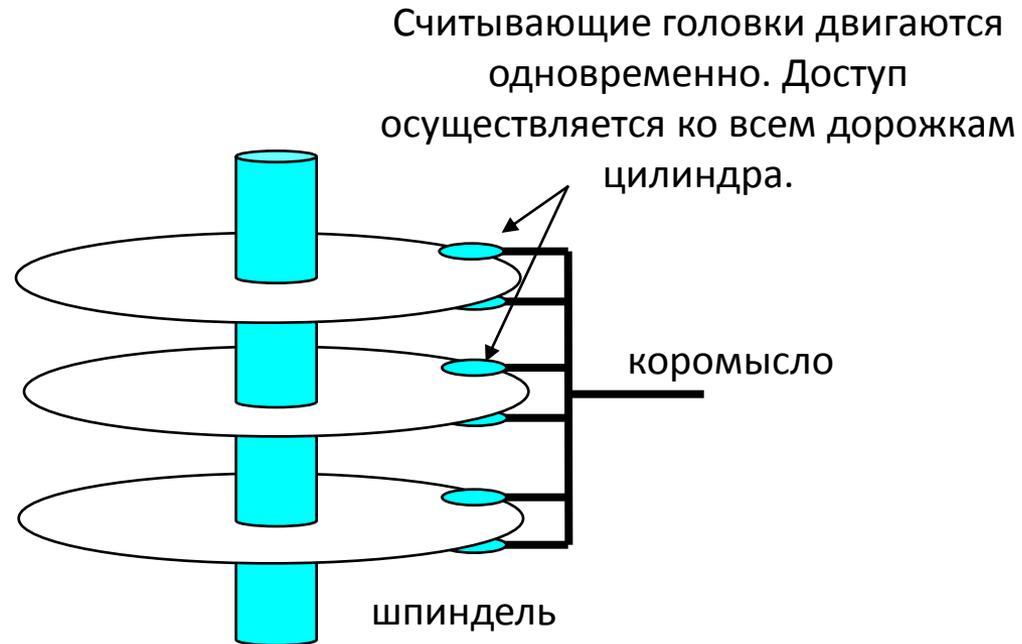


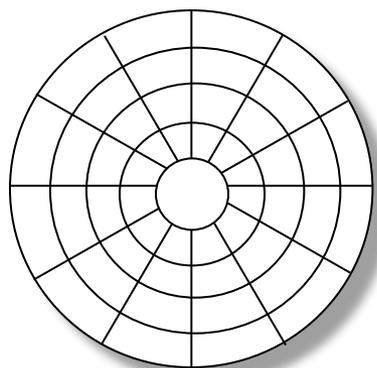
Лекция 21

23 апреля

Работа с диском (несколько пластин)



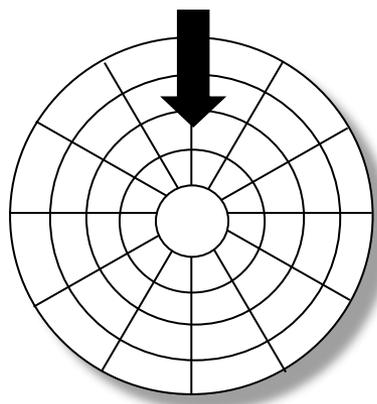
Структура диска – вид сверху на одну пластину



Поверхность разбита на дорожки

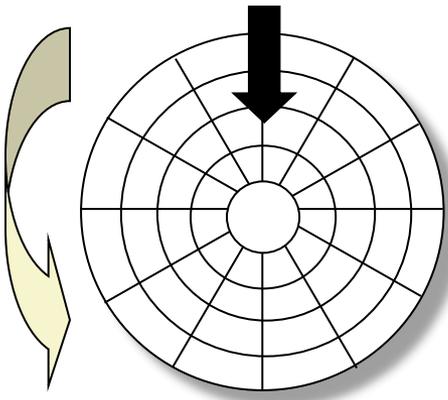
Дорожки разделены на сектора

Доступ к диску



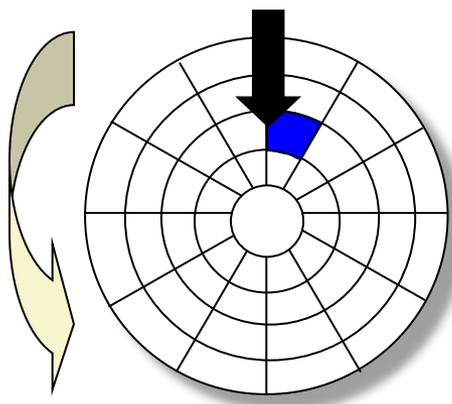
Считывающая головка в указанной
позиции над диском

