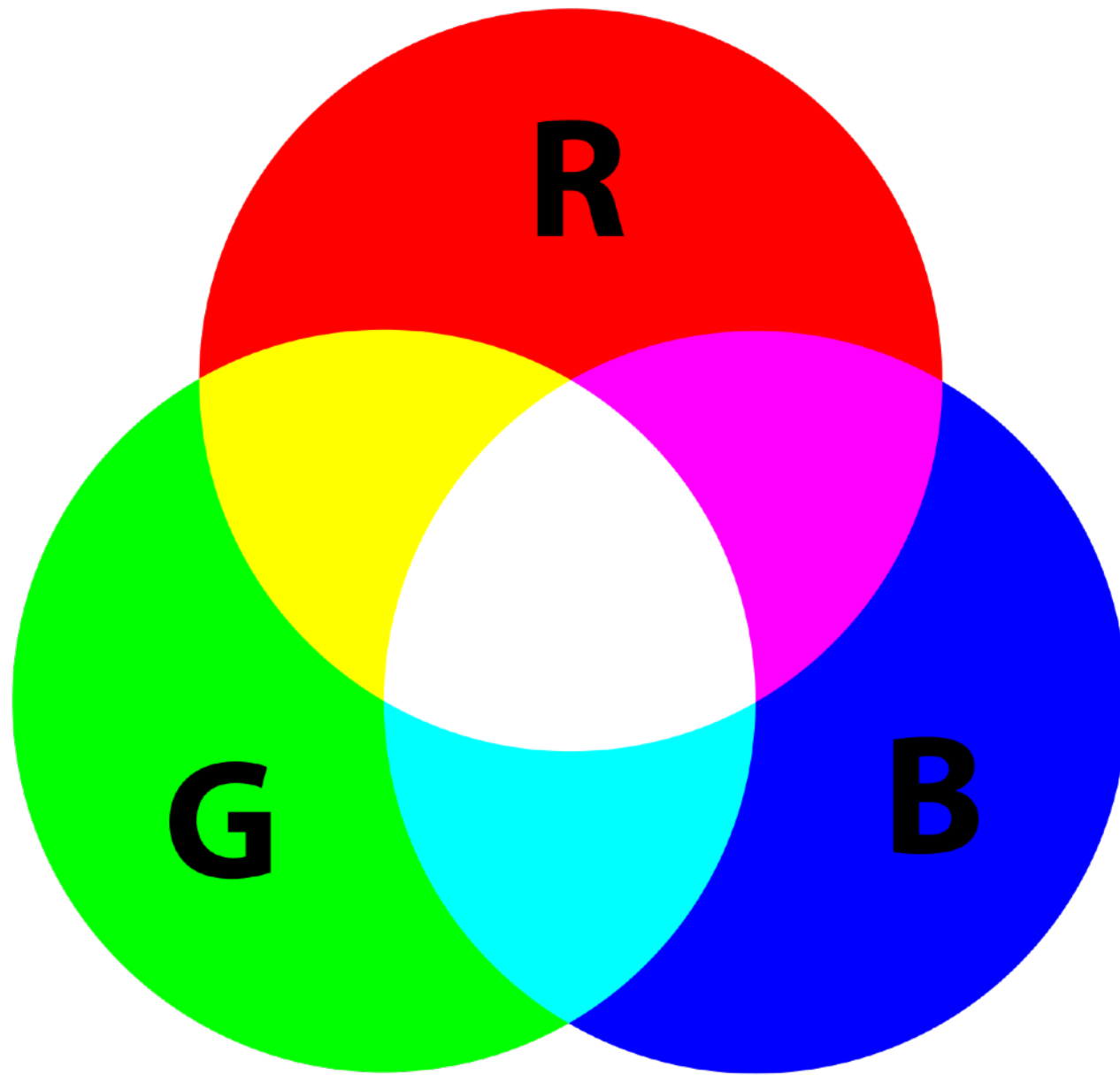


HSL и HSV

- HSL, HLS или HSI — цветовая модель, в которой цветовыми координатами являются тон (Hue), насыщенность (Saturation) и светлота (Lightness/Intensity). В HSV или HSB светлота заменяется яркостью (Brightness) или значением цвета (Value).

