



COMPUTER GRAPHICS

ЗАСОБИ ПРОГРАМУВАННЯ КОМП'ЮТЕРНОЇ ГРАФІКИ



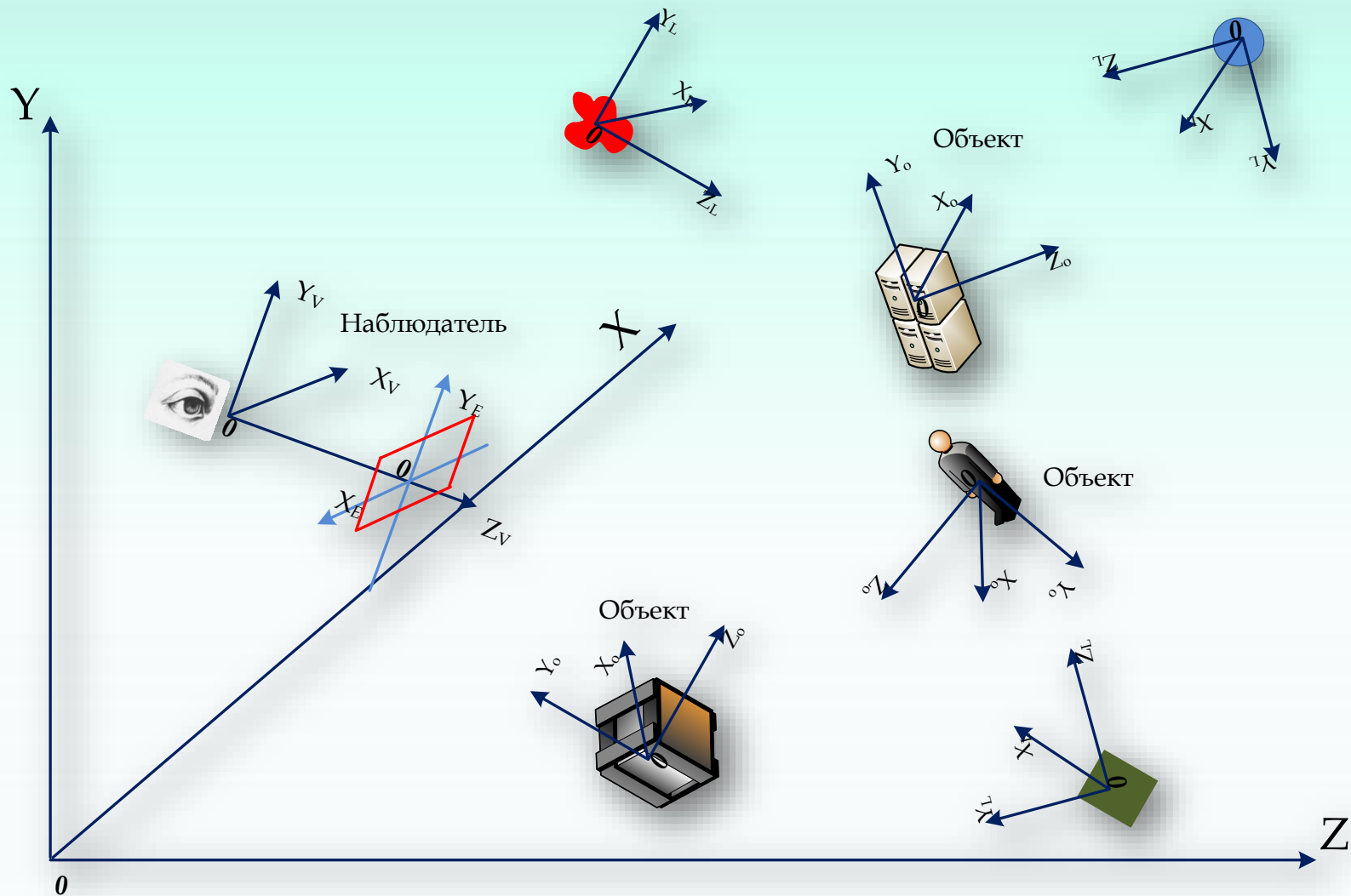
COMPUTER GRAPHICS

OPEN_GL (part 1)