Доступ к диску



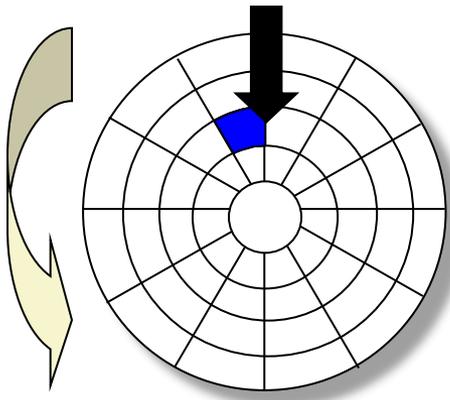
Направление вращения – против часовой стрелки

Доступ к диску – Чтение



Перед чтением синего сектора

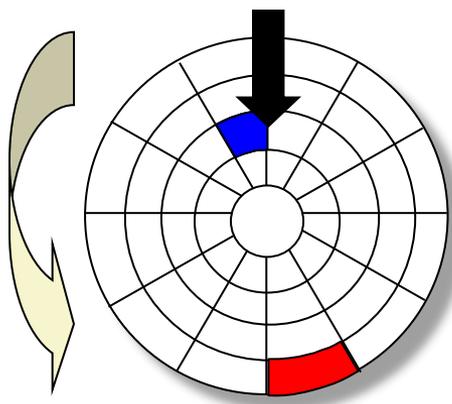
Доступ к диску – Чтение



Синий сектор
считан

После чтения синего сектора