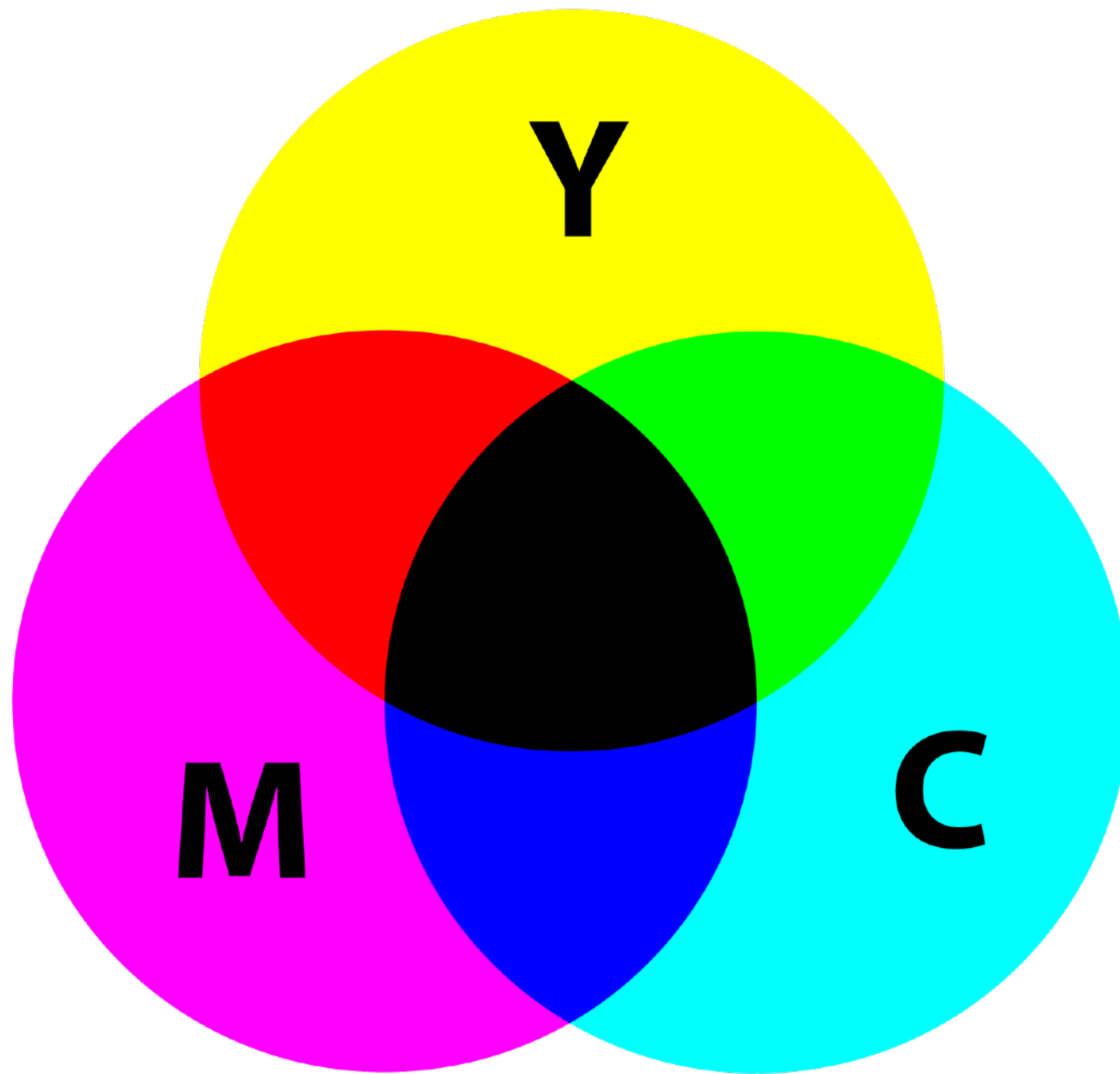


АДДИТИВНЫЙ СИНТЕЗ



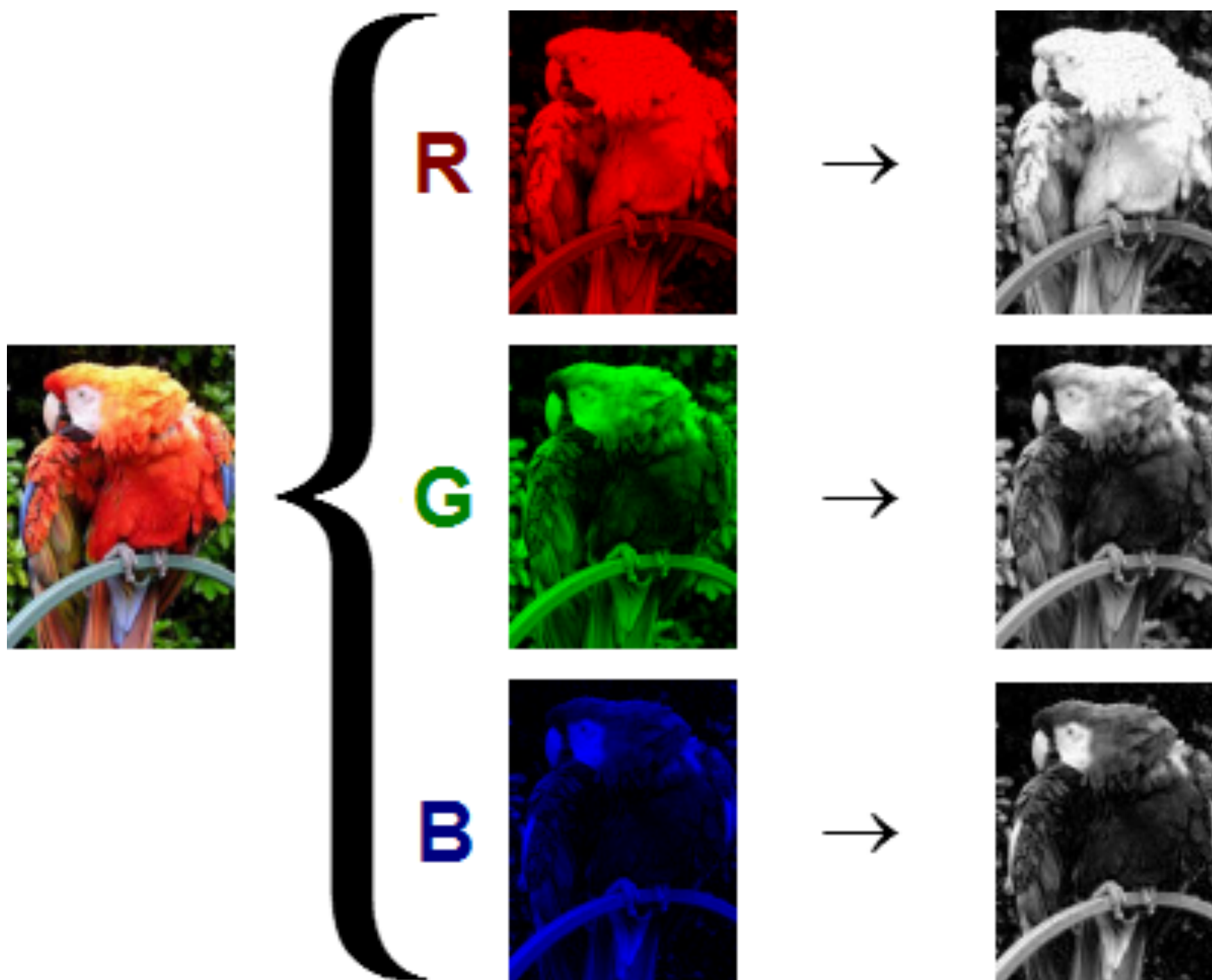
- Аддитивный цвет - это цвет, созданный путем смешивания нескольких различных цветов *света*
- Компьютерные мониторы и телевизоры являются наиболее распространенными примерами аддитивного синтеза.