Open GL

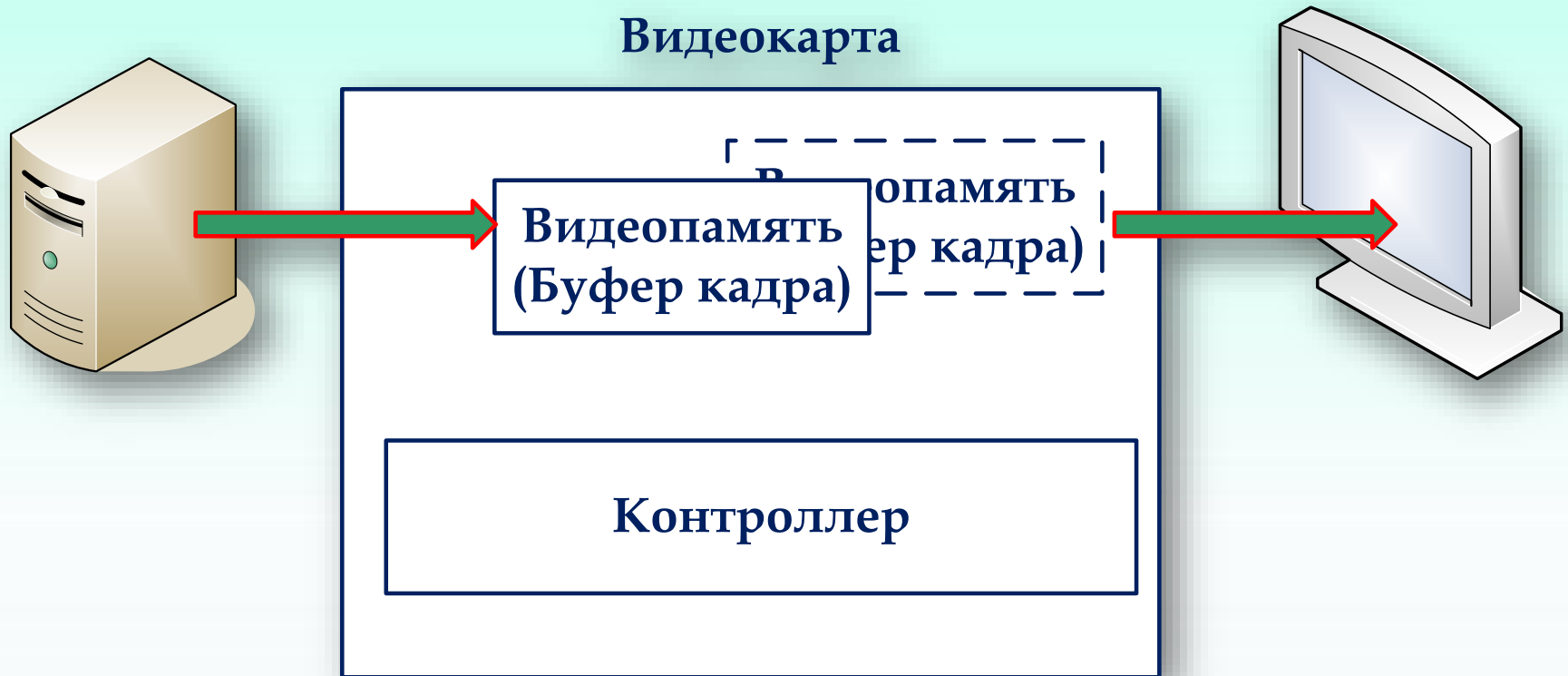
- Развитие видеокарт.
- Что такое OpenGL.
- Обзор архитектуры.
- Графический конвейер
- GLSL. Шейдеры

Задача визуализации



«Путь» видеокарт 1-е поколение (без графики)

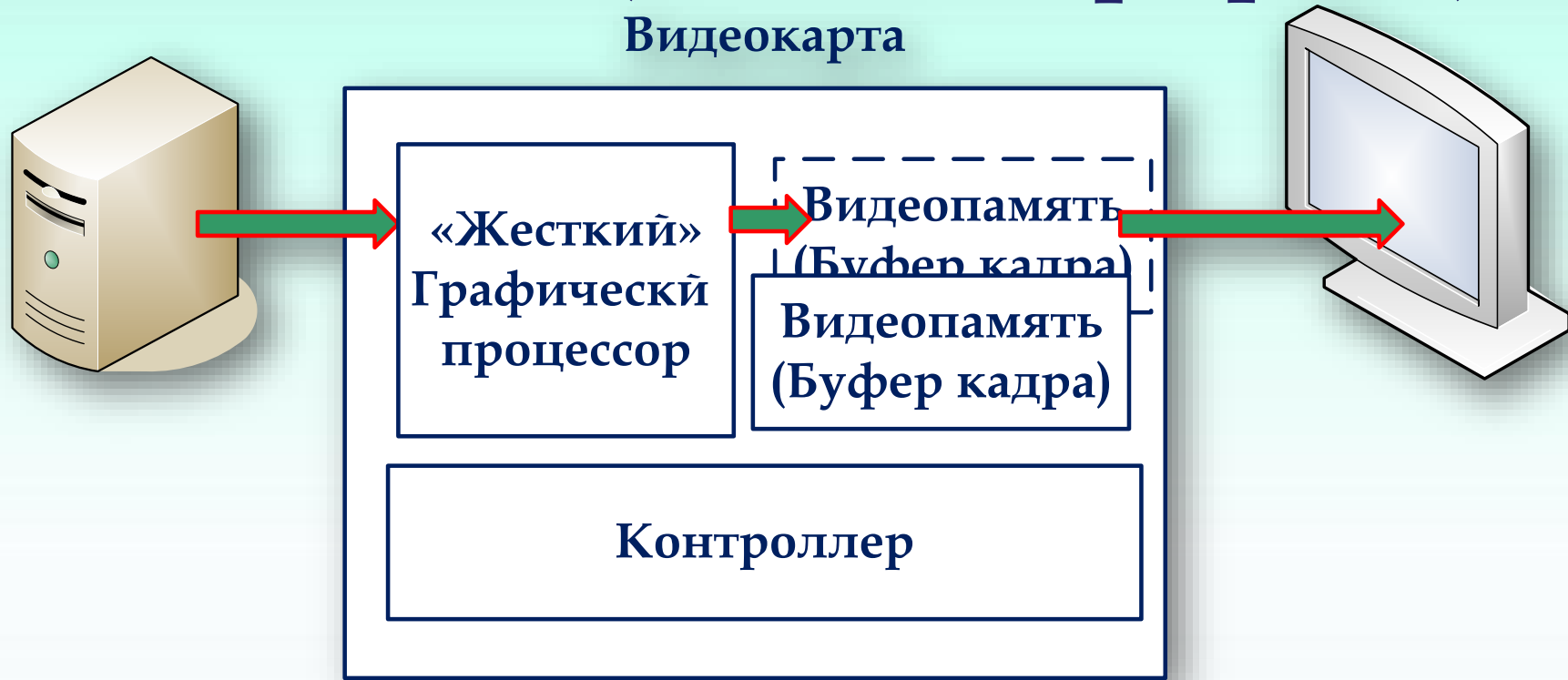
Видеокарта



«Путь» видеокарт

2-е поколение (жесткая графика)