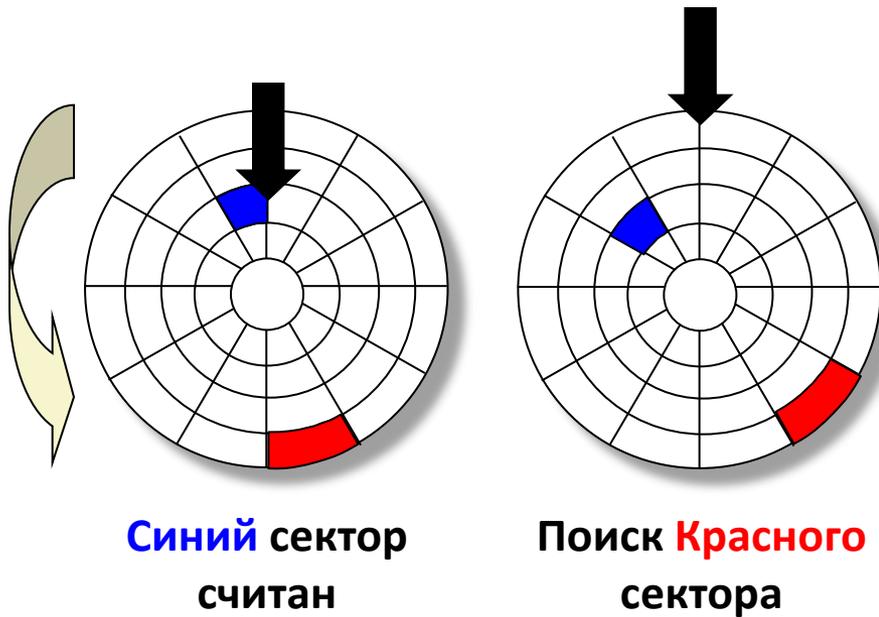
Доступ к диску – Чтение



Синий сектор
считан

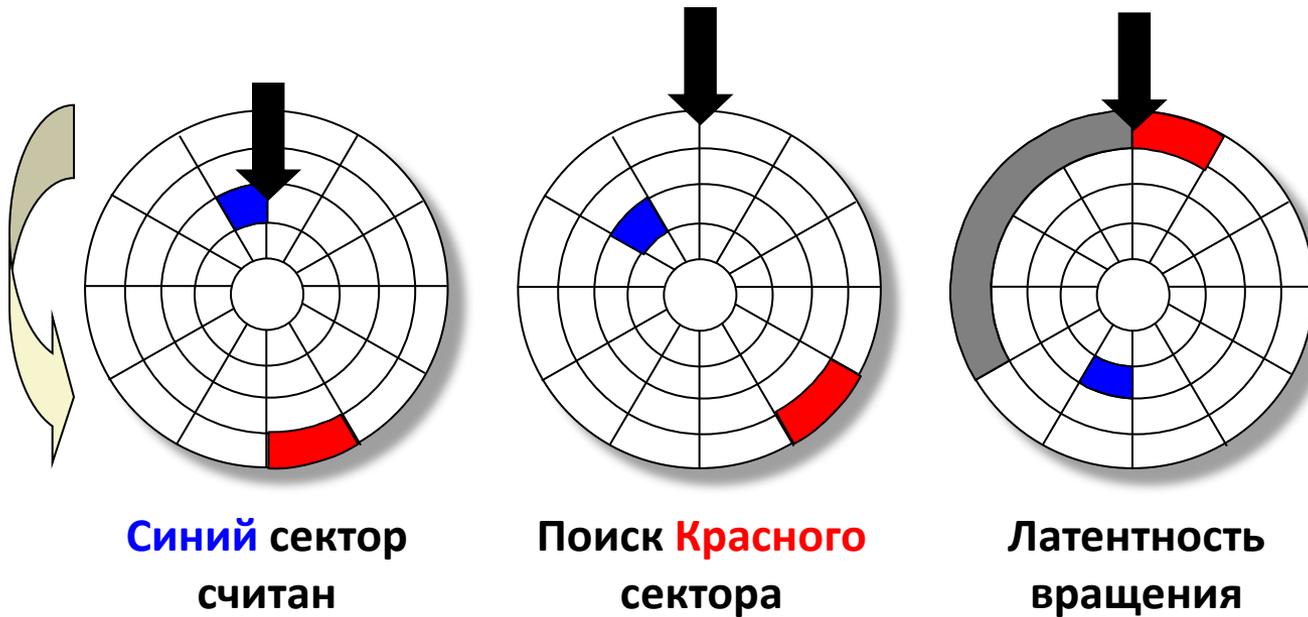
Поступил запрос на чтение красного сектора

Доступ к диску – Поиск



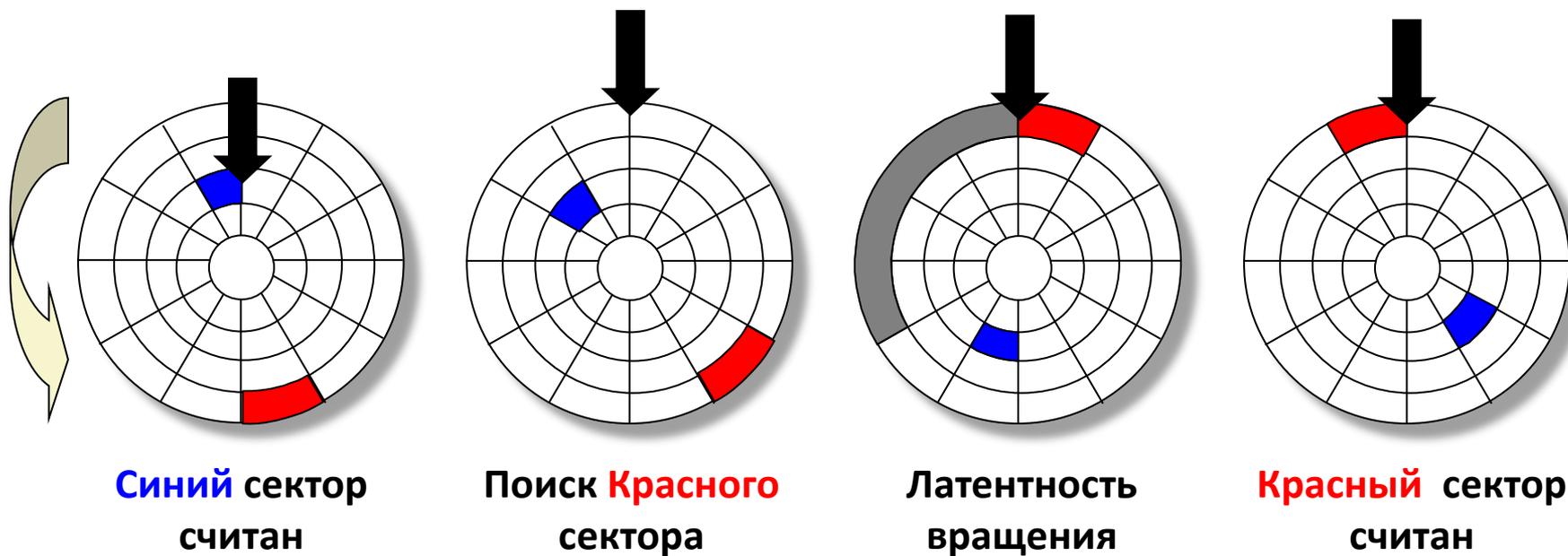
Ищем дорожку на которой расположен красный сектор