Субтрактивный синтез



- Субтрактивный синтез объясняет смешение ограниченного набора **красителей, красок, пигментов** или натуральных красителей для создания более широкого диапазона цветов, каждый из которых является результатом частичного или полного вычитания (то есть, поглощения) некоторых длин волн света.

Каналы

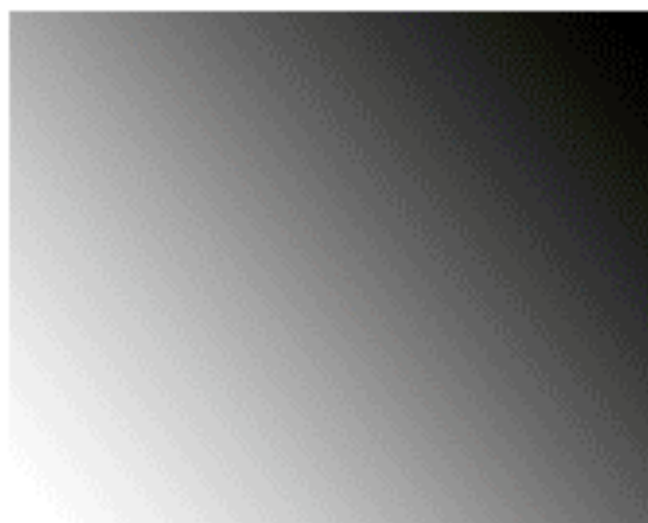


Альфа-каналы



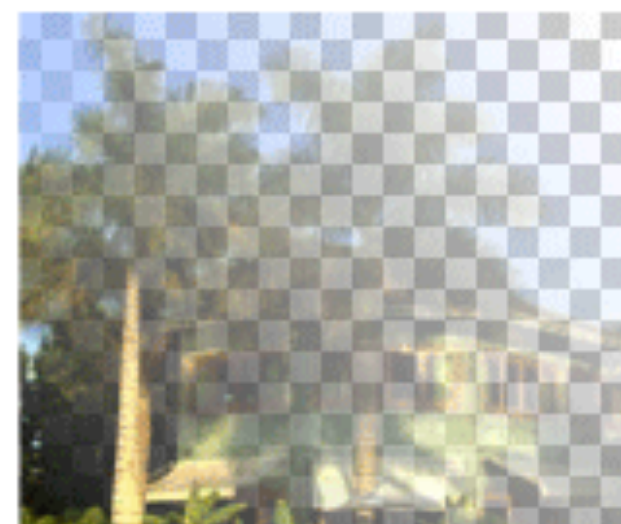
RGB

+



Alpha

=



Composite

Альфа-смешение



+



Background

=



$$\begin{cases} \alpha_{out} = \alpha_{src} + \alpha_{dst}(1 - \alpha_{src}) \\ out_{RGB} = \frac{(src_{RGB}\alpha_{src} + dst_{RGB}\alpha_{dst}(1 - \alpha_{src}))}{\alpha_{out}} \end{cases}$$

Прямая α

$$\begin{cases} \alpha_{out} = \alpha_{src} + \alpha_{dst}(1 - \alpha_{src}) \\ out_{RGB} = src_{RGB}\alpha_{src} + dst_{RGB}(1 - \alpha_{src}) \end{cases}$$