Видеокарта



Графический спецпроцессор,
аппаратная поддержка
графики

«Путь» видеокарт 3-е поколение

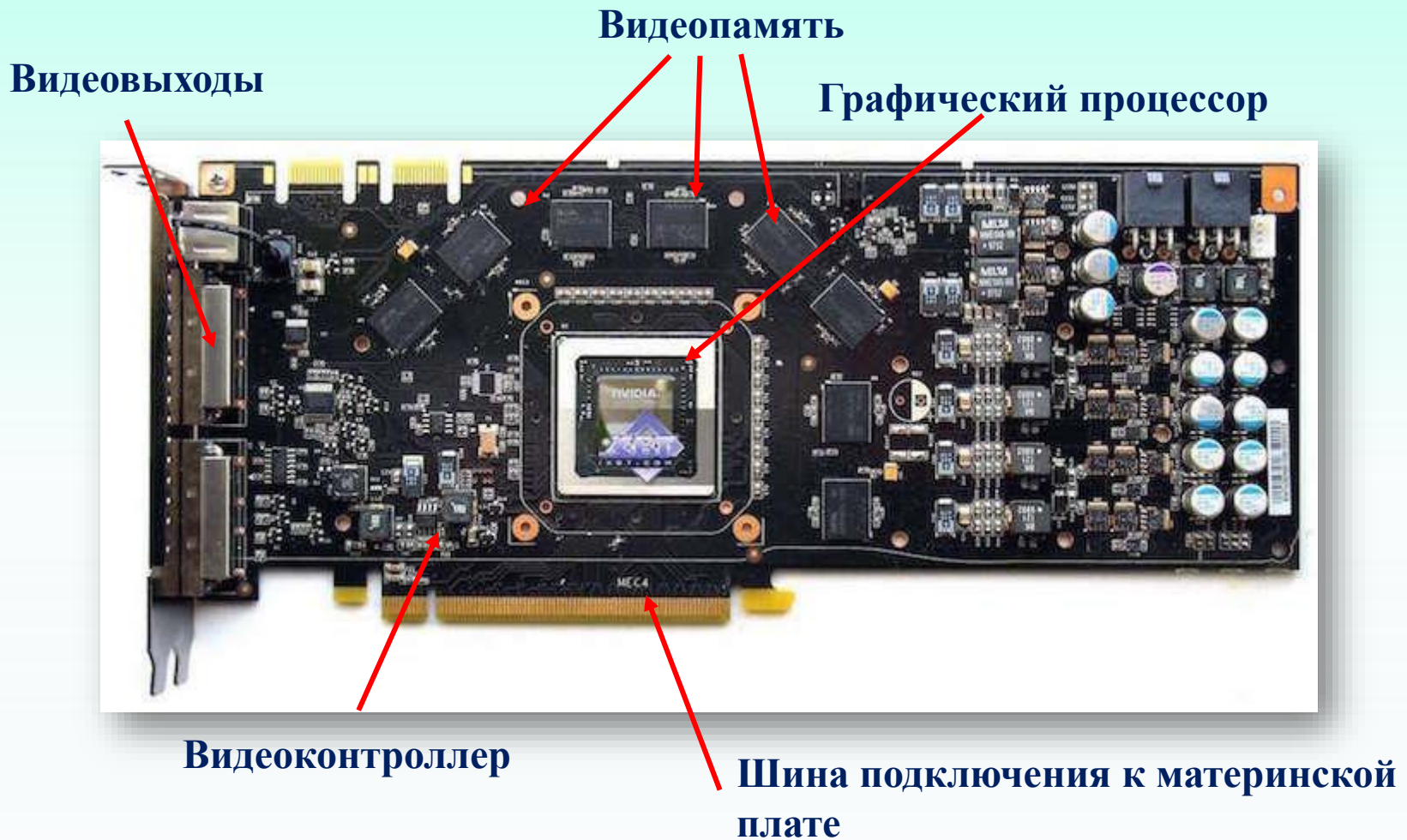
Видеокарта



Программируемый
графический спецпроцессор =
графический процессор = GPU

«Путь» видеокарт

Типичная видеокарта



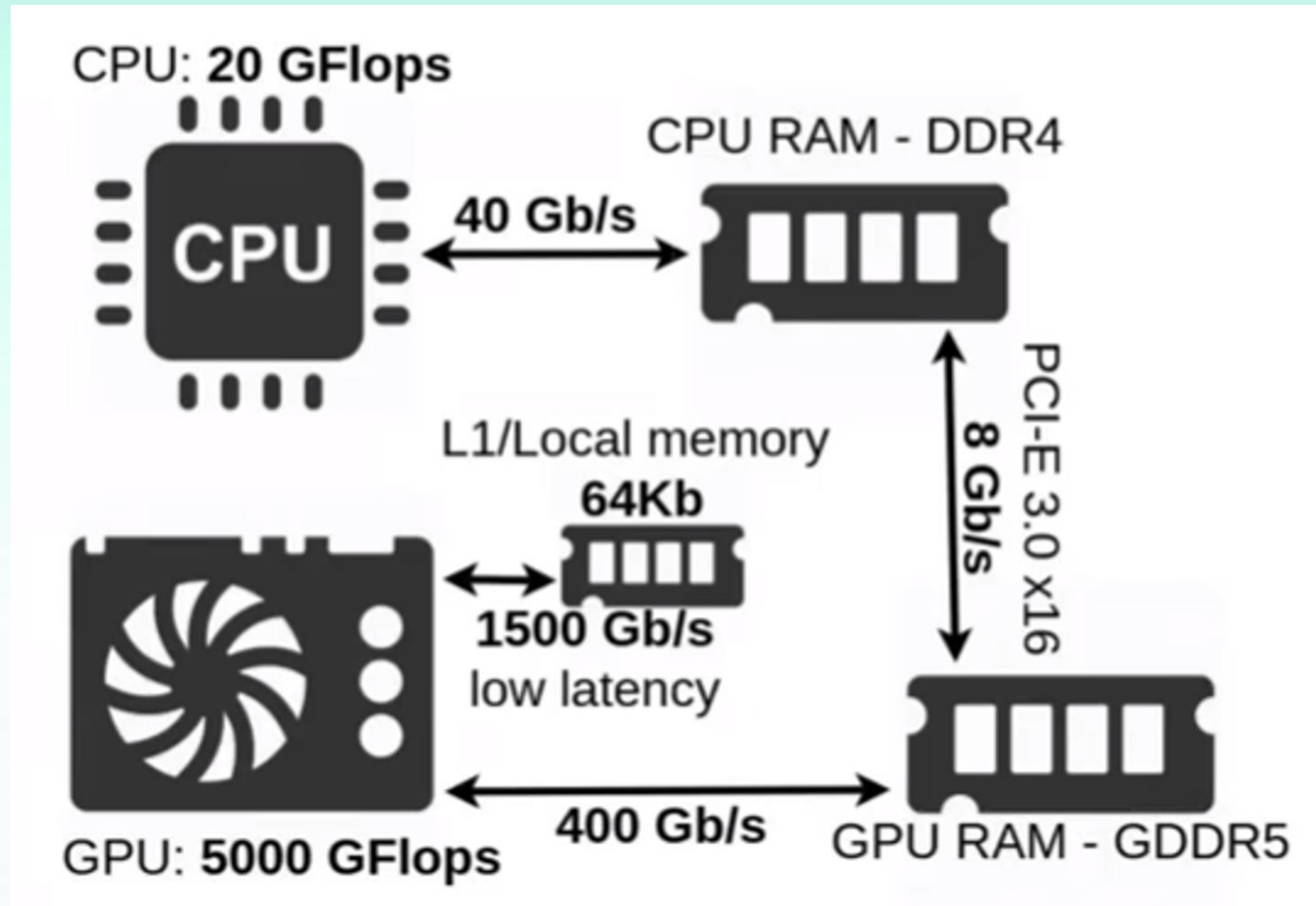
«Путь» видеокарт

Типичная видеокарта

Система охлаждения



«Путь» видеокарт



«Путь» видеокарт



Open GL

- OpenGL - прикладной программный интерфейс (API) для разработки приложений в области трехмерной графики.
- Был утвержден в 1992 году. Основой стандарта стала библиотека IRIS GL (Silicon Graphics Inc.)
- *Основная функция: интерактивная визуализация трехмерных моделей*

Open GL

- OpenGL ориентируется на следующие две задачи:
- Скрыть сложности адаптации различных 3D-ускорителей (GPU), предоставляя единый низкоуровневый API.
- Скрыть различия в возможностях аппаратных платформ, требуя реализации недостающей функциональности с помощью программной эмуляции.

Open GL

- По своей сути OpenGL — это просто спецификация, то есть документ, описывающий набор функций и их точное поведение. Производители оборудования на основе этой спецификации создают реализации — библиотеки функций, соответствующих набору функций спецификации.

Open GL

Open GL 1.0	1992	
Open GL 2.0	2001	Язык шейдеров
Open GL 3.0	2008	Khronos Group
Open GL 4.0	2010	
Open GL 4.5	2014	

Open GL

- **Архитектура Клиент-Сервер**
- Приложение -> «КЛИЕНТ» посылает команды, которые интерпретируются и обрабатываются реализацией Open GL -> «СЕРВЕРОМ»!
- Команды обрабатываются в том же порядке, в котором они получены сервером.

Open GL

- **Связывание данных** происходит во время КОМПИЛЯЦИИ команд – параметры функций интерпретируются, и копируются в память. Дальнейшее изменение параметров в теле программы не приводит к изменению данных внутри Open GL

Open GL

- **Буфер кадра (frame buffer).**

Память <-> окно на экране.