Доступ к диску – временная задержка из-за вращения



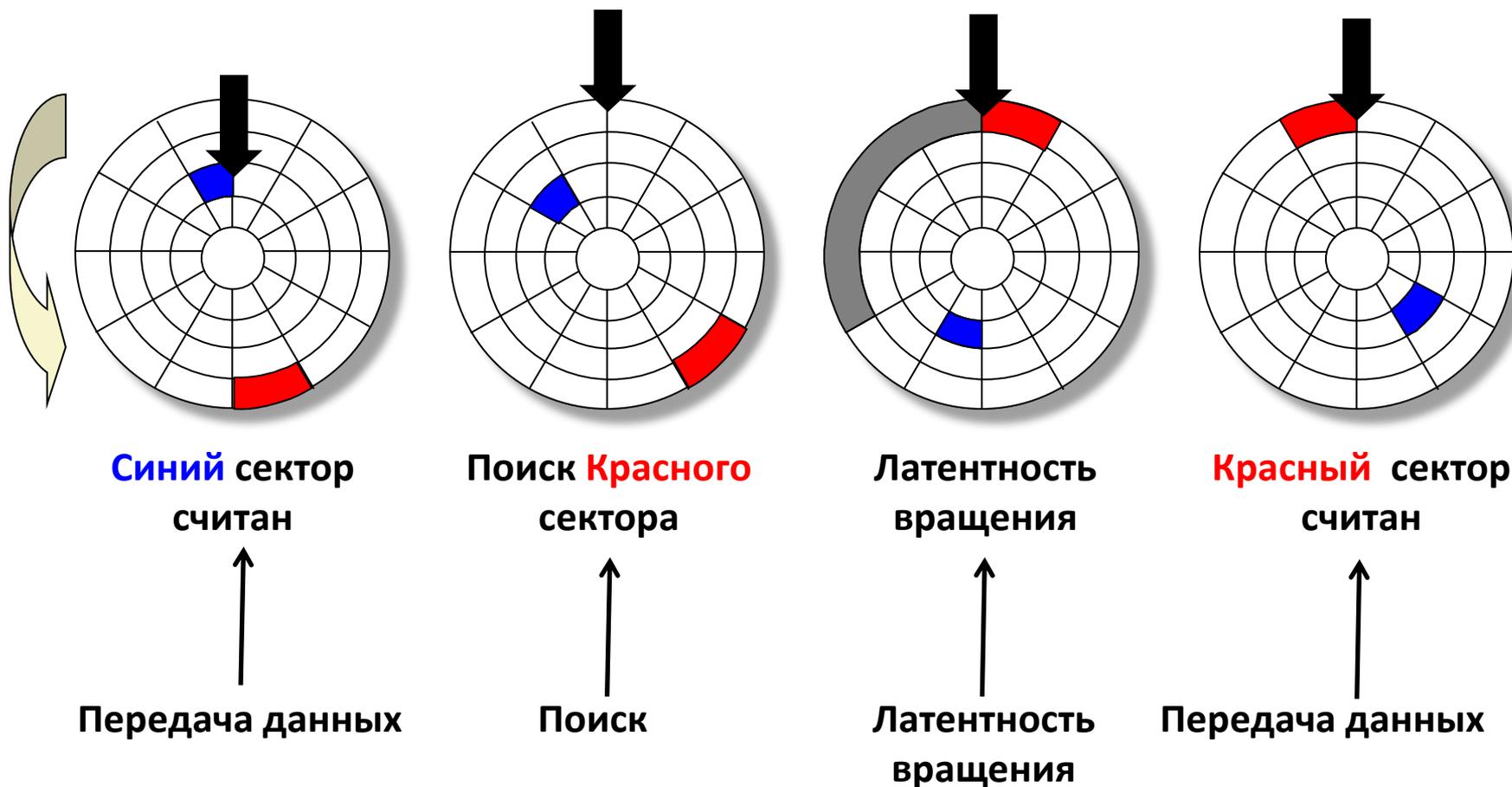
Вынужденное ожидание того момента, когда красный сектор достигнет считывающей головки

Доступ к диску – Чтение



Чтение красного сектора завершено

Доступ к диску – распределение времени



Время доступа к диску

- $T_{\text{доступа}} = T_{\text{ср. поиск}} + T_{\text{ср. вращения}} + T_{\text{ср. передача}}$
- **Время поиска** ($T_{\text{ср. поиск}}$)
 - Время, требуемое для перемещения считывающей головки в цилиндр, содержащий требуемый сектор.
 - Как правило $T_{\text{ср. поиск}}$ занимает 3—9 мс.
- **Латентность вращения** ($T_{\text{ср. вращения}}$)
 - Время ожидания момента, когда первый бит запрашиваемого сектора достигнет считывающей головки.
 - $T_{\text{ср. вращения}} = 1/2 \times 1/\text{RPM} \times 60 \text{ с} / 1 \text{ мин}$
 - Типичная скорость вращения – 7200 RPM. $T_{\text{ср. вращения}} \approx 4 \text{ мс}$.
- **Время передачи** ($T_{\text{ср. передача}}$)
 - Время чтения содержимого сектора.
 - $T_{\text{ср. передача}} = 1/\text{RPM} \times 1/(\text{ср. \# секторов на дорожке}) \times 60 \text{ с./1 мин}$.