Премультимплексированная α

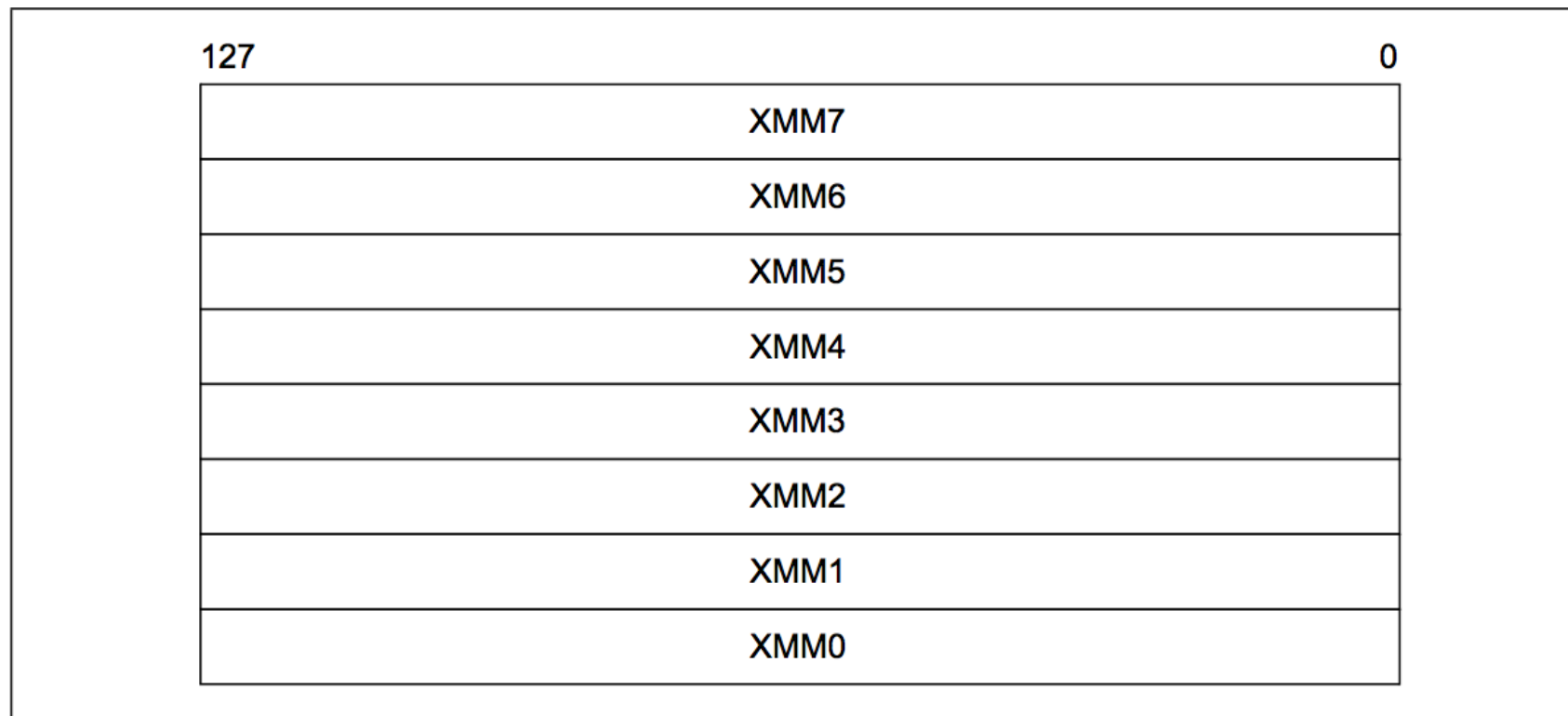
Распараллеливание вычислений

- Идея распараллеливания вычислений основана на том, что большинство задач может быть разделено на набор меньших задач, которые могут быть решены одновременно.
- Характер увеличения скорости программы в результате распараллеливания объясняется *законами Амдала и Густавсона*.
- Параллельные вычисления существуют в нескольких формах: на уровне битов, на уровне инструкций, на уровне данных.
- Взаимодействие и синхронизация между процессами представляют большой барьер для получения высокой производительности параллельных систем.