Цветовой буфер (до 4-х)

- + Буфер глубины (depth)
- + Буфер трафарета (stencil)
- + Буфер аккумуляции (accumulation)
- + Буфер мультсемплинга (multisampling)

Front buffer – передний буфер

Back buffer – задний буфер

Open GL

- **Контекст - Состояния**

Процесс трансформации графических примитивов и битовых карт (пикселей, фрагментов) управляется параметрами состояний (настроек) сервера Open GL.

Переменные состояния собраны в специальную структуру –
ГРАФИЧЕСКИЙ КОНТЕКСТ

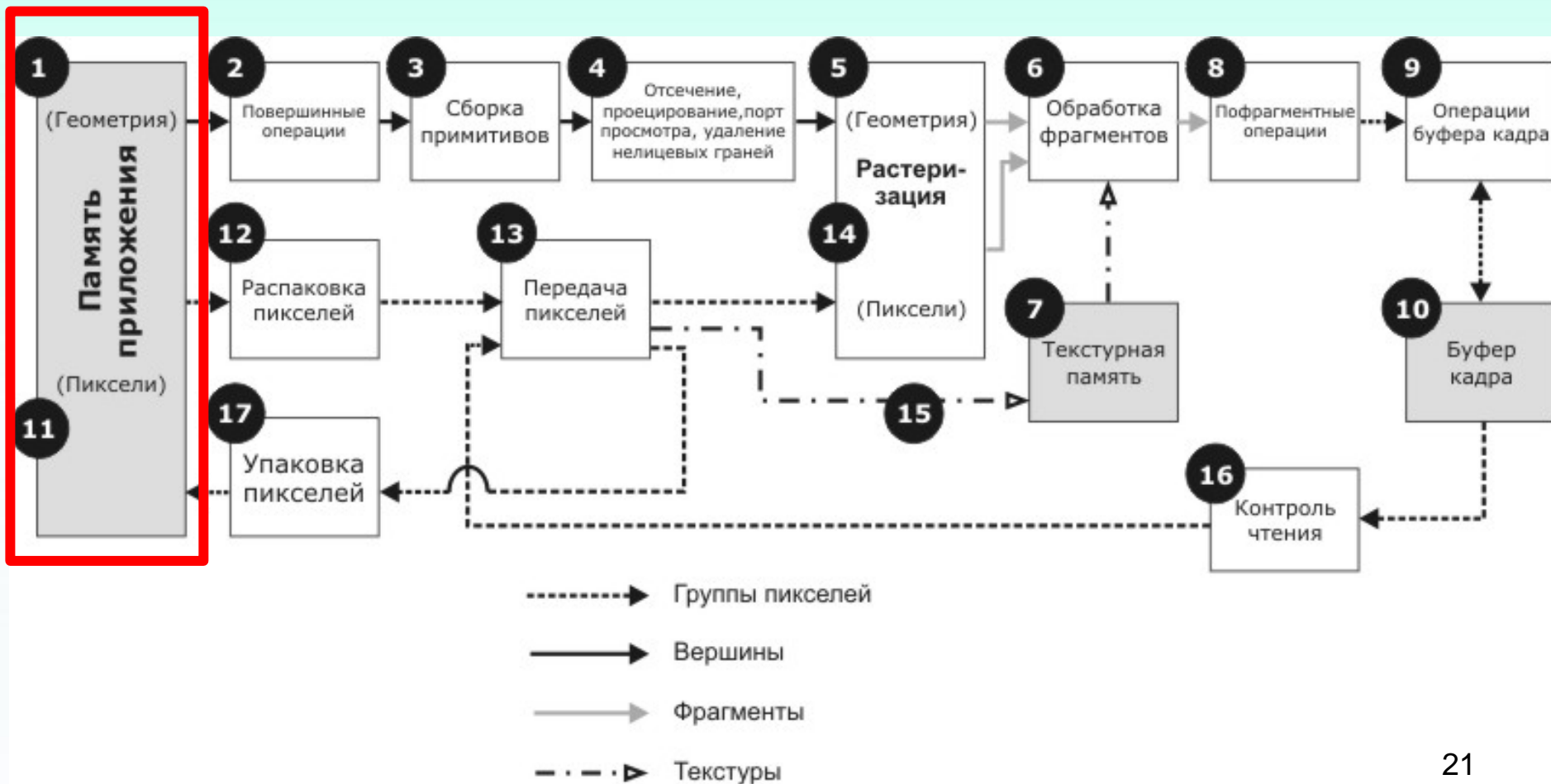
Open GL

Конвейер (fixed functionality)



Open GL

Конвейер (fixed functionality)



Open GL

1. Память приложения

Хранит данные для рисования графических примитивов (вершины, отрезки, полигоны и т.д.)

Хранит наборы пиксельных блоков (фрагменты, битовые карты).

Физически м.б. оперативная память
CPU или память GPU

Open GL

Конвейер (fixed functionality)



Open GL

2. Вершинные операции (T&L – transformation & Lighting)

- Координаты вершин преобразуются по видовым матрицам
- Преобразуются нормали
- Преобразуются текстурные координаты
- Освещение (До 8 ТОЧЕЧЕЧНЫХ источников)
- «Вершина за раз»
- Указатель на массив вершин
- Дисплейный список

Open GL

Конвейер (fixed functionality)



Open GL

3. Сборка примитивов

Формирование примитивов

Точки <- вершины

Отрезки <- 2 вершины

Треугольники <- 3 вершины

Полигоны <- произвольное количество
вершин

Open GL

Конвейер (fixed functionality)



Open GL

4. Обработка примитивов

Отсечение по пирамиде видимости

Перспективное преобразование

Отбрасывание задних граней (отбраковка)

Open GL

Конвейер (fixed functionality)



Open GL

5. Растеризация

Геометрический примитив -> совокупность пикселей (фрагментов). Каждый элемент примитива, полученный после растеризации, называется ФРАГМЕНТ. Для каждого примитива – свои методы растеризации (ширина, шаблон, ...)
Закраска – Гуро, Фонг

Антиалайсинг (сглаживание)

Open GL

Конвейер (fixed functionality)



Open GL

6. Обработка фрагментов

Выполняется текстурирование!!!