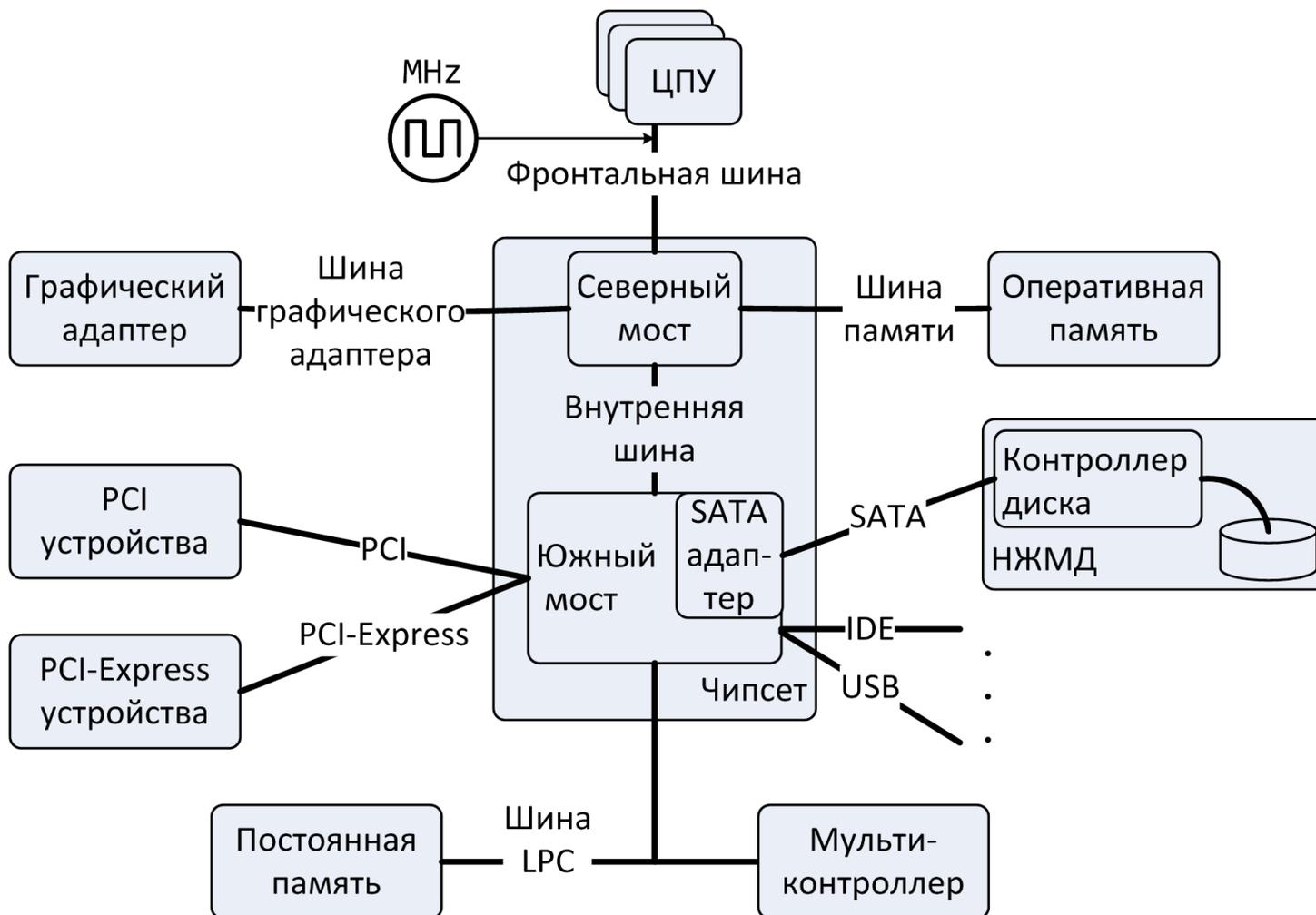
Пример оценки времени доступа

- Исходные характеристики:
 - Скорость вращения = 7,200 RPM
 - Среднее время поиска = 9 мс.
 - Среднее # секторов на дорожке = 400.
- Оцениваем слагаемые и общую сумму:
 - $T_{\text{ср. вращения}} = 1/2 \times (60 \text{ с}/7200 \text{ RPM}) \times 1000 \text{ мс} = 4 \text{ мс}.$
 - $T_{\text{ср. передача}} = 60/7200 \text{ RPM} \times 1/400 \text{ с/дорожка} \times 1000 \text{ мс} = 0.02 \text{ мс}$
 - $T_{\text{доступа}} = 9 \text{ мс} + 4 \text{ мс} + 0.02 \text{ мс}$
- Выводы:
 - Время передачи существенно меньше остальных слагаемых.
 - Читать первый бит из сектора – «дорогая» операция, считывание остальных битов – «дешево».
 - Время доступа SRAM $\approx 4 \text{ нс}$ для двойного слова, DRAM $\approx 60 \text{ нс}$
 - Диск медленнее SRAM в 40,000 раз, и ...
 - ... в 2,500 раз, чем DRAM.