CUDA

- Позволяет существенно увеличить вычислительную производительность благодаря использованию GPU (графических процессоров) фирмы NVIDIA.
- Достоинствами CUDA является огромный прирост скорости выполнения расчётов по сравнению с расчетами на центральном процессоре компьютера.
- Недостатками является сложность программирования для CUDA (хотя производитель утверждает обратное), привязка к картам NVIDIA, и, как следствие, отсутствие переносимости между архитектурами.

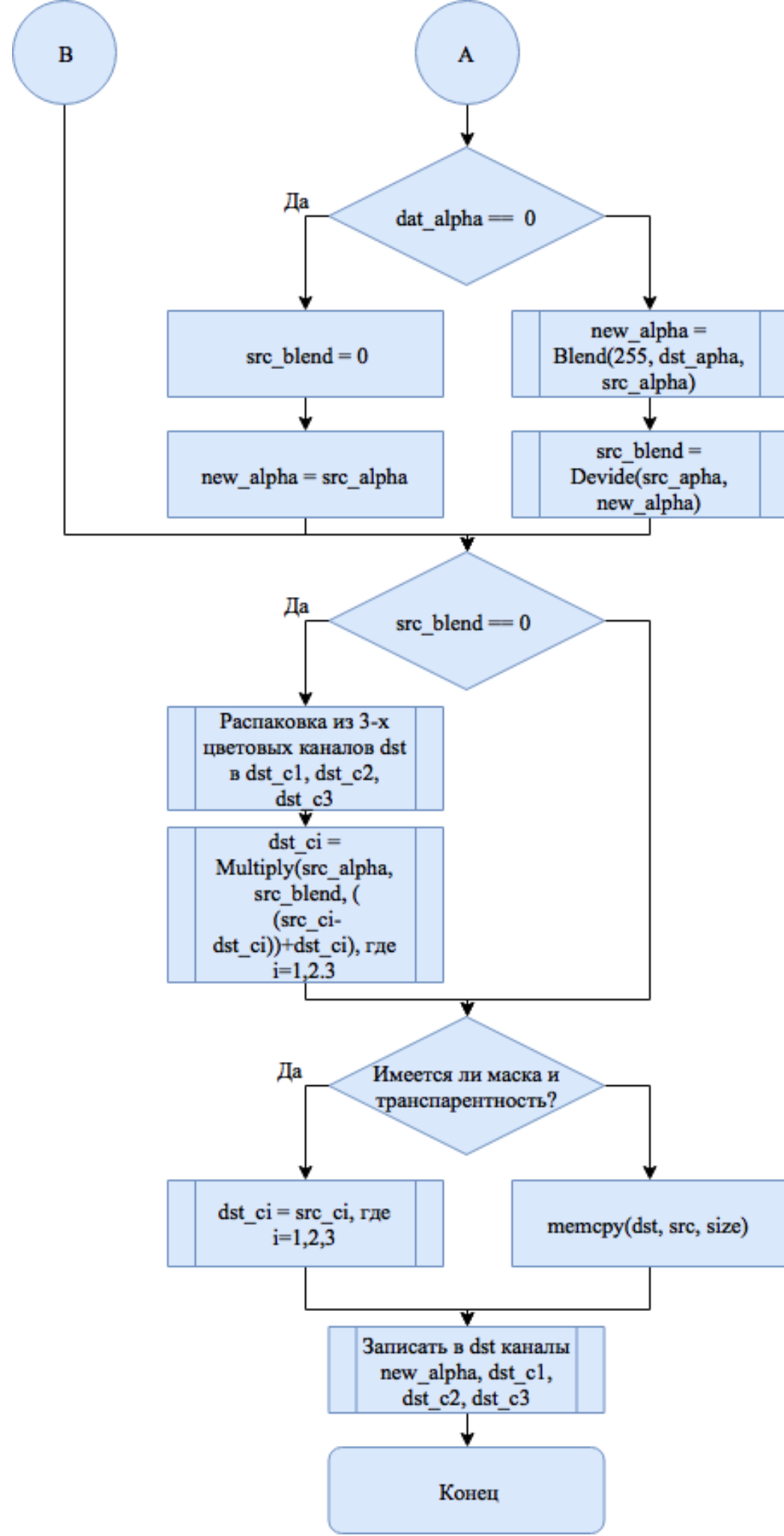
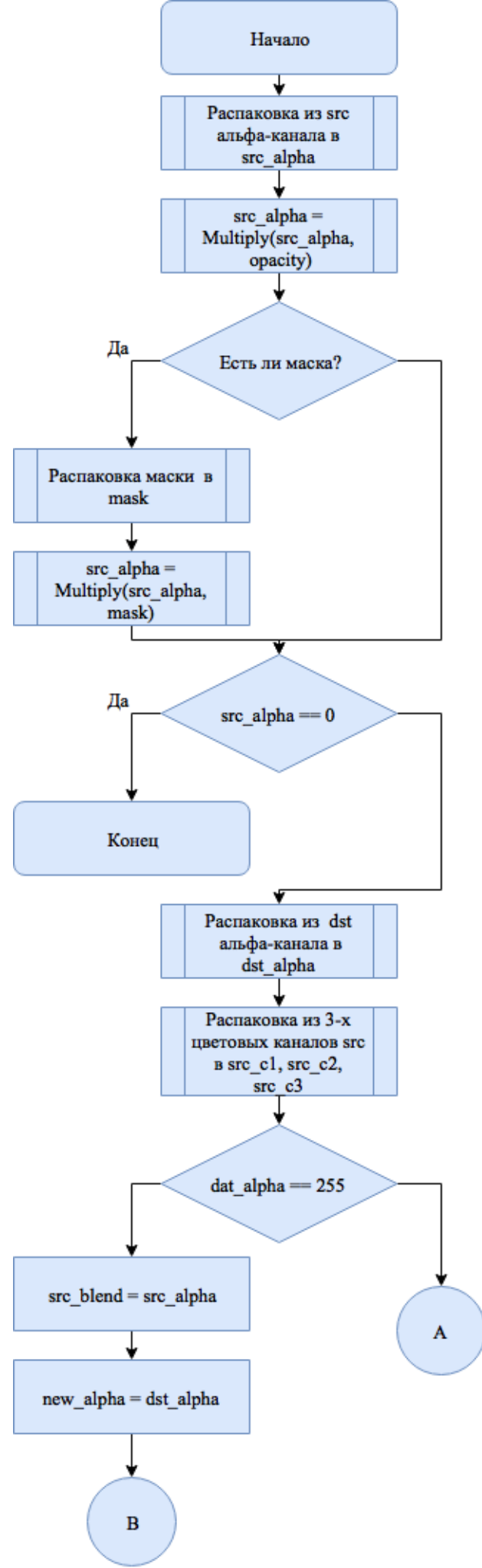
SSE