Вычисление текстурных координат и обращение к текстурной памяти (7)

Дымка

Сложение цветов

Open GL

Конвейер (fixed functionality)



Open GL

8. Фрагментные операции

- Отсечение по прямоугольнику
- Проверка прозрачности
- Отсечение по шаблону
- Проверка глубины
- Смещение цветов
- Сглаживание
- Логические операции

Open GL

Конвейер (fixed functionality)



Open GL

9. Операции с буфером кадра

Управление работой буфера кадра

- Работа без буферизации
- Работа с буферизацией
- Стерео изображение

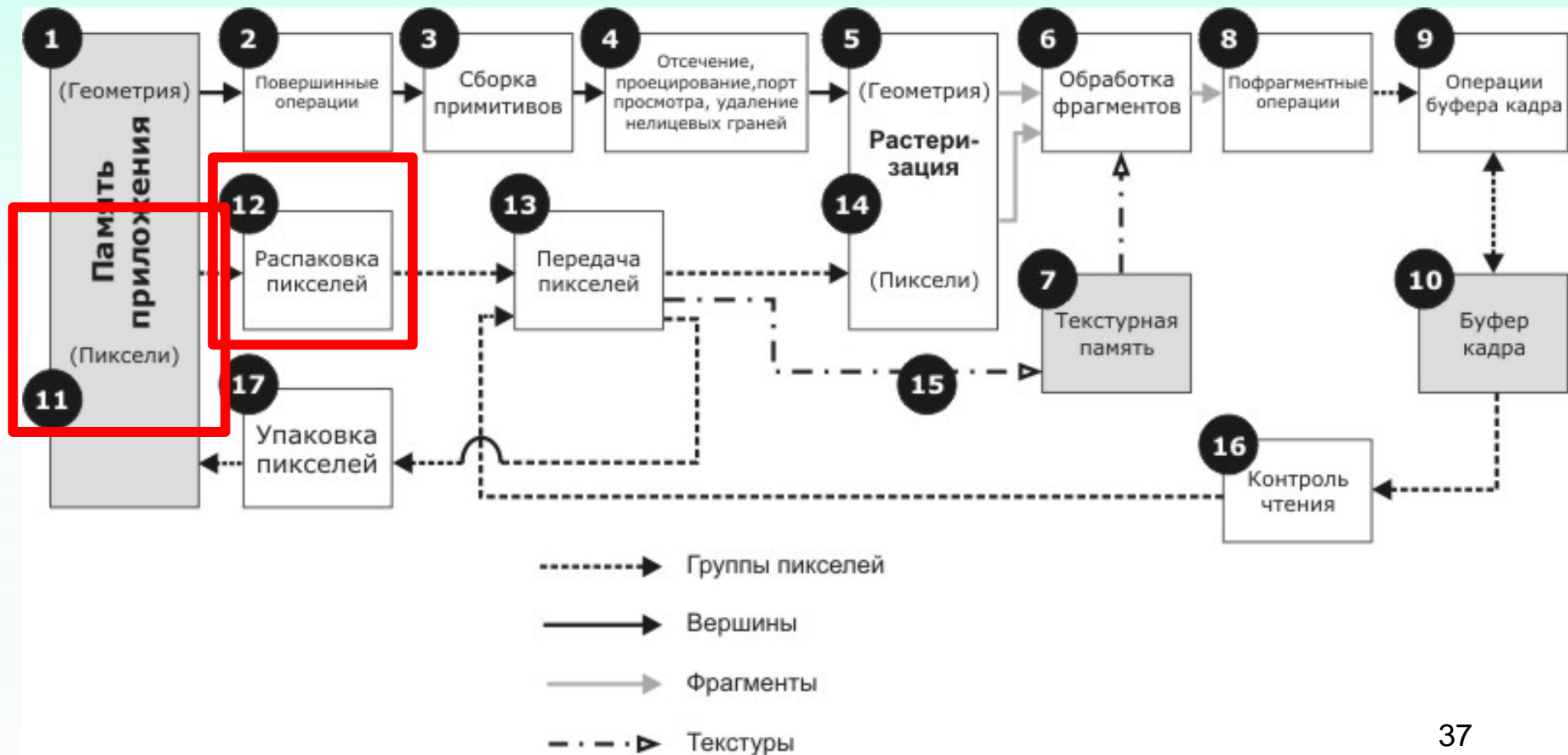
Управление дополнительными буферами

10. Буфер кадра.

Это мы видим на экране дисплея

Open GL

Конвейер (fixed functionality)



Open GL

11-12. Работа с пиксельными блоками (прямоугольниками)

Пиксельный блок (прямоугольник) хранится в памяти.

Распаковка – параметры цвета в буфер цвета, параметры глубины – в буфер глубины, маска – в буфер шаблона.