Логические блоки

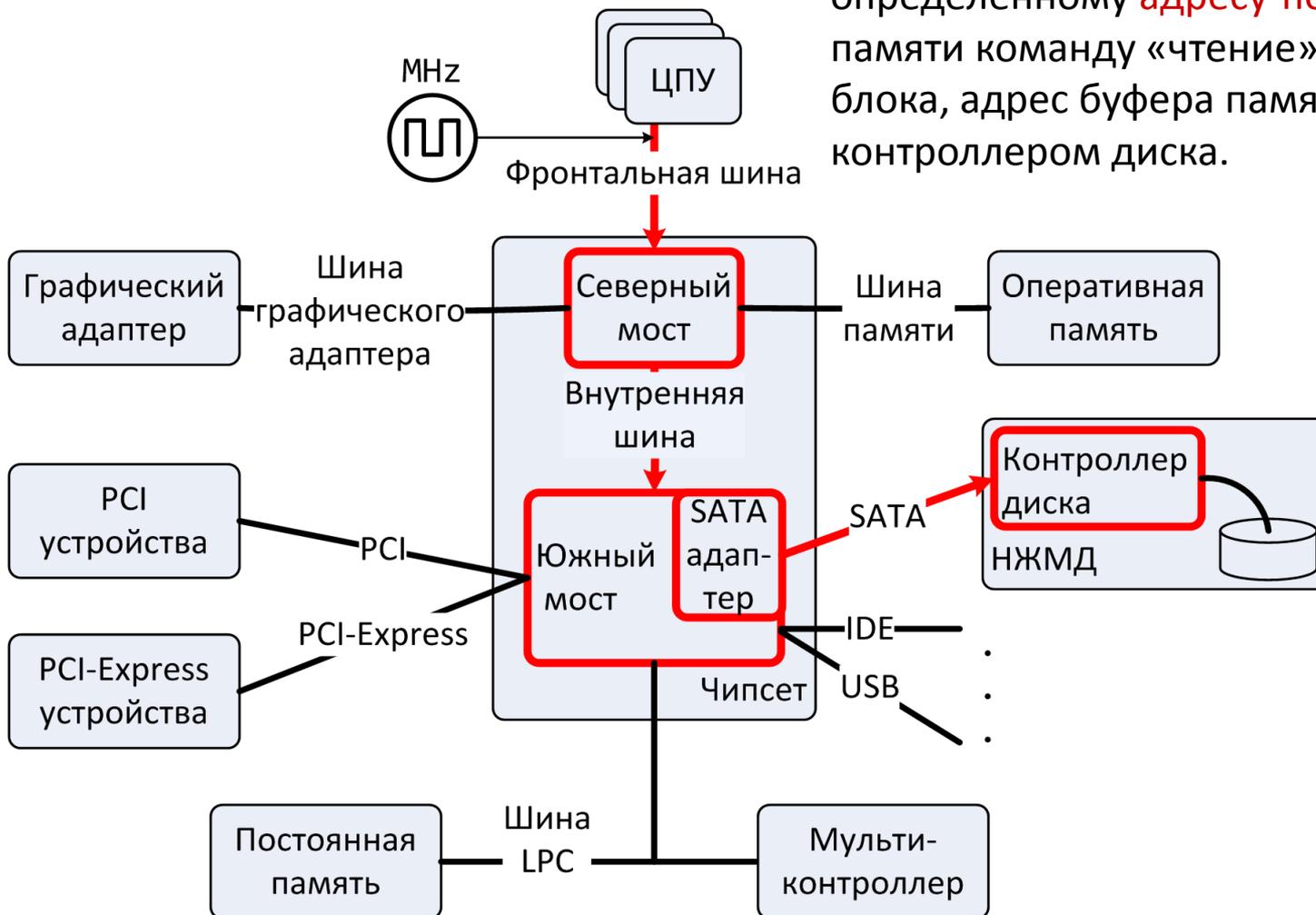
- Более простой метод обращения к данным:
 - Сектора рассматриваются как последовательность **логических блоков** (0, 1, 2, ...)
- Соответствие между логическими блоками и (физическими) секторами
 - Отображение поддерживается аппаратурой + «прошивкой» – контроллером диска.
 - Номер логического блока → (поверхность, дорожка, сектор).
- Защита от выхода из строя отдельных цилиндров. Каждая зона записи содержит запасные цилиндры.
 - Размер диска после форматирования становится ощутимо меньше.