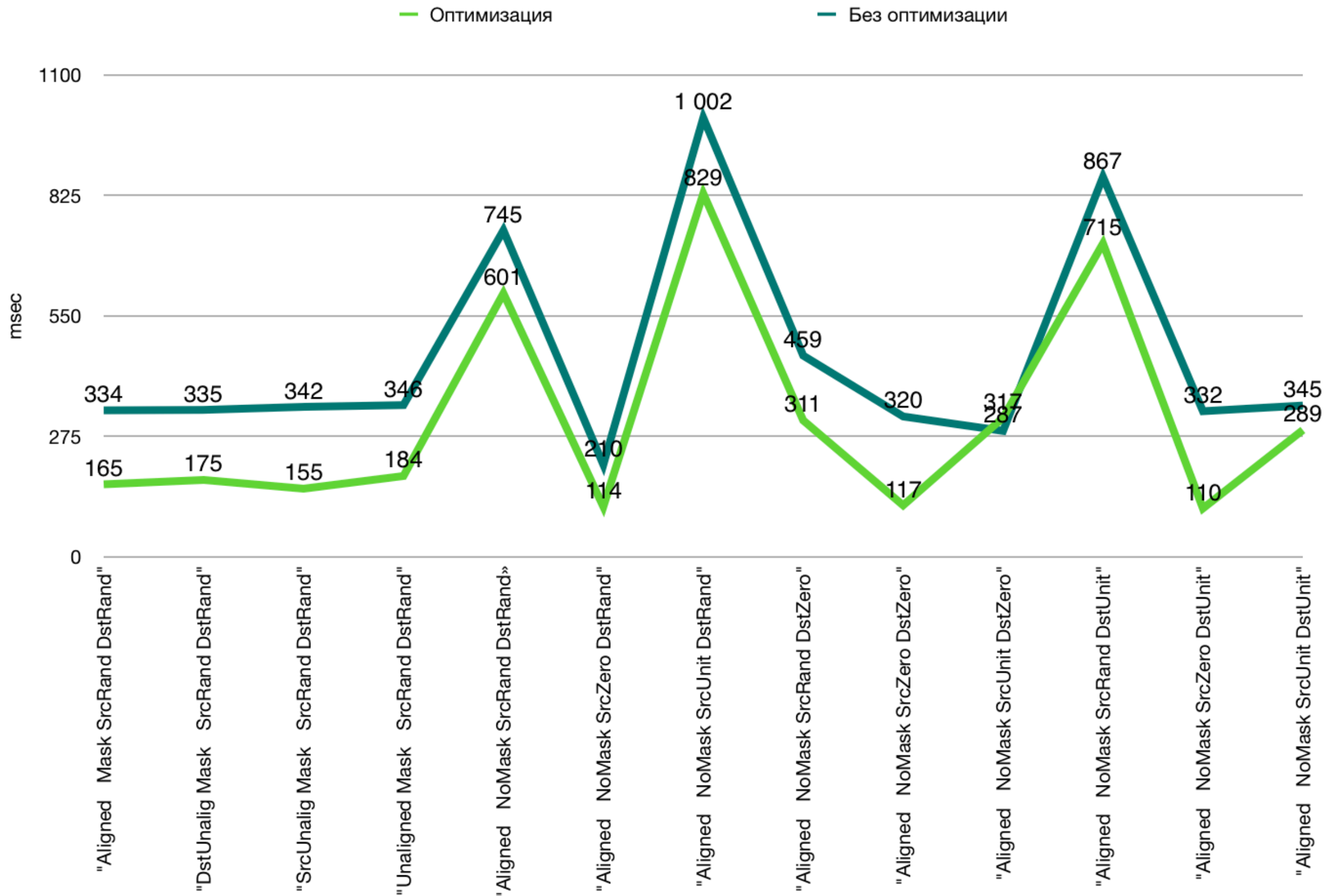


- Включает в архитектуру процессора восемь 128-битных регистров и набор из инструкций, работающих со скалярными и упакованными типами данных.
- Преимущество в производительности достигается в том случае, когда необходимо произвести одну и ту же последовательность действий над разными данными.

AVX

- Advanced Vector Extensions (AVX) являются расширениями x86 набора инструкций архитектуры для микропроцессоров от Intel и AMD.
- AVX использует шестнадцать регистров YMM0--YMM15 вместо XMM0—XMM15.
- Упрощено требование выравнивания операндов памяти.
- Смешивание неадаптированных SSE и AVX инструкций приводит к заметному снижению производительности.





Исследование

Спасибо за внимание!