Open GL

Конвейер (fixed functionality)



Open GL

13. Перемещение пиксельных блоков

Масштабирование

Сдвиг

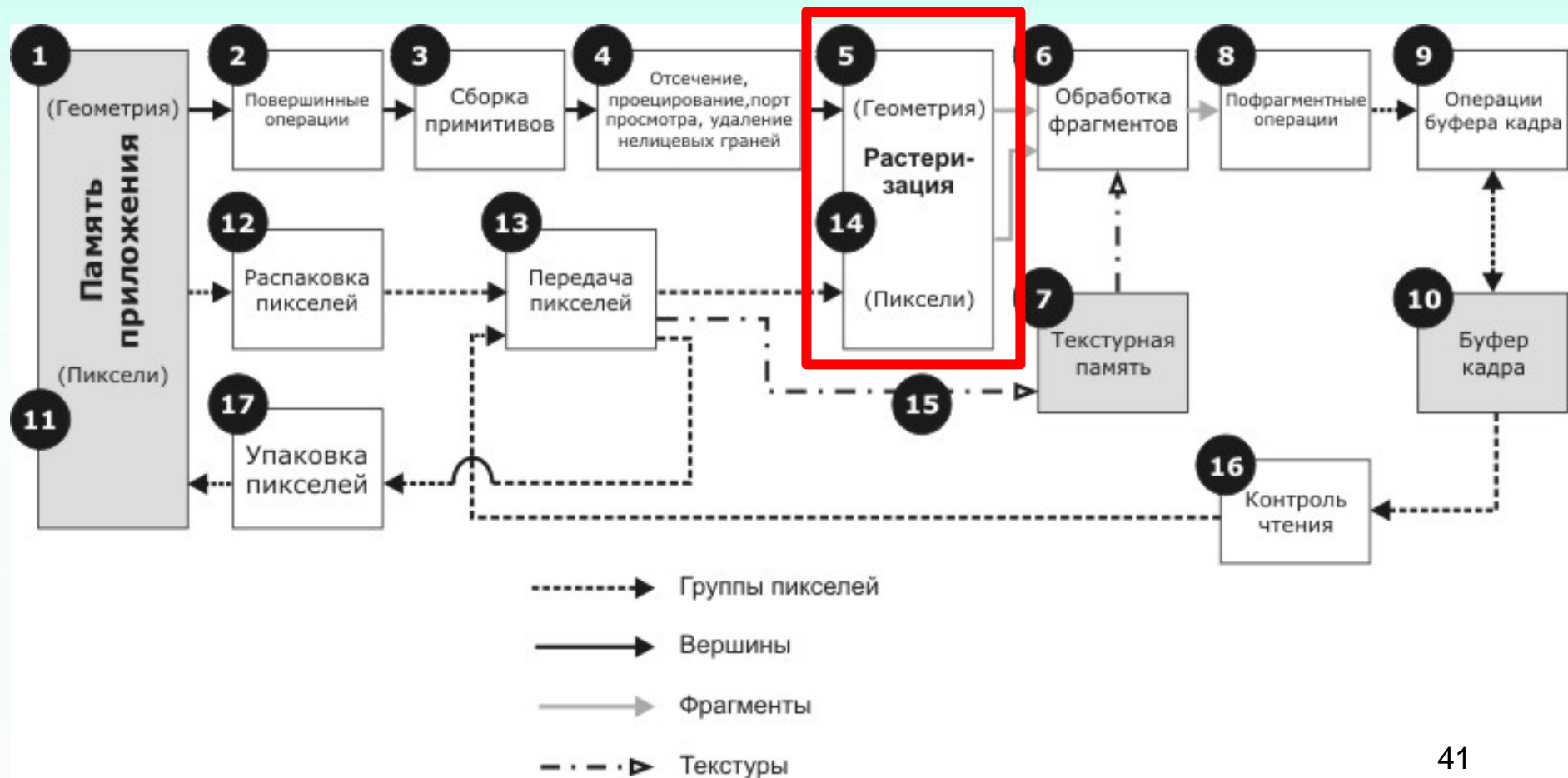
«Табличные» преобразования

Не стандартизовано:

- Искривление
- Линейное преобразование
-

Open GL

Конвейер (fixed functionality)