SATA: шина ввода/вывода



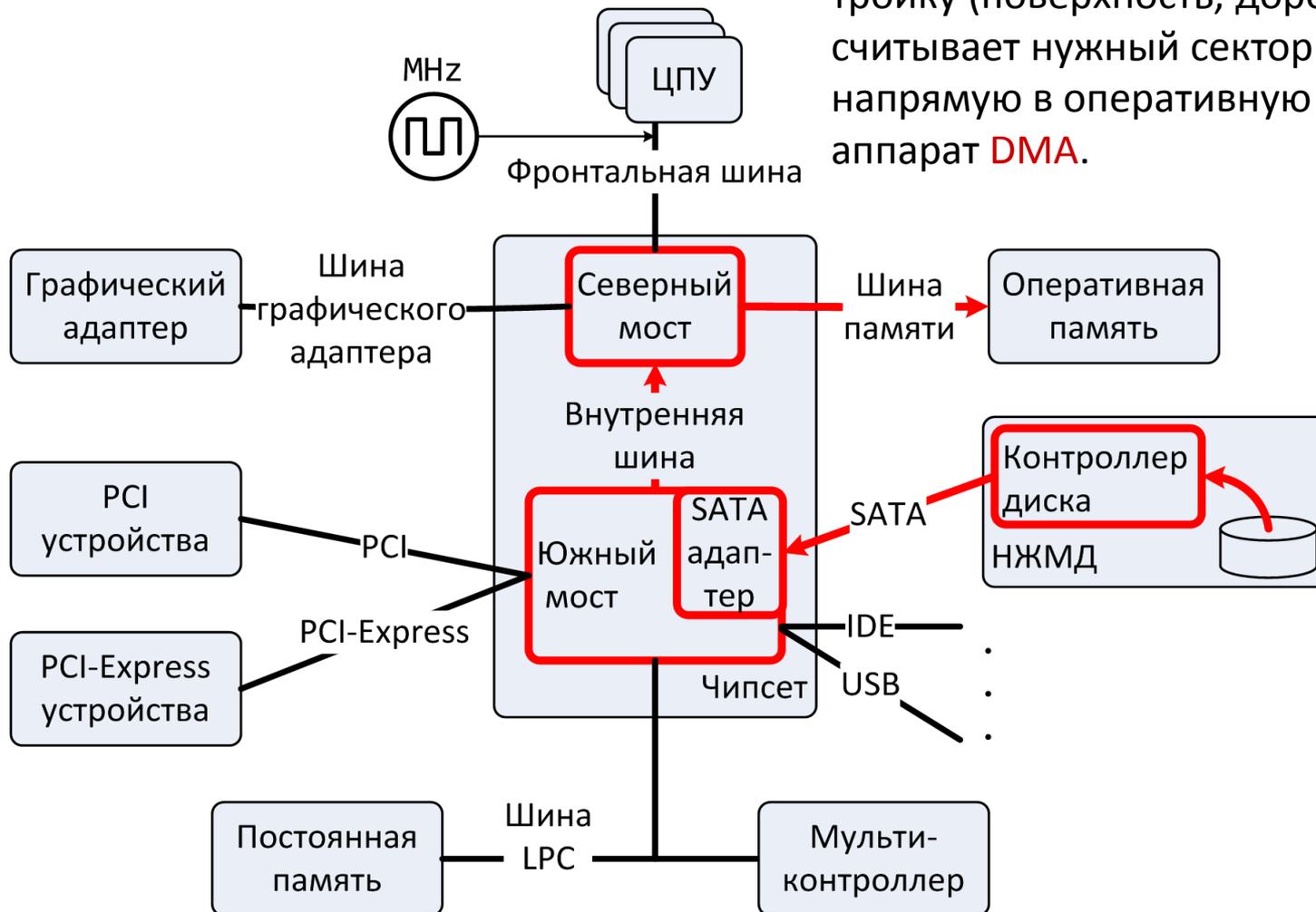
Чтение сектора (1)

ЦПУ запускает чтение диска, записав по определенному **адресу-порту** оперативной памяти команду «чтение», номер логического блока, адрес буфера памяти. Порт связан с контроллером диска.