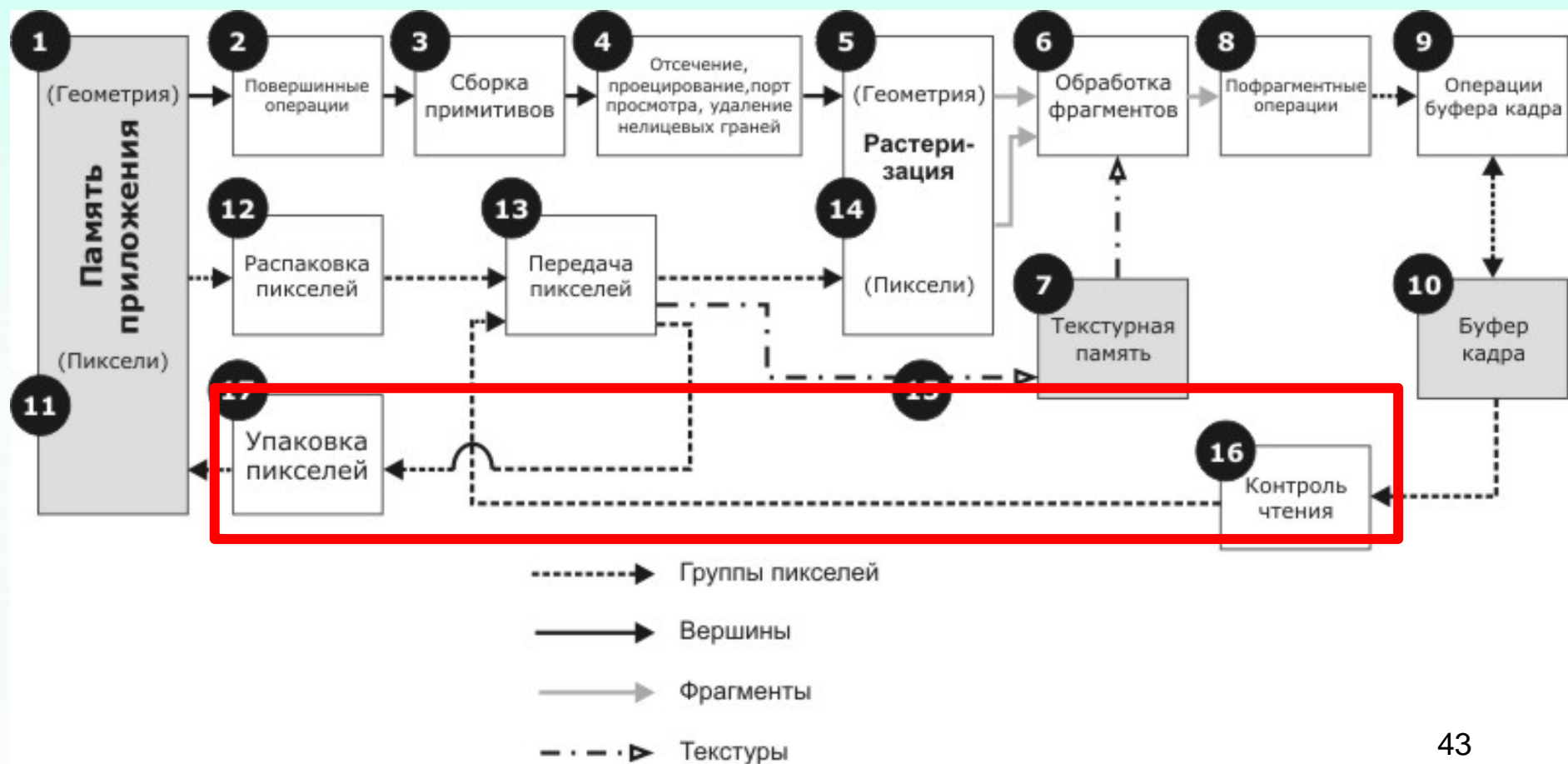
Open GL

14. Растеризация и окончательная обработка

Окончательное формирование фрагментов
(пиксельных блоков) для размещения в
буферной памяти

Open GL

Конвейер (fixed functionality)



Open GL

16-17. Чтение пиксельных блоков, Упаковка и запись

Чтение из буферной памяти пиксельных
блоков, проверка пиксельных блоков.

Запись в память приложения

Open GL Shading Language (2.0)



Open GL Shading Language

Shading Language – высокоуровневый язык программирования, предоставляющий возможность программирования обработки данных на двух ключевых стадиях графического конвейера.

Вершинный процессор – > вершинные шейдеры

Фрагментный процессор -> фрагментные шейдеры

Вершинный процессор

Vertex Processor

- Преобразование вершин
- Преобразование нормалей, нормализация векторов
- Генерация текстурных координат
- Преобразование текстурных координат
- Суммирование цветов
- Настройка освещения
- Наложение цвета материала

Фрагментный процессор

Fragment Processor

- Операции над интерполированными значениями
- Доступ к текстуре
- Наложение текстуры
- Туман
- Суммирование цветов

Особенность – фрагментные шейдеры не могут менять позицию (x, y) фрагмента

ЛИТЕРАТУРА

Официальный сайт

https://www.opengl.org/wiki/Main_Page

Open GL programming guide

<http://glprogramming.com/red/index.html>

Русскоязычный сайт

http://www.opengl.org.ru/books/open_gl/index.html

- **OpenGL super bible: comprehensive tutorial and reference / Richard S. Wright Jr. – 5th ed., Pearson Education Inc. - 2011**
- **Рост Р.Дж. Open GL. Трехмерная графика и язык программирования шейдеров. Для профессионалов.- Спб.: Питер, 2005.- 428 с.: ил.**
- **Бу М., Девис Т., Нейдер Дж., Шрайнер Д. Open GL. Руководство по программированию. Библиотека программиста. 4-е изд. – СПб.: Питер, 2006 .- 624 с.: ил.**

ЛИТЕРАТУРА

Тutorials:

Khronos

<https://www.khronos.org/>

Open GL tutorial

<http://www.opengl-tutorial.org/>

Уроки по OpenGL 3

<https://code.google.com/archive/p/gl33lessons/>

Вопросы для экзамена

ТЕМА: OPEN GL

1. Графический конвейер Open GL. Основные стадии обработки графической информации.
2. Библиотеки и основные функции Open GL
3. Поясните основные операции вершинного процессора.
4. Поясните основные операции фрагментного процессора.

END #10