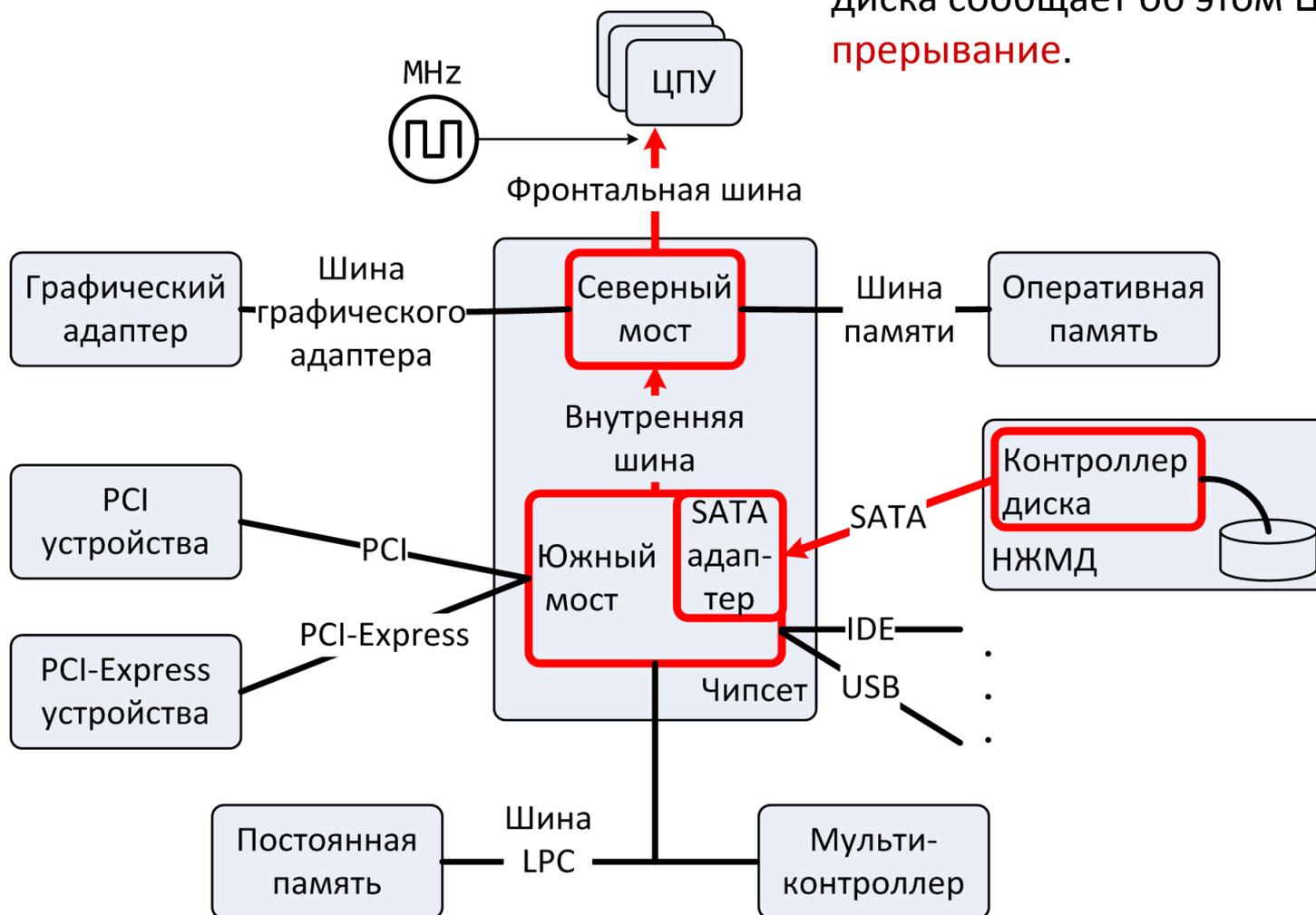
Чтение сектора (2)

Контроллер диска преобразует номер блока в тройку (поверхность, дорожка, сектор), считывает нужный сектор и передает его напрямую в оперативную память, используя аппарат **DMA**.



Чтение сектора (3)

Когда **DMA-трансферт** закончен, контроллер диска сообщает об этом ЦПУ, вызывая **прерывание**.



Port IO vs. Memory Mapped IO

- Port IO: помимо пространства памяти вводится дополнительное пространство портов ввода/вывода
 - Работа с периферией: команды `in` и `out`
 - Все данные проходят через ЦПУ
 - Удобно при небольшом размере памяти
- Memory Mapped IO: все управляющие регистры устройств отображаются на определенные адреса оперативной памяти
 - Требуется программировать контроллер памяти/северный мост
 - Перекрытая память не используется
 - Существенно более высокая производительность

```
• IN EAX, DX ; AX, AL
  • EAX ← I/O[DX]
• OUT DX, EAX ; AX, AL
  • I/O[DX] ← EAX